Queen Mary University of London

PhD Thesis

Nominal Models of Linear Logic

Thomas David Cuvillier

supervised by

Dr. Nikos Tzevelekos

Submitted in partial fulfillment of the requirements of the Degree of Doctor of Philosophy

I, Thomas David Cuvillier, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged below and my contribution indicated. Previously published material is also acknowledged below.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

Signature: Thomas Cuvillier Date: 17/01/2017

Details of collaboration and publications:

Rechercher le commun qui n'est pas semblable. C'est ainsi que le poête peut dire: Une hirondelle poignarde le ciel, et fait d'une hirondelle un poignard.

Georges Braque

This work was supported by Queen Mary University of London.

More than 30 years after the discovery of linear logic, a simple fully-complete model has still not been established. As of today, models of logics with type variables rely on di-natural transformations, with the intuition that a proof should behave uniformly at variable types. Consequently, the interpretations of the proofs are not concrete. The main goal of this thesis was to shift from a 2-categorical setting to a first-order category.

We model each literal by a pool of resources of a certain type, that we encode thanks to sorted names. Based on this, we revisit a range of categorical constructions, leading to nominal relational models of linear logic.

As these fail to prove fully-complete, we revisit the fully-complete game-model of linear logic established by Melliès. We give a nominal account of concurrent game semantics, with an emphasis on names as resources. Based on them, we present fully complete models of multiplicative additive tensorial, and then linear logics. This model extends the previous result by adding atomic variables, although names do not play a crucial role in this result. On the other hand, it provides a nominal structure that allows for a nominal relationship between the Böhm trees of the linear lambda-terms and the plays of the strategies.

However, this full-completeness result for linear logic rests on a quotient. Therefore, in the final chapter, we revisit the concurrent operators model which was first developed by Abramsky and Melliès. In our new model, the axiomatic structure is encoded through nominal techniques and strengthened in such a way that full completeness still holds for MLL. Our model does not depend on any 2-categorical argument or quotient. Furthermore, we show that once enriched with a hypercoherent structure, we get a static fully complete model of MALL.

Acknowledgements

First, I would like to thank my supervisor, Nikos Tzevelekos, for his constant support, guidance, for his tremendous proof-reading, and for being a constant source of inspiration (to this acknowledgement section itself). I also have a special thought for my second supervisor, Edmund Robinson, who placed confidence in me along all these years (although I am still not sure why).

I am particularly grateful to Paul-André Melliès, who has, after being an unforgettable teacher of category theory, provided guidance to me, helped me find this PhD, was always present to answer my questions, and agreed to examine this thesis. This thesis owes a lot to him, and most of the ideas exposed along it emanates from his work.

I would also like to thanks Samson Abramsky, whose work has been greatly influential to me, and who notably co-wrote (with Paul-André Melliès) "Concurrent Games and Full-Completness" [10], a paper that I found so great that I have worked on semantics of Linear Logic ever since I read it.

I greatly thanks Nikos Tzevelkos, Paulo Oliva and Paul-André Melliès (the latter two acting as examiners, the former as supervisor) for reading the thesis (something I wouldn't wish on my worst enemy) and suggested several corrections and improvements.

Many thanks go to my friends and family who supported me during this Ph.D. I notably think of Sandra, who has been there for me all these years, and my parents who supported and had faith in me ever since I was born. Last but not least, all my colleagues from the Theory Group, and especially my fellow Ph.D students (and post-doc) whom I shared CS420-CS438-CS249 (in chronological order) with during this time.

Finally, I would like to acknowledge the financial support of Queen Mary University of London.

Contents

List of Figures

Part I

Introductory Chapters

Chapter 1

Introduction

1.1 Layman's introduction

1.1.1 The calculus of computation

Arguably one of the prominent goals of theoretical computer science is to establish a *science* of computation. Just as the physicist can predict the behaviour of a system thanks to an associated theory; or the architect may ensure, thanks to a reliable science of materials and engineering, that a building will stand; one would like to have a robust theory of computation, where it is possible to predict the behaviour of programs, check their properties and make sure they process without defects.

Interestingly, computer science as a discipline was born *before* the actual birth of modern computers designed with electronic circuits. The foundation of the discipline deals with any computing machines that rely on laws of physics. Therefore, it encompasses, for instance, the study of mechanical calculators such as the one Pascal invented in the early 17^{th} century. From 1933 to 1936, three attempts laid potential foundations for a science of computability. The first one, designed by Kleene, consisted of μ-recursive functions, whose outputs could be computed finitely following a recursive procedure. The second one relied on the design of a theoretical machine, the Turing machine. Finally, the last one leant on a new calculus, the lambda-calculus. It was proven by Church and Turing that these 3 designs lead to the same set of functions, making them equivalent.

The lambda-calculus can be described as the calculus of composition. The basic idea is to describe a computing system, or machine, by a term. This one has inputs, or arguments, and outputs. Systems can be composed by putting them side by side, corresponding to the idea that the outputs of the first can be wired into the inputs of the second. A defect of the lambda calculus is that it is too inclusive, allowing for self-application. As a result, there are terms that correspond to never-ending computations (for instance, think about a machine whose outputs are rewired into its inputs). We often say that such terms "loop". Therefore, it was sought a condition on the formation of terms that would ensure they denote well-founded computations. The solution was brought through *typing*. This consists in giving a type to all terms of the calculus. For instance, a term that takes inputs of type A and returns outputs of type B is of type $A \Rightarrow B$. On the other hand, a term that takes inputs of type A, and returns as output functions of type $B \Rightarrow C$ has type $A \Rightarrow (B \Rightarrow C)$. Finally, a term that takes inputs of type A, B and returns outputs of type C, D has type $(A \times B) \Rightarrow (C \times D)$. One can prove that the lambda-calculus enriched with "simple" types, called simply-typed lambda calculus, is convergent: every computation it encodes terminates. This calculus is the at the heart of current programming languages, that are designed through blocks, called functions, that are specified by the type of their inputs and outputs.

At last, we would like to emphasize that the lambda-calculus is by no means the unique calculus devised to reason about computation. Many more have been developed, following different paradigms. For instance, the π-calculus can be understood as the calculus of communications, emphasising computations as communications between different processes. Arguably what made the lambda-calculus so popular is its simplicity; it morally simply consists in composing functions.

1.1.2 Interlude: logics

In this subsection we will talk about something without, *a priori*, no relations to the previous paragraph: logics. Logic is the art of formalising the rules that govern a valid argument. To the novice, it might seem like logic is intrinsically related to the notion of truth, and that a reasoning is logically valid if it is common sense. For instance, when scientists try to prove an argument, either orally or mathematically, they never begin by displaying the logical rules on which the argument relies, as those are universally accepted. This set of rules, that forms the core of the human mathematical reasoning, is called classical logic. To enable the reader to grasp the underlying concept, we present some of its rules, together with their labels:

- (Axiom) A true entails A true.
- (Left and) If A true entails C true, then A and B true entails C true.
- (Right and) If A true entails B true, and if A true entails C true, then A true entails B and C true.
- (Left negation) If A true entails B or C true, then A true and B false entails C true.
- (Right negation) If A and B true entails C true, then A true entails either B false, or C true.
- ...

Classical logic is the logical system used throughout all mathematical textbooks. Its foundation relies on the following principle: a proposition is either true or false. This is called, in Latin "principium tertii exclusi", translated as law of the excluded middle. A proposition cannot be neither true nor false. In particular, this entails that negating twice a proposition leaves it unchanged, written $\neg\neg A \simeq A$. In that case, we say that the negation is **involutive**, meaning applying twice has no noticeable effects on the original proposition. Literally, it translates into: if A is true then A is not not true (that is, not false). And, if A is not false then A is true.

This involutivity of the negation allows, in mathematics, to prove the existence of objects without defining them. Such proofs are proofs by contradiction. We assume something is false, and prove a contradiction. We then deduce that it is not false, that is, it is true. For instance, the intermediate value theorem (that states that a continuous function f that has both positive and negative values inside a interval, must accept an element x within this interval such that $f(x) = 0$), accepts an easy proof by contradiction. That is, the proof assumes that there is no such x, and deduces a contradiction.

Intuitionism is a branch of mathematical philosophy that views mathematics as the sole result of human mind constructions, and not as an objective truth. It resulted in constructive mathematics, that were defined as the restriction of mathematics where only the objects that are constructed, or exhibited, are accepted to exist. The attempt to formalise this new kind of reasoning leads to the definition of a fragment of classical logic, called intuitionistic logic. The most salient feature of intuitonistic logic is its different handling of the negation. In it, if A is true entails that A is not false, the reverse does not necessarily hold. That is, A is not false does not entail that A is true. In other terms, the law of excluded middle is rejected: a formula can be neither true nor false. This corresponds to a proposition A such that one cannot prove neither A, nor $\neg A$.

Finally, the incompleteness theorems proven by Gödel in the early 1930's ended the dream of defining a universal notion of truth through logic and rigorous reasoning. These theorems are well-known within the community, but too often misunderstood and misinterpreted, and we will not try summarising them here. However, their most important consequence is surely that there is no proof of the consistency of mathematics (that is, the basic axioms that we use to do mathematics nowadays might be in contradiction to one another). Furthermore, there are mathematical propositions that are true, but not provable. If these discoveries closed the doors on the dreams of some scientists at the time, they opened many others. Indeed, by distancing logic from the notion of truth, it allowed the definition of many other logical systems, each emphasising a different paradigm (that is, way of reasoning). Logic nowadays deals with the way of deducing propositions, not with concluding about their truthfulness.

1.1.3 The Curry-Howard-Lambek correspondence

At the heart of a consequent part of the research currently happening in theoretical computer science lies the correspondence, first established in 1969 by Howard, between the terms of the simply typed lambda-calculus and proofs of intuitionistic logic. The discovery, linking two seemingly unrelated formalisms, has been deeply studied and enriched since then. This correspondence is far from accidental. In constructive mathematics, a proof seemingly behaves as an algorithm. The analogy is drawn between the way the proof relies on its hypotheses to establish the conclusion, and the handling of the inputs of the lambda-term in order to produce its outputs. To the composition of programs corresponds the composition of proofs through entailment. That is, a proof of A entails B can be composed with a proof of B entails C to provide a proof that A entails C. Finally, it exposes the computational nature of proofs of intuitonistic logic.

The context of this thesis is the denotational semantics field of research, that aims to devise proper mathematical models for programs. One of the arguably main purposes of semantics is to convey an abstract representation of programs, that defines their meaning in terms of computation. Following the Curry-Howard correspondence, this research deals on an equal footing with denotational semantics of proofs, conceptualising proofs and dealing with them as mathematical objects. Mathematical models considered along this thesis are categories. That is, we consider a collection of objects, corresponding to types / formulas. The model consists, for each pair of objects A, B of a set of elements which are seen as potential representations of proofs (respectively programs) that A entails B (respectively that take an object of type A as input and output one of type B). Furthermore, just as we can compose proofs/programs, we can compose the elements of our models. The first step of much research in denotational semantics consists in axiomatising the properties that a category must satisfy to provide a sound model of a given system of programs/proofs. Lambek characterised precisely those that were sound models of the simply typed lambda-calculus in the 1970's, and hence provided a robust basis for definitions of future models.

Since this triptych discovery, a great deal of work has been produced in order to generalise this correspondence to various types of logics and, in particular, to classical logic. A major stumbling block in this direction was the discovery that the only categories that could soundly model classical logic where "boolean": each set between objects A, B was either empty, or consisted of a unique element. In other terms, there is no abstract representation of proofs of classical logic: all proofs are equivalent. If an adaptation of the simply typed lambda-calculus has been provided to cater for the specificities of classical logic (and, in particular, its double negation), this calculus does not enjoy all the equivalences one naturally expects to hold.

1.1.4 Intuitonistic linear logic

Linear logic was built as a refinement of intuitionistic logic, where some hidden computational aspects were made explicit. For instance, let us consider the following proposition:

$$\text{If } (A \text{ is true and } B \text{ is true}) \text{ then } A \text{ is true.} \quad (1)$$

A proof of this proposition in intuitionistic logic basically consists in taking A, B as hypotheses, disregarding B and returning the input A as conclusion. Hence a proof of intuitonistic logic has the ability to disregard hypotheses. Similarly, a proof of:

$$\text{If } (A \text{ is true}) \text{ then } (A \text{ is true and } A \text{ is true}) \quad (2)$$

duplicates the hypothesis A to produce the conclusion.

Linear logic is the system obtained from intuitonistic logic by preventing the proof-system, and hence the programs, from disregarding hypotheses or duplicating them implicitly. Hence, it makes the handling of hypotheses throughout the proof, or the program, explicit. This underscores the role of hypotheses as resources. For instance, the proposition (1) above would now become:

If (A is true and B is true) then (A is true and $\top$)

where $\top$ is a "garbage collector", that collects hypotheses that are unused: in this case, B. Similarly, the second proposition could be translated as:

If !(A is true) then (A is true and A is true)

where the symbol " !" makes explicit that the resource A is available in a "as much as you want" quantity.

This logic of resources has an intuitive counterpart in terms of daily speech, by seeing each proposition as a consumable good. For instance, let A denotes 10 cents, and B denotes a "pain au chocolat". Then, according to some french politicians, A entails B, meaning:

With 10 cents, you can buy a pain au chocolat. (3)

However, A entails B does not lead to A entails B and B, which would correspond to:

With 10 cents, you can buy a pain au chocolat and another pain au chocolat,

that is, two pains au chocolat. The ! can be seen as an infinite supply, and $\top$ would be akin to the bank, allowing you to get rid of your spare change.

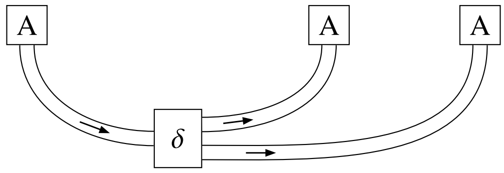
Intuitonistic linear logic emphasises the roles of hypotheses as resources within the proof, and the role of the proof as a resource management machine, proceeding in channelling resources.

1.1.5 The communication of intuitonistic logic

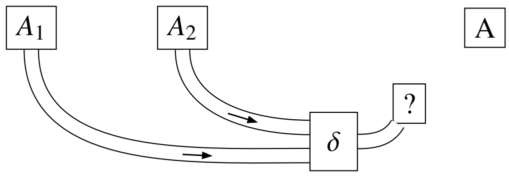
Intuitonistic proofs, and lambda terms, are seen as oriented. That is, the way the hypotheses flow through the computing machine is constrained and can only happen in one direction. This orientation is mandatory due to the asymmetry of the system. For instance, let us remind the proposition:

If A is true then (A is true and A is true)

The proof of intuitonistic logic associated duplicates the left resource, where the duplicator operator is written δ in the following diagram:



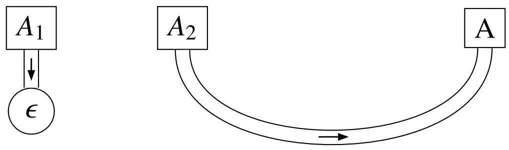
This duplication prevents it from being symmetric. For instance, let us consider the reverse direction, and suppose that two different hypotheses are coming simultaneously:



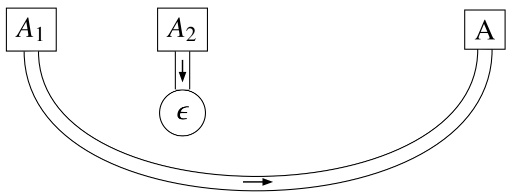
Then a proof would need to make a choice. This choice is hard-coded in any proof of the reverse direction. That is, the diagrams corresponding to the proofs of the reverse directions:

(A is true and A is true) entails A is true

are of the forms:



or:

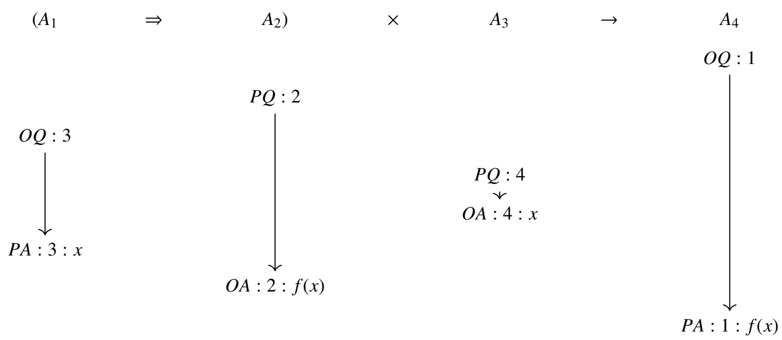


where the proof has to make a choice between the two hypotheses, while disregarding the other (the disregard operation is written ϵ in the diagram). This ability of proofs to disregard hypotheses is also part of what causes proofs to be asymmetric. For instance, there is a proof of the proposition "(A is true and B is true) entails B is true" and this proof consists, basically, in forgetting about B. However, the reverse direction A is true entails (A is true and B is true) is wrong since it would correspond to a proof that invents B.

To sum up, the computation of proofs consists of communications, and these communications are directed. Therefore, most models of logic/programming language perform a precise modelling of the progression of the proof/computation through a sequential decomposition of it in alternating steps, corresponding to exchanges of hypotheses/data between the proof/program and the so-called environment. We often refer to the protagonists as *Proponent* for the proof/program, and *Opponent* for the environment. The proof, or the program, is going to communicate with its environment through a rigorously constrained language, that consists of a trade of questions and answers. A question can be exchanged for an answer, but not the other way around, making the system asymmetric. Looking at it in terms of goods as in the previous section, it means that the exchange money - good has to come from the buyer: one cannot force someone to buy your goods. Therefore, just as an exchange money - good can happen only if the buyer is willing to pay, an exchange question - answer can only happen if one desires to ask a question. This notion of dialogue is at the heart of the current models of programming languages. For instance, the program P of type $((A \Rightarrow A) \times A) \Rightarrow A$, where A be any atomic type variable, defined by $P : (f, x) \mapsto f(x)$ is modelled though the set of lookalike conversations:

- Opponent - Question 1: What is the result of the program ?
- Proponent - Question 2: What is the result of the function f you gave me in input.
- Opponent - Question 3: For what input ?
- Proponent : Question 4: What is your second argument ?
- Opponent : Answer to question 4: My second argument is x
- Proponent - Answer to question 3: x
- Opponent - Answer to question 2: f(x)
- Proponent - Answer to question 1: f(x)

represented as follows:



Let us note that the program establishes a communication from A_2 to A_4 and from A_3 to A_1, through a copy-cat behaviour, that consists in copying the questions of opponent from A_4/A_1 into A_2/A_1, and replicating its answers the other way around. On the other hand, opponent has an almost similar behaviour from A_2 to A_1. We will re-use this example in the next paragraph, representing the program as a diagram as above.

To conclude this section, at the heart of the computation of intuitonistic proofs lies communication. Due to the asymmetrical nature of proofs, this communication is directed.

1.1.6 Classical linear logic

As linear logic is obtained through intuitonistic logic, the computations we get are naturally oriented. However, this built-in orientation is no longer necessary. Indeed, the asymetrical nature of intuitonistic logic, that was due to the capacity of the proof to disregard, or duplicate hypotheses, has been removed. Therefore, we can decide to forget about the direction of the communication, and consider that they might go both directions. We shall explain it in term of resources. The proposition (3) above means that a pain au chocolat is exactly worth 10 cents, otherwise, you would get a proposition like "With 10 cents, one can buy a pain au chocolat and gets x cents remaining", with x being different than 0. As a pain au chocolat is worth exactly 10 cents, one can actually sell one for 10 cents. Therefore, there are two directions: buying and selling, and a symmetry between the two. Reversing the direction is embodied by the negation, and the symmetry by the involution of the negation $\neg\neg A = A$, just as in classical logic.

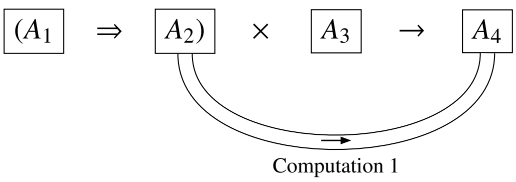
This choice to drop the direction highlights the true nature of classical linear logic. **Intuitonistic proofs compute by communicating through channels. Classical linear logic is the logic of the communication through these channels.**

This has major consequences on the way to think about the computational aspect of classical linear logic, since it shifts the focus away from the nature of intuitonistic computation, to shed light on the result. For instance, let us consider the following steps of actions:

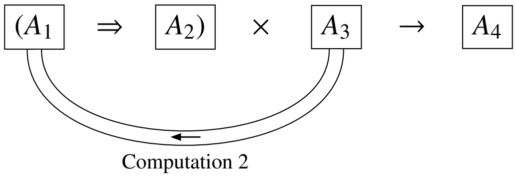
1. I buy you a paint au chocolat with 10 cents.
2. I sell you back your paint au chocolat for the 10 cents I just gave you.

Then, these steps have a computational flavour, that is, they correspond to a two-step process in a computing machine. Calling C the proposition "having 10 cents and no pain au chocolat", $\neg C$ can be understood as the reverse " Having a pain au chocolat and 0 cents". The global principle of our example is that from C, you can go to $\neg C$ using a one-step computation, that corresponds to the exchange. That is, **negation encodes computation**. This is what is done in step 1. In the second step, we go from $\neg C$ to $\neg\neg C$. However, from the linear logic point of view, as $\neg\neg C = C$, doing two computations is interpreted just as doing nothing. Of course, in terms of results, we end up in the same state as at the beginning. However, the computing machine has just done two computations, that we intentionally disregard. That is, by looking at proofs of intuitonistic logic from a classical point of view, we forget the computations that create the channels of communication, to only keep in mind the latter.

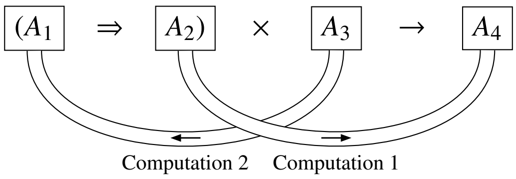
For instance, let us re-examine our program $(f, x) \mapsto f(x)$ above. The moves $OQ : 1, PQ : 2, OA : 2, PA : 1$ encode a communication from A_2 to A_4.



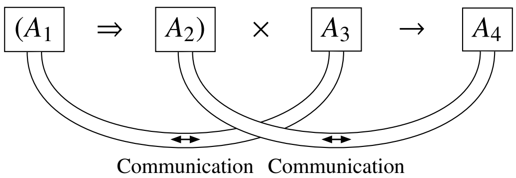
On the other hand, the moves $(OQ : 3, PQ : 4, OA : 4, PA : 3)$ encode a computation from A_3 to A_1.



Leading to the final representation of the program:



Its corresponding proof in classical linear logic forgets both the order in which the computations happen, and the directions of the communications.



Despite having, like classical logic, an involutive negation, linear logic enjoys having non-trivial mathematical models. That is, there is a suitable notion of computation underneath it. Quite surprisingly, there is no simple language to express it. It was not until recently (2012) that a consistent formal system was presented [92] [19], whose terms could be seen as encoding proofs of linear logic. However this one is, with reservation, counter-intuitive. On the other hand, this enables us to talk about the computational extent of proofs without concerns.

A prominent feature of the computational aspect of linear logic is its concurrency. Getting a firm grip on the handling of resources reveals the truly concurrent nature of the intuitonistic proofs. For instance, in the proof of "A true entails A true and A true", once the duplicating happens, the two resources flow concurrently throughout their respective channels. Similarly, the classical linear logic view of the program described above reveals two different channels of communication through which computations can happen concurrently. One of the main rationale behind the study of classical linear logic is to have a logic of concurrent programs, that would enable us to ensure the soundness of programs running concurrently.

1.2 General introduction

1.2.1 Denotational semantics of linear logic

This thesis is concerned about denotational semantics of classical linear logic, that consists in designing mathematical models of its proofs. We restrain ourselves to the "perfect" (as called by Girard [37]) fragment without exponential, called Multiplicative Additive Linear Logic, abbreviated MALL. We refer to MLL for its multiplicative sub-fragment. We aim for a perfect modelling, where the elements of the model and the proofs are in correspondence. This way, one can see each element of the model as the exact abstract representation of a proof. This has been the subject of a lot of research, that unfortunately fell short of the ultimate goal. The intent of this dissertation is to obtain a perfect modelling of proofs of fragments of linear logic, while relying on similar recipes as current models of programming languages. To start, let us highlight the main motivations behind this research. First, having such a model would give us a precious insight about the computational nature of proofs, and would help us understand the calculus of resources that forms the backbone of the lambda-calculus. The second rationale is to have a model that could potentially forms the basis of a fully abstract model of a programming language derived from the calculus of linear logic proofs, such as session types [92, 19].

The most famous semantics for linear logic is surely the coherence spaces [39]. Coherence spaces first emerged as a semantics for the lambda-calculus, notably through their strong relationship with stable functions. As a model, they provide a key insight into the nature of the intuitonistic negation, highlighting that it can be divided into two connectives: a *pure* negation followed by a modality. This discovery originated classical linear logic [33], by focussing on the pure fragment. This approach led to the definition of the semantics prior to the definition of the logic, that was designed accordingly.

Since that pioneering work, the search for a better model of linear logic began, which could establish a perfect, syntax-free model. This research was undoubtedly fruitful, producing various models that were successfully re-used in the sibling field of semantics for programming languages. The flagship of these being certainly the games that provided the first fully complete models of multiplicative linear logic [5], being a great many times refined or completed for other fragments [50, 10, 66], and being reused for constructing fully abstract models of programming languages [49, 7, 80, 43, 45], those being versatile enough to incorporate effects [9, 57, 76, 56] and even model mainstream programming languages [78]. On the other hand, this research motivated plenty of static models, such as the hypercoherences [27], Chu-spaces [24, 20], double-glued categories [53] or even relations that are seemingly similar to games [51].

This thesis is in line with these previous works, however taking a different stance on atomic variables. Current models of logic rely on 2-categorical techniques to model atomic variables and parametricity, following technologies originally introduced for system F [13]. Consequently, these greatly differ in nature to the fully abstract models of programming languages, where the model has to precisely characterise the nature of computation, while dealing with fixed base objects possessing a generally truly simple structure, such as the booleans. The goal of this research is to shift to a 1-categorical structure while keeping a perfect modelling.

1.2.2 Nominal models

Nominal sets were first introduced in the 1930s in order to create a model of set-theory not satisfying the axiom of choice. They re-appeared in the 1990s as a means to provide a clean formalism to deal with names in computer science and notably to binding or α-equivalence. Names are widespread throughout computer science, being especially prevalent in the presentations of formal languages, such as the lambda or pi-calculus. Since then, their utilisation has been generalised in programming languages to encompass any kind of resources or methods, such as exceptions, channels, threads, or references. Names are atomic entities that are available in infinite quantity, and can be freely passed around or generated. The names we use on this thesis are sorted. That is, each name belongs to a certain kind, and all names belonging to the same kind are equivalent, that is, they can only be compared for equality.

The paradigm of names as resources has been used extensively in games for providing semantics of effectful languages, being a key ingredient to provide a clean presentation of effects such as references [89], exceptions [79], or polymorphism [41]. On the other hand, names also appear as a key ingredient in modelling proofs via terms through the Curry-Howard isomorphism. For instance, the identity is mapped to the alpha-equivalence class of terms $\lambda x.x$, the formula on the left of the sequent being interpreted as a name x that is bound under λ to be passed as a variable to the right hand side. However, the modelling of computation flow has, to the best of the author's knowledge, not been modelled nominally when non-syntactical models have been considered. More precisely, names have so far not been used to model the linear use of resources in linear logic, and this is what this thesis is targeting.

As of today, models of logics with atomic variables are all built using di-natural transformations [10, 16, 60], with the intuition that a proof should behave uniformly at variable types [13]. In this work, the uniformity condition is encoded thanks to nominal techniques. Each literal is seen as a pool of resources of a certain kind, that we encode with names. The names are given a polarity following the polarity of the literal. Thus, the denotation of a proof is an equivariant element, dealing with the names in a blind manner, and establishing links between negative and positive literals. Within this setting, equality between names of same polarity is irrelevant, and we sometimes rely on substitutions to prevent the proof from acting on it.

1.2.3 Full completeness

When are two proofs the same? To answer this question, one must define a notion of equivalence between proofs, relating proofs that differ only by syntactical considerations, but are essentially the same. We denote by $[\![.]\!]$ the denotation function from formulas and proofs to the mathematical model. The basic property we expect from our models is soundness, meaning that the denotation function respects the equivalence of proofs; two proofs that are essentially equal have same denotation:

$$\text{soundness: } \pi \simeq \pi' \;\Rightarrow\; [\![\pi]\!] = [\![\pi']\!]$$

In this thesis, we will consider only categorical models, causing the denotation function $[\![.]\!]$ to be a functor. Full completeness has been introduced in [5] and was described as "the tightest possible connection between syntax and semantics". A categorical model is fully complete if the denotational functor is full. In some sense, this is the analogue of definability in models of programming languages. Writing $\mathcal{M}$ for the model:

$$\text{Full completeness: } \forall m \in \mathcal{M}(A, B).\exists \pi : A \vdash B.\ [\![\pi]\!] = m.$$

In [5], Abramsky and Jagadeesan proposed an even tighter connection, "One may even ask for there to be a unique cut-free such proof (π such that $[\![\pi]\!] = m$), i.e. that the above functor be faithful". In such models, the denotation functor defines an equivalence of categories, mapping each class of equivalent proofs to a different element. Within such models, the morphisms are perfect abstract representations of the essence of proofs.

However, such models are infamously hard to design. As of today, there is no known such model for the whole linear logic, but some have been obtained for fragments. Their number to date is rather limited, perhaps because a good characterisation of proofs up to equivalence is still an obstacle to overcome. The difficulty might also be related to our poor understanding of the computational extent of the proofs of classical linear logic.

1.2.4 A trick, tensorial logic

The quest for fully-complete models of linear logic led many authors to rely on a realisability criterion to define models. As explained above, linear logic might be seen as arising from intuitonisitic computations by forgetting about direction and order. Therefore, several attempts relied on models of computation, often games, followed by a forgetful operation. In general, this consisted in looking at the positions reached while forgetting about the plays leading to them. For instance, the first full-completeness result for MALL was motivated by concurrent games [10], and the first and only full-completeness result for the whole logic by asynchronous games [66].

The fact that a full completeness result was obtained through an intermediate game model meant that the framework (hence, the games) was the perfect dynamic counterpart of linear logic. In other terms, they form an ideal model of the "intuitonistic computations", whose communications flawlessly account for the channels described by linear logic. Therefore, it was undertaken in [71] to unravel the logic and proof-system corresponding to those asynchronous games.

The result, tensorial logic, is deceptively simple. To summarise, it is as classical linear logic, except that the negation is not involutive anymore. This leads to an intuitonistic logic (only one formula on the right-hand side of the sequent) called tensorial logic, whose proofs are akin to focalised proofs of linear logic. Tensorial logic can also be described as a restricted sub-system of intuitonistic linear logic, whose constraints induce a better behaved notion of polarity. Several translations between classical linear logic and tensorial logic have been reported [71], most being analogous to translations between intuitonistic and classical logic.

Consequently, to obtain a full-completeness for linear logic, one can focus on first establishing a full-completeness result for tensorial logic, and then rely on one of the translations to provide a fully-complete result for linear logic. This is, how, implicitly, was established the full-completeness result of linear logic in [66].

1.2.5 Historical perspective

The problem of constructing mathematical abstractions of proofs of linear logic has been widely studied. At least four distinct directions were taken to tackle this problem.

- Proof structures try to abstract away the bureaucracy of syntax when constructing proofs by seeing them as graphs, and then rely on geometric or algebraic considerations to characterise those graphs that are valid, that are, denotations of proofs. Proof nets were introduced in the seminal article of linear logic [33], and were undeniably very successful for multiplicate linear logic without units [23]. They now have been extended to cover the additives [36] [46]. The handling of the multiplicative units is quite difficult [48], and the additive units still remain out of the scope of those structures.
- Geomertry of interaction originally consisted in looking at proofs as operators of linear algebra [38]. Since that, much work has been done, generalising the original idea to cover a wide range of different models [6]. It was notably given a combinatorial approach, enabling the construction of models for full linear logic [85].
- Ludics [35] was born as an attempt to somehow reconciliate polarities (encompassed by the interactive nature of logic), syntax and semantics within one single framework. As such, it cannot be really considered as a model of linear logic, but a correspondence was proven nonetheless.
- Denotational semantics, which we expand more on below.

Within denotational semantics, one must again consider two traditions. Inline with the inception of linear logic as the logic for coherent spaces [33], that form a sub-category of the category of sets and relations, a first class of models arise as refinements of the relational model. Those are deemed static. Distinguished among them are the hypercoherence spaces [27], the totality spaces [60], or the Chu-spaces [20]. Notably, hypercoherences have lead to a full-completeness result for MALL without units [16], whereas Chu-spaces, coherence spaces and totality spaces can be the ingredients of fully complete models of MLL without units [24, 60, 83]. One might complete this list by adding vector spaces, lattices, posets and relations, logical relations... Hyland and Schalk review and present a wealth of such models in [53].

On the other hand, games have emerged as a powerful tool to interpret semantically linear logic. At their core live strategies, that produce results only through interacting with a peer protagonist, called Opponent. Therefore, those models can be seen as dynamics. The use of games for logics goes back to the work of Lorenzen [62, 61], or even Gentzen (although in different terms). However, games for linear logic were pioneered by Blass [15], though he did not succeed in organising them in a category. Abramsky and Jagadeesan were the first to notice that games could be used for full-completeness although the original paper dealt with unitless MLL only and the games were permissive of the MIX-rule [5]. However, these were shortly refined afterwards by Hyland and Ong to remove the MIX-rule [50]. The incorporation of additives was quite challenging, since sequential games are inherently unable to form a category having both products and coproducts, as highlighted by the Blass problem. A solution was found through concurrent games by Abramsky and Melliès[10], that circumvented the problem and led to a full-completeness result. However, these relied on a realisability technique that was introduced by Baillot, Danos, Ehrhard and Regnier [12], consisting in forgetting about "time", to project games on relations.

The concurrent games shed light to a very peculiar property: they enjoy being both static and dynamic. This was at the heart of the line of research developed by Méllies on asynchronous games [63, 64, 65, 66, 70] who redefined innocence as a positional property, and highlighted the static, and concurrent nature of the lambda-calculus. Relying on these results, Méllies was able to prove a full-completeness result for linear logic. Dually, this gave born to tensorial logic [71].

1.3 Contributions and thesis outline

The purpose of this thesis is to design models of linear logic with type variables while staying within the realm of first-order category theory. The pinnacle of this work is the presentation, in Chapter 7 of a model of MALL, not relying on 2-categorical tools nor quotient, that is fully-complete for MALL without units. This thesis is divided in 3 parts. The first is devoted to background material used throughout the thesis on logic and static nominal models. The second revisits the work on asynchronous games and tensorial logic through a nominal perspective. The main source of inspiration of the third are the concurrent games, though we quickly shift to introducing new models. We present a more detailed outline of the chapters and their contributions below.

The first Chapter is the current introduction.

The second Chapter 2 is devoted to recalling some general results concerning linear logic and tensorial logic. We notably present the focalised proof systems for them, and a translation between the two. Most of the material presented during this chapter originates from the literature, and our contributions are minimal. It often consists in clarifying and exposing properties that could be considered as folklore. We introduce a notion of (global) focalised sequent calculus for both linear and tensorial logic, present the multiplicative tensorial lambda calculus, expose the proof of the translation between tensorial logic and linear logic. Also, we exhibit all the equivalences that proofs of MALL enjoy, together with those for multiplicate additive tensorial logic. This allows us to define a notion of normal form for proofs of tensorial logic, that will be pivotal for our future proof of full completeness.

In the third chapter 3, we give a brief exposition of how nominal sets can be used to provide representations of various linear categories. We start with monoidal categories, and show how the usual tracing amounts to the use of substitutions within this setting. Based on this, we display a category of nominal polarised relations, that forms a compact-closed category, and show how one can decorate them with hypercoherences. The use of names refines the relational model by giving an intensional content to the atoms it deals with, and allows for a precise formalisation of the notion of resources. Though the nominal treatment of atomic variables allows us to enforce linearity, and the hypercoherence structure prevents us from having a degenerate compact closed category, we conclude this chapter by noticing that the final model we obtain is not fully complete.

The purpose of the second part is to refine the previous model by precisely characterising relations that arise as denotations of proofs. In order to do so, we rely on the connection between linear logic and tensorial logic, and the full-completeness result established for asynchronous games for tensorial logic [69]. Therefore, we introduce nominal asynchronous game semantics, by recasting the whole work achieved by Melliès in [66, 69, 64] within the nominal framework. This achieves the following contributions. First, it shortens the gap between syntactical and mathematical models, by making the strategies and the terms live within the same universe, allowing one to establish a nominal correspondence between the plays and the Böhm trees. Second, it advocates the vision of type variables as typed resources, and emphasises the ability of names to model them. Third, it extends the previous fully complete game model of linear logic [66] by adding type-variables to it. Furthermore, this method allows us to deal with them while staying within first-order category theory. The second part is divided between chapters 4,5, 6, that we present shortly below.

In Chapter 4, we present the nominal structures underlying the nominal games. These structures allow us to define the nominal arenas. We expose informally how these are related to Böhm trees. In particular, those shed light on the handling of names one should expect within the strategies. In Chapter 5 we define the nominal strategies for tensorial logic. Finally, in Chapter 6 we prove that the categorical model thus formed is sound for tensorial logic, and establish full completeness. We then project strategies onto nominal relations and prove that the model thus obtained is fully-compete for MALL.

The third part consists of a single Chapter 7, bringing attention to a different model. If the previous part led to a fully-complete model of MALL, it was through an intermediate medium. This part is devoted to forget about this medium to obtain a direct characterisation. The model we rely on has been discovered through a careful analysis of the concurrent games model of Abramsky and Melliès [10]. These have the particularity to be simultaneously static and dynamic. Notably, they can be seen as a special kind of relation, and compose as such. The original model is strong enough to only prevent directed cycles in the proof structures, the full completeness being achieved through clever tricks using dinatural transformations. As we drop them, we have to strengthen the conditions to get rid of undirected cycles, and impose connectedness. This is done partially using ideas relating games and static models, especially those developed by Hyland and Schalk [51, 53, 52]. This model addresses two issues that current fully complete models have: it tackles the use of 2-categorical tools, whilst avoiding relying on a quotient. In a second section, we enrich our model with hypercoherences. Those allow us to replace the extensional content of concurrent games in a static manner and enables us, following a method devised by Blute, Hamano and Scott [16] to achieve full completeness for MALL without units.

Chapter 2

Linear Logic, Tensorial Logic, and their Models

This short chapter briefly introduces some background material on logic, which will be used throughout the thesis. This thesis is mainly concerned with linear logic, but tensorial logic will be used as an intermediate logic to achieve the desired results. The next section describes a particular translation between these two logics, the focalised one.

2.1 Introduction

2.1.1 Logic

A logic $\mathcal{L}$ comprises of a set formulas, given through an explicit grammar, called syntax, together with a proof system, that allows us to establish propositions that are deemed provable.

The logics we introduce along this thesis all rely on the same recipe. The building blocks of the syntax consist of an enumerable set of **atomic variables** TVar, that we may also refer to as type variables, or propositional variables, and a finite set of **constants** . These come together with a set of binary or unary **connectives**, sometimes relabelled operations, that allow us to form new elements. A **formula**, or **proposition** of $\mathcal{L}$ is an element of the syntax. Each constant introduced acts as a unit for one of the operations. For instance, the binary operation $\otimes$, called tensor, will have a constant named I as a unit. That is, $A \otimes I$ and A will be isomorphic from a proof theory point of view: a proof of one can be turned into a proof of the other. The only unary operation introduced will be the **negation**, denoted $(.)^{\perp}$ or $\neg$. A **literal** is an atomic variable or its negation . An **atomic formula** is either a constant or an atomic variable.

The proof systems presented along this thesis will all be different variations of sequent calculi. A sequent can be either one-sided, or two-sided. A **two-sided sequent** is an ordered list of two finite sequences of formulas, $((F_1, ..., F_n), (G_1, ..., G_m))$, and will be written $F_1, .., F_n \vdash$

$G_1, .., G_m$. The F_i are referred to as hypotheses, and the G_i as conclusions. We will commonly write $\Gamma, \Delta, ..$ for sequences of formulas, and write $\Gamma, \Delta, G_1, .., G_m$ for the concatenated sequence $\Gamma.\Delta.G_1. \ldots .G_m$. A **one-sided sequent** is a two-sided sequent whose first sequence is empty, and will be denoted $\vdash \Gamma$.

A **sequent calculus system** is a set of rules, that take as input 0 or a finite number of sequents, and lead to a single sequent as output. The number of inputs of a given rule will be referred to as its **arity**. A **proof** of $\Gamma \vdash \Delta$ is a tree (by respect to mother-nature, we consider that the root is at the bottom of the tree, and the leaves at the top) such that each leaf of the tree is a rule of arity 0 in the sequent calculus, each node corresponds to the application of a rule, and the root of the tree is $\Gamma \vdash \Delta$. Proofs will be denoted by π and variants. We write $\pi : \Gamma \vdash \Delta$ to emphasise that π is a proof of $\Gamma \vdash \Delta$. The rules of the sequent calculus allow us to combine proofs together to provide proofs of new sequents. For instance, we present below an example of a rule that introduces the binary connective $\otimes$:

$$\frac{\Gamma \vdash \Delta, A \qquad \Gamma' \vdash \Delta', B}{\Gamma, \Gamma' \vdash \Delta, \Delta', A \otimes B} \otimes :$$

A **fragment** of a logic $\mathcal{L}$ is a restriction of the syntax and of the set of rules, such that this restriction still defines a logic. That is, a rule in the restricted sequent system cannot lead to a formula out of the restricted syntax.

Among the rules of arity different than 0, all but one, named cut, will satisfy the *sub-formula* property. This property roughly states that each formula in the input sequents will be present in the output sequent, either as an element of the sequent or as a sub-formula, and reversely: each formula in the output sequent comes from a combination of formulas from the input sequents. If every rule of arity 0 introduces atomic formulas only, this entails that every atomic sub-formula in the conclusion must have been introduced by a rule at a leaf of the proof-tree, and conversely. This property allows us to derive the consistency of logic for the fragment without cut: if there was a proof of falsehood $\pi :\vdash \bot$, then the only possible case is if this one came as a rule of arity 0. So we simply need to ensure that there is no rule of arity 0 that introduces solely the false statement.

The cut-rule is a rule of the shape :

$$\text{Cut} : \frac{\Gamma \vdash \Delta, A \qquad \Gamma', A \vdash \Delta}{\Gamma, \Gamma' \vdash \Delta, \Delta'}$$

for two-sided sequents, or

$$\text{Cut} : \frac{\vdash \Gamma, A \qquad \vdash A^{\perp}, \Delta}{\vdash \Gamma, \Delta}$$

for one-sided sequents. It allows us to "compose" proofs through A. All the logics introduced satisfy **cut-elimination**. That is, there is an algorithm that, given a proof with some cut-rules, removes the cuts while still resulting in a proof of the same sequent. It allows us to conclude about the consistency of the whole logic. Furthermore, this enables us to define a notion of composition of cut-free proofs, where the composition is defined by applying the cut rule followed by the cut-elimination algorithm. Finally, for each formula A, there is a canonical proof $A \vdash A$, referred to as id_A. Therefore, the proofs seemingly behave as a category.

2.1.1.1 Invariants of proofs

However, due to the "bureaucracy of syntax" (term coined by Girard in [38]), proofs do not generally form a category. For instance, when applying a cut-elimination algorithm, one may find that $\pi;_{cut}\mathsf{id} \neq \pi$, or equivalent problematic cases, such as a non-associative composition, due to slight syntactic differences. Therefore, we introduce a notion of invariant, that behaves well with regard to the rules of logic, and notably the cut. This means that:

- Writing π for a proof with cuts, and π' for any proof obtained by applying (partially or totally) the cut-elimination procedure to π, then the invariant of π and π' must coincide.
- A proof π and the result of applying the cut elimination to $\pi;_{cut}\mathsf{id}$ or $\mathsf{id};_{cut}\pi$ must have same invariant.
- The results of applying the cut-elimination to $(\pi_1;_{cut}\pi_2);_{cut}\pi_3$ and $\pi_1;_{cut}(\pi_2;_{cut}\pi_3)$ have same invariant.

Equivalently, one can speak about equivalence: two proofs are equivalent if and only if they have same invariant. We write $[.]$ for the function that maps a proof to its invariant, or, analogously, to its equivalence class. We denote $\sim$ for the relation of equivalence between proofs. Of course, two proofs are equivalent only if they have equal conclusion. Furthermore, one needs the invariants to be modular: given two proofs $\pi :\vdash \Gamma, A$ and $\pi' :\vdash A^{\perp}, \Delta$, then one can deduce directly from the invariants of π, π' what is the invariant of cutting π against π'. Then the invariants of proofs organise themselves as a (multi)-category. Of course, it might be that depending on the cut-elimination procedure we choose, we obtain different notions of invariant and equivalence.

Similarly, one expects the invariants to behave well with regard to the other rules, such as the $\otimes$ introduced above, especially from a categorical point of view. For instance, the tensor product might behave *almost* like a monoidal product when it comes to proofs, but not quite. Generally, one should add some more quotienting in order to get a well-behaved tensor. For more on that, we refer to [67].

The proof invariants are such that proofs whose rules order differ only "in minor details", without significant structural differences, have same invariant. Two proofs are equivalent if they are equal up to insignificant syntactic considerations. Finally, one considers that all proofs that are deemed equivalent, from a syntactic point of view, are indeed, equivalent from a proof-theoretic point of view. That is, the quotient only gets rid of bureaucracy, and reveals the intrinsic nature of proofs.

If one could argue that the equivalence chosen is not adequate, for instance that we consider two proofs as similar whereas they are intrinsically different, it seems that a general consensus has been achieved within the community for the logics presented in this thesis. It is the same consensus that allows us to speak, for instance, about *the* categorical semantics of linear logic, and not *a*. Even if there now is a standard model, it does not prevent us from exploring what would happen if we relax some of the equivalences. For instance, if we consider that two exchange rules do not necessarily commute:

$$\dfrac{\dfrac{\vdash \Gamma, A, B}{\vdash \Gamma, B, A}\text{Exchange}}{\vdash \Gamma, A, B}\text{Exchange} \quad \not\sim \quad \vdash \Gamma, A, B$$

then the model fundamentally differs. In that case, the category of invariants will no longer be a symmetric monoidal category but a braided category.

2.1.1.2 Denotational semantics

Denotational semantics is the art of mapping proofs into a mathematical model $\mathcal{M}$, and such that the operations on proofs translate well as operations on the model. Once the mathematical model is clearly defined, we will denote $[\![.]\!]$ the function from proofs and formulas to it. For instance, given two proofs $\pi_A :\vdash A$ and $\pi_B :\vdash B$, one can form a new proof $\pi_A \otimes \pi_B : A \otimes B$. We expect this operation to lift to the model: $[\![\pi_A \otimes \pi_B]\!] = [\![\pi_A]\!] \otimes_{\mathsf{M}} [\![\pi_B]\!]$. Similarly, the function must satisfy $[\![\pi_1;_{\mathsf{cut}}\pi_2]\!] = [\![\pi_1]\!];_{\mathcal{M}}[\![\pi_2]\!]$. Therefore, it seems appropriate to consider that $\mathcal{M}$ forms a category.

Formally, given a category C, we introduce the denotation function $[\![.]\!]_C$ from $\mathcal{L}$- formulas and proofs to C, where $\mathcal{L}$ is a given logic. Generally, C and $\mathcal{L}$ will be clear from the context, and we will simply write $[\![.]\!]$. As explained before, in order for $[\![.]\!]$ to be a functor, the proofs have to be considered up to equivalence. C is a model of $\mathcal{L}$ if $[\![.]\!]$ factorises through $[.]$, that is, given two proofs $\pi, \pi' : A \to B$, $\pi \sim \pi' \Rightarrow [\![\pi]\!] = [\![\pi']\!]$. Furthermore, we expect $[\![.]\!]$ to behave nicely with regards to the rules of the sequent calculus. Therefore, the rules of the sequent calculus lift to additive structure at the categorical level. For instance, the $\otimes$ rule implies the monoidality of the category.

Formally, the denotation of a proof $\pi : F_1, ..., F_n \vdash G_1, .., G_m$ will be a map $[\![F_1]\!]\Box...\Box[\![F_n]\!] \to [\![G_1]\!]\Diamond....\Diamond\,[\![G_m]\!]$, where $\Box$ and $\Diamond$ are associative functors : $C \times C \to C$.

Given a logic $\mathcal{L}$ (that satisfies cut-elimination), together with an appropriate notion of equivalence, one can devise what categorical structure S precisely corresponds to it, that is, the models of $\mathcal{L}$ are exactly those with the S-structure.

2.2 A brief introduction to linear logic

Since its initial discovery in the 1980s [33], linear logic has been widely studied and used to reason about programming languages. Following its inception, a translation of intuitionistic logic into it has been established, that was one of the keys to understand the semantics of programming languages, and a step towards the first fully complete model of PCF [7, 49]. This initial translation has been extended to others logics (classical logic, affine logic...) and paradigms (call-by-name / call-by value), establishing linear logic as a central tool in computer science. By its ability of subsuming both intuitionistic and classical logic, it led the original author, Girard, to ask whether there was finally a unique structural logic [34] for computer science, and that linear logic was, ultimately, the most primitive one [34]. In this thesis, we do not focus on the logical aspect, but on the denotational side. Therefore, we will keep short the presentation of linear logic.

As we will deal with neither the exponential nor the polymorphic part, we will not include them in our presentation. The remaining subset is often referred to as Multiplicative Additive Linear Logic, or MALL, and is perceived as being the heart of the whole system. The key aspect of linear logic is its resource awareness: each hypothesis can be used only once, and has to been used once. Often, one draws an analogy between hypotheses and resources. In a proof of linear logic, each resource has to be consumed exactly once to produce the conclusion, just as, for instance, in a chemical equation. This leads to the distinction between two kinds of binary connectives: the multiplicatives, where the two resources are either used or produce; and the additives, where a choice between the two resources is made. We hence obtain 4 binary connectors. Two "or"s, the multiplicative ($\parr$) (referred to as parr) and additive one ($\oplus$), together with two "and"s, the multiplicative ($\otimes$) and additive ($\&$) one. Together with them come four units $\bot, 0, 1, \top$. We also consider an enumerable set of atomic variables $\mathsf{TVAR} = X, Y, ...$, and a negation $(.)^\perp$ obeying De Morgan equations, produced below. Finally, the formulas of MALL are defined by recursion as follows:

$$F \ ::= \ \bot \ | \ 0 \ | \ 1 \ | \ \top \ | \ X \in \mathsf{TVar} \ | \ F \parr F \ | \ F \oplus F \ | F \otimes F \ | \ F \,\&\, F \ | \ (F)^\perp .$$

We shall use F, A, B, C and variants to range over formulas. The negation obeys the following equalities :

$$\begin{array}{ll}
(X^\perp)^\perp = X & . \\
1^\perp = \bot & \bot^\perp = 1 \\
\top^\perp = 0 & 0^\perp = \top \\
(A \otimes B)^\perp = A^\perp \parr B^\perp & (A \parr B)^\perp = A^\perp \otimes B^\perp \\
(A \,\&\, B)^\perp = A^\perp \oplus B^\perp & (A \oplus B)^\perp = A^\perp \,\&\, B^\perp .
\end{array}$$

We say that negation is **involutive**, that is, $((.)^\perp)^\perp = \mathsf{id}$. Also, in the sequel, we will write $A \multimap B$ for $A^\perp \parr B$.

Figure 2.1: Sequent Calculus for MALL

$$\text{Axiom}: \frac{}{\vdash A^{\perp}, A} \qquad \text{Cut}: \frac{\vdash \Gamma, A \qquad \vdash A, \Delta}{\vdash \Gamma, \Delta} \qquad \text{Exchange}: \frac{\vdash \Gamma_1, A, B, \Gamma_2}{\vdash \Gamma_1, B, A, \Gamma_2}$$

$$\text{Par}: \frac{\vdash \Gamma_1, A, B, \Gamma_2}{\vdash \Gamma_1, A \parr B, \Gamma_2} \qquad \text{Tensor}: \frac{\vdash \Gamma, A \qquad \vdash \Delta, B}{\vdash \Gamma, \Delta \vdash A \otimes B}$$

$$1: \frac{}{\vdash 1} \qquad \perp \frac{\vdash \Gamma}{\vdash \Gamma, \perp} \qquad \top: \frac{}{\vdash \Gamma, \top}$$

$$\oplus 1: \frac{\vdash \Gamma, A}{\vdash \Gamma, A \oplus B} \qquad \oplus 2: \frac{\Gamma \vdash B}{\vdash \Gamma, A \oplus B} \qquad \&: \frac{\vdash \Gamma, A \qquad \vdash \Gamma, B}{\vdash \Gamma, A \& B}$$

We present the sequent calculus of MALL in figure 2.1.

Given a proof with some cuts in it, there is an algorithm, often referred to as "cut-elimination procedure" that transforms it into a proof with same conclusion, but without cuts. This topic exceeds the scope of the introduction, but such an algorithm has been presented in [67] for instance. As a proof without cut obeys the sub-formula property, one can conclude about the consistency of the MALL fragment, that is, there is no proof π such that $\pi :\vdash \perp$.

Sometimes, we might speak about an additional rule, called mix, presented below :

$$\mathsf{mix}: \frac{\vdash \Gamma \qquad \vdash \Delta}{\vdash \Gamma, \Delta}$$

To assume this supplementary rule is equivalent to, at a categorical level, having morphisms $A \otimes B \to A \parr B$. It turns out that numerous models of linear logic satisfy the mix-rule, such as for instance, the first fully-complete model [5], or models that appear as refinement of the relational model, such as the coherence [33] or hypercoherence models [27]. This is perceived, along this thesis, as a defect that should be corrected.

There is an alternative presentation of MALL, where the sequents are two-sided, and where to each rule on the right-hand-side corresponds a rule on the left-hand-side. To avoid the doubling, we consider a slightly different alternative but equivalent presentation. We add to the above sequent calculus the two following rules:

$$\text{Left negation}: \frac{\Gamma \vdash A, \Delta}{\Gamma, A^{\perp} \vdash \Delta} \qquad \text{Right negation}: \frac{\Gamma, A \vdash \Delta}{\Gamma, \vdash A^{\perp}, \Delta}$$

Considering a two-sided sequent calculus is useful for the denotational presentation. Given a proof of a one-sided sequent, it is not clear what will be the image and pre-image of the proof, that is, how to model it in a category.

Despite that, in the sequel, we will only focus on the one-sided sequent fragment. This fragment is large enough for a comprehensive study of the logic, and easier to deal with as there is no duplication of the rules. Indeed, to any two-sided sequent can be assigned a one-sided sequent by applying the right negation to every formula on the left-hand part of the sequent. In the sequel, we will also not pay attention to the exchange rule, that commutes with absolutely every rule. Somehow, we look at sequents just as if they were finite multisets, that are, sets where each element may appear a finite number of times.

Along this thesis, we will sometime speak about multiplicative linear logic, often refereed to as MLL. It is the fragment of MALL formed from propositional variables, constants $I, \bot$, the connectives $\otimes, \parr$, and their associated rules.

2.2.1 Equivalence of proofs in linear logic

2.2.1.1 A case study

We would like the tensor $\otimes$ to define a monoidal product in linear logic. However, this leads us to impose some equivalence between proofs that appears natural.

Among them, is the equivalence between the following following proofs:

$$(1) \qquad \dfrac{\dfrac{}{\vdash \Gamma, A} \qquad \dfrac{\dfrac{}{\vdash \Delta, B} \qquad \dfrac{}{\vdash \Theta, C, D}}{\vdash \Delta, \Theta, C, B \otimes D}\otimes}{\vdash \Gamma, \Delta, \Theta, A \otimes C, B \otimes D}\otimes$$

$$\leftrightsquigarrow$$

$$(2) \qquad \dfrac{\dfrac{}{\vdash \Delta, B} \qquad \dfrac{\dfrac{}{\vdash \Gamma, A} \qquad \dfrac{}{\vdash \Theta, C, D}}{\vdash \Gamma, \Delta, A \otimes C, D}\otimes}{\vdash \Gamma, \Delta, \Theta, A \otimes C, B \otimes D}\otimes$$

This equivalence is required to make cut-elimination work well with identities. We present below a cut-elimination algorithm, and prove this equivalence using it. The cut-elimination procedure is classic, and follows from [67]. We do not display the full cut-elimination procedure, only the relevant part. The cut-elimination algorithm starts by picking one of the top cuts, and then applies the cut-elimination steps presented below. In the case of a cut where the premises are principal, the proof evolves as follows:

$$\dfrac{\dfrac{\dfrac{}{\vdash \Theta, A^\perp, B^\perp}}{\vdash \Theta, A^\perp \parr B^\perp} \qquad \dfrac{\dfrac{}{\vdash \Gamma, A} \qquad \dfrac{}{\vdash \Delta, B}}{\vdash \Delta, \Gamma, A \otimes B}}{\vdash \Theta, \Delta, \Gamma}\text{Cut}$$

$$\leftrightsquigarrow$$

$$\dfrac{\dfrac{\dfrac{}{\vdash \Theta, A^\perp, B^\perp} \qquad \dfrac{}{\vdash \Delta, B}}{\vdash \Theta, \Delta, A^\perp}\text{Cut} \qquad \dfrac{}{\vdash \Gamma, A}}{\vdash \Theta, \Delta, \Gamma}\text{Cut}\ .$$

The order between the cuts is chosen arbitrarily, but, as proven in [67], it does not matter and the order can be changed. The second case we explore is the secondary (that is, non principal) versus secondary case. We present it for $\otimes$.

$$\dfrac{\dfrac{\dfrac{}{\vdash \Gamma, A} \qquad \dfrac{}{\vdash \Delta, B, E}}{\vdash \Gamma, \Delta, A \otimes B, E} \qquad \dfrac{\dfrac{}{\vdash \Theta, C} \qquad \dfrac{}{\vdash \Sigma, D, E^{\perp}}}{\vdash \Theta, \Sigma, C \otimes D, E^{\perp}}}{\vdash \Gamma, \Delta, \Theta, \Sigma, A \otimes B, C \otimes D}\,\text{Cut}$$

In that case, two choices can be made. Either we choose to give priority to the left branch, or the right one (as we will see later, in games semantics, this precisely correspond to the case where player has to choose between two independent moves). We first display the right-first priority policy.

$$\dfrac{\dfrac{}{\vdash \Theta, C} \qquad \dfrac{\dfrac{\dfrac{}{\vdash \Gamma, A} \qquad \dfrac{}{\vdash \Delta, B, E}}{\vdash \Gamma, \Delta, A \otimes B, E}\,\otimes \qquad \dfrac{}{\vdash \Sigma, D, E^{\perp}}}{\vdash \Gamma, \Delta, \Sigma, D}\,\text{Cut}}{\vdash \Gamma, \Delta, \Theta, \Sigma, A \otimes B, C \otimes D}\,\otimes$$

We could just as well pick the left priority policy.

$$\dfrac{\dfrac{}{\vdash \Gamma, A} \qquad \dfrac{\dfrac{}{\vdash \Delta, B, E} \qquad \dfrac{\dfrac{}{\vdash \Theta, C} \qquad \dfrac{}{\vdash \Sigma, D, E^{\perp}}}{\vdash \Theta, \Sigma, C \otimes D, E^{\perp}}\,\otimes}{\vdash \Delta, \Theta, \Sigma, C \otimes D, B}\,\text{Cut}}{\vdash \Gamma, \Delta, \Theta, \Sigma, A \otimes B, C \otimes D}\,\otimes$$

So finally, consider the proof (1) as before, and suppose we use the first cut-elimination procedure as before, using the left priority policy. To simplify things, we replace Γ, A with $A^{\perp}, A$ (and similarly for B, C, D), as this will not change the scope of our remarks. We make it interact with the η expansion of the identity $A \otimes C \to A \otimes C$ on the left. As we want our proof invariants to be stable under composition with identities, the following proofs have the same invariant as the proof (1).

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash A^{\perp}, A} \qquad \dfrac{}{\vdash C^{\perp}, C}}{\vdash A^{\perp}, C^{\perp}, A \otimes C}\,\otimes}{\vdash A^{\perp} \parr C^{\perp}, A \otimes C} \qquad \dfrac{\dfrac{}{\vdash A^{\perp}, A} \qquad \dfrac{\dfrac{}{\vdash B^{\perp}, B} \qquad \dfrac{}{\vdash C^{\perp} \otimes D^{\perp}, C, D}}{\vdash B^{\perp}, C^{\perp}, \otimes D^{\perp}, B \otimes D}\,\otimes}{\vdash A^{\perp}, B^{\perp}, C^{\perp} \otimes D^{\perp}, A \otimes C, B \otimes D}\,\otimes}{\vdash A^{\perp}, B^{\perp}, C^{\perp} \otimes D^{\perp}, A \otimes C, B \otimes D}\,\text{Cut}$$

$$\leftrightsquigarrow$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash A^{\perp}, A} \qquad \dfrac{}{\vdash C^{\perp}, C}}{\vdash A^{\perp}, C^{\perp}, A \otimes C}\,\otimes \qquad \dfrac{\dfrac{}{\vdash B^{\perp}, B} \qquad \dfrac{}{\vdash C^{\perp} \otimes D^{\perp}, C, D}}{\vdash B^{\perp}, C, C^{\perp} \otimes D^{\perp}, B \otimes D}\,\otimes}{\vdash A^{\perp}, B^{\perp}, B \otimes D, C^{\perp} \otimes D^{\perp}, A \otimes C}\,\text{Cut} \qquad \dfrac{}{\vdash A^{\perp}, A}}{\vdash A^{\perp}, B^{\perp}, C^{\perp} \otimes D^{\perp}, A \otimes C, B \otimes D}\,\text{Cut}$$

$$\leftrightsquigarrow \text{ (left priority)}$$

$$\dfrac{\dfrac{\dfrac{\overline{\vdash A^\perp, A} \qquad \dfrac{\overline{\vdash C^\perp, C} \qquad \overline{\vdash C^\perp \otimes D^\perp, C, D}\;\otimes}{\vdash C^\perp \otimes D^\perp, C, D}\;\text{Cut}}{\vdash A^\perp, C^\perp \otimes D^\perp, A \otimes C, D}\;\otimes \qquad \overline{\vdash B^\perp, B}}{\vdash A^\perp, C^\perp \otimes D^\perp, A \otimes C, B \otimes D}\;\otimes \qquad \overline{\vdash A^\perp, A}}{\vdash A^\perp, C^\perp \otimes D^\perp, A \otimes C, B \otimes D}\;\text{Cut}$$

$$\leftrightsquigarrow$$

several easy steps to get rid of the cut from $A^\perp, A$

$$\leftrightsquigarrow$$

$$\dfrac{\dfrac{\overline{\vdash A^\perp, A} \qquad \overline{\vdash C^\perp \otimes D^\perp, C, D}}{\vdash A^\perp, C^\perp \otimes D^\perp, A \otimes C, D} \qquad \overline{\vdash B^\perp, B}}{\vdash A^\perp, C^\perp \otimes D^\perp, A \otimes C, B \otimes D}$$

So the cut-elimination procedure makes (1) equivalent to (2). One could argue that this comes from our cut-elimination policy. But actually, if we had chosen the alternative policy, we would have obtained a similar result by composing with the identity on the right-hand-side.

2.2.1.2 All the equivalences

We present in the table below the equivalences of proofs for MALL. These were partially presented, without the additive, in [40]. For each of the equivalence presented, we assume that there are proofs that lead to the leaves of the trees, and make sometimes the proof explicit by writing:

$$\dfrac{\pi}{\vdash \Gamma}$$

We start with the pure multiplicative fragment.

$\perp$ vs $\perp$:

$$\dfrac{\dfrac{\vdash \Gamma}{\vdash \Gamma, \perp_a}}{\vdash \Gamma, \perp_a, \perp_b} \sim \dfrac{\dfrac{\vdash \Gamma}{\vdash \Gamma, \perp_b}}{\vdash \Gamma, \perp_a, \perp_b}$$

$\perp$ vs $\parr$:

$$\dfrac{\dfrac{\vdash \Gamma, A, B}{\vdash \Gamma, A \parr B}}{\vdash \Gamma, A \parr B, \perp} \sim \dfrac{\dfrac{\vdash \Gamma, A, B}{\vdash \Gamma, A, B, \perp}}{\vdash \Gamma, A \parr B, \perp}$$

$\perp$ vs $\otimes$:

$$\dfrac{\dfrac{\vdash \Gamma, A}{\vdash \Gamma, A, \perp} \qquad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B, \perp} \sim \dfrac{\dfrac{\vdash \Gamma, A \qquad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B}}{\vdash \Gamma, \Delta, A \otimes B, \perp} \sim \dfrac{\vdash \Gamma, A \qquad \dfrac{\vdash \Delta, B}{\vdash \Delta, B, \perp}}{\vdash \Gamma, \Delta, A \otimes B, \perp}$$

$\parr$ vs $\parr$:

$$\dfrac{\dfrac{\vdash \Gamma, A, B, C, D}{\vdash \Gamma, A \parr B, C, D}}{\vdash \Gamma, A \parr B, C \parr D} \sim \dfrac{\dfrac{\vdash \Gamma, A, B, C, D}{\vdash \Gamma, A, B, C \parr D}}{\vdash \Gamma, A \parr B, C \parr D}$$

$\parr$ vs $\otimes$:

$$\dfrac{\dfrac{\vdash \Gamma, A, B, C \qquad \vdash \Delta, D}{\vdash \Gamma, \Delta, A, B, C \otimes D}}{\vdash \Gamma, \Delta A \parr B, C \otimes D} \sim \dfrac{\dfrac{\vdash \Gamma, A, B, C}{\vdash \Gamma, A \parr B, C} \qquad \vdash \Delta, D}{\vdash \Gamma, \Delta, A \parr B, C \otimes D}$$

⊗ vs ⊗ :

$$\dfrac{\vdash \Gamma, A \qquad \dfrac{\vdash \Delta, B, C \qquad \vdash \Xi, D}{\vdash \Delta, \Xi, B, C \otimes D}}{\vdash \Gamma, \Delta, \Xi, A \otimes B, C \otimes D} \sim \dfrac{\dfrac{\vdash \Gamma, A \qquad \vdash \Delta, B, C}{\vdash \Gamma, \Delta, A \otimes B, C} \qquad \vdash \Xi, D}{\vdash \Gamma, \Delta, \Xi, A \otimes B, C \otimes D}$$

Then the pure additive one.

⊤ vs ⊤ :

$$\dfrac{}{\vdash \top_a, \top_b, \Gamma}\,\top_a \sim \dfrac{}{\vdash \top_a, \top_b, \Gamma}\,\top_b$$

⊤ vs ⊕ :

$$\dfrac{\dfrac{}{\vdash \Gamma, A, \top}\,\top}{\vdash \Gamma, A \oplus B, \top} \sim \dfrac{}{\vdash \Gamma, A \oplus B, \top}\,\top \sim \dfrac{\dfrac{}{\vdash \Gamma, B, \top}\,\top}{\vdash \Gamma, A \oplus B, \top}$$

⊤ vs & :

$$\dfrac{\dfrac{}{\vdash \Gamma, A, \top}\,\top \qquad \dfrac{\pi}{\Gamma, \top, B}}{\vdash \Gamma, A \,\&\, B, \top} \sim \dfrac{}{\vdash \Gamma, A \,\&\, B, \top}\,\top \sim \dfrac{\dfrac{\pi'}{\Gamma, A, \top} \qquad \dfrac{}{\Gamma, B, \top}\,\top}{\Gamma, A \,\&\, B, \top}$$

⊕ vs ⊕ :

$$\dfrac{\dfrac{\vdash \Gamma, A, C}{\vdash \Gamma, A \oplus B, C}}{\vdash \Gamma, A \oplus B, C \oplus D} \sim \dfrac{\dfrac{\vdash \Gamma, A, C}{\vdash \Gamma, A, C \oplus D}}{\vdash \Gamma, A \oplus B, C \oplus D}$$

⊕ vs & :

$$\dfrac{\dfrac{\vdash \Gamma, A, C \qquad \vdash \Gamma, A, D}{\vdash, \Gamma, A, C \,\&\, D}}{\vdash \Gamma, A \oplus B, C \,\&\, D} \sim \dfrac{\dfrac{\vdash \Gamma, A, C}{\vdash \Gamma, A \oplus B, C} \qquad \dfrac{\Gamma, A, D}{\vdash \Gamma, A \oplus B, D}}{\vdash \Gamma, A \oplus B, C \,\&\, D}$$

& vs & :

$$\dfrac{\dfrac{\vdash \Gamma, A, C \qquad \vdash \Gamma, B, C}{\vdash \Gamma, A \,\&\, B, C} \qquad \dfrac{\vdash \Gamma, A, D \qquad \vdash \Gamma, B, D}{\vdash \Gamma, A \,\&\, B, D}}{\vdash \Gamma, A \,\&\, B, C \,\&\, D}$$

$$\sim$$

$$\dfrac{\dfrac{\vdash \Gamma, A, C \qquad \vdash \Gamma, A, D}{\vdash \Gamma, A, C \,\&\, D} \qquad \dfrac{\vdash \Gamma, B, C \qquad \vdash \Gamma, B, D}{\vdash \Gamma, B, C \,\&\, D}}{\vdash \Gamma, A \,\&\, B, C \,\&\, D}$$

And finally when the two fragments interact :

⊥ vs ⊤:

$$\dfrac{\dfrac{}{\vdash \Gamma, \top}\,\top}{\vdash \Gamma, \bot, \top} \sim \dfrac{}{\vdash \Gamma, \bot, \top}\,\top$$

⊥ vs & :

$$\dfrac{\dfrac{\vdash \Gamma, A \qquad \vdash \Gamma, B}{\vdash \Gamma, A \,\&\, B}}{\vdash \Gamma, A \,\&\, B, \bot} \sim \dfrac{\dfrac{\vdash \Gamma, A}{\vdash \Gamma, A, \bot} \qquad \dfrac{\vdash \Gamma, B}{\vdash \Gamma, B, \bot}}{\vdash \Gamma, A \,\&\, B, \bot}$$

⊥ vs ⊕ :

$$\dfrac{\dfrac{\vdash \Gamma, A}{\vdash, \Gamma, A, \bot}}{\vdash \Gamma, A \oplus B, \bot} \sim \dfrac{\dfrac{\vdash \Gamma, A}{\vdash, \Gamma, A \oplus B}}{\vdash \Gamma, A \oplus B, \bot}$$

⊗ vs ⊤ :

$$\dfrac{\dfrac{}{\vdash \Gamma, A, \top}\,\top \qquad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B, \top} \sim \dfrac{}{\vdash \Gamma, \Delta, A \otimes B, \top}\,\top \sim \dfrac{\vdash \Gamma, A \qquad \dfrac{}{\vdash \Delta, B, \top}\,\top}{\vdash \Gamma, \Delta, A \otimes B, \top}$$

⊗ vs & :

$$\dfrac{\dfrac{\vdash \Gamma, A, C \qquad \vdash \Gamma, B, C}{\vdash \Gamma, A \,\&\, B, C} \qquad \vdash \Delta, D}{\vdash \Gamma, \Delta, A \,\&\, B, C \otimes D} \sim \dfrac{\dfrac{\vdash \Gamma, A, C \qquad \vdash, \Delta D}{\vdash \Gamma, \Delta, A, C \otimes D} \qquad \dfrac{\vdash \Gamma, B, C \qquad \vdash, \Delta, D}{\vdash \Gamma, \Delta, B, C \otimes D}}{\vdash \Gamma, \Delta, A \,\&\, B, C \otimes D}$$

⊗ vs ⊕ :

$$\dfrac{\dfrac{\vdash \Gamma, A, C \qquad \vdash \Delta, D}{\vdash \Gamma, \Delta, A, C \otimes D}}{\vdash \Gamma, \Delta, A \oplus B, C \otimes D} \sim \dfrac{\dfrac{\vdash \Gamma, A, C}{\vdash \Gamma, A \oplus B, C} \qquad \vdash \Delta, D}{\vdash \Gamma, \Delta, A \oplus B, C \otimes D}$$

⅋ vs ⊤ :

$$\dfrac{\dfrac{}{\vdash \Gamma . A, B, \top}\top}{\vdash \Gamma, A \parr B, \top} \sim \dfrac{}{\vdash \Gamma, A \parr B, \top}\top$$

⅋ vs & :

$$\dfrac{\dfrac{\vdash \Gamma, A, B, C \qquad \vdash \Gamma, A, B, D}{\vdash \Gamma, A, B, C \,\&\, D}}{\vdash \Gamma, A \parr B, C \,\&\, D} \sim \dfrac{\dfrac{\vdash \Gamma, A, B, C}{\vdash \Gamma, A \parr B, C} \qquad \dfrac{\vdash \Gamma, A, B, D}{\vdash \Gamma, A \parr B, D}}{\vdash \Gamma, A \parr B, C \,\&\, D}$$

⅋ vs ⊕ :

$$\dfrac{\dfrac{\vdash \Gamma, A, B, C}{\vdash \Gamma, A \parr B, C}}{\vdash \Gamma, A \parr B, C \oplus D} \sim \dfrac{\dfrac{\vdash \Gamma, AB, C}{\vdash \Gamma, A, B, C \oplus D}}{\vdash \Gamma, A \parr B, C \oplus D}$$

2.2.2 Focusing in linear logic

The rules of the sequent calculus of MALL are of two kinds: the synchronous ones, and the asynchronous ones. Asynchronous rules are those that can always be permuted with other rules, that is, pushed down the proof-tree. For instance, the introduction rule of $\bot$ is asynchronous. They will later correspond to opponent moves in games semantics, and the synchronous ones correspond to player, or proponent moves. Seeing a proof as a strategy, if a proof is able to play a (player)-move at some point in the game, then it can still produce it after some more opponent moves. Therefore, the opponent-moves that happened after the player-move can be inverted with this move and played before-hand. On the other hand, in a proof, a player move cannot be inverted with the opponent moves that happened before, since it might *depend* on some of the informations produced by these opponent-moves.

By extension, the connectives of linear logic are given a **polarity**, in relation with their introduction rules. The connectives $\otimes, \oplus$ are positive, and their negations $\parr, \&$ are negative. Similarly, $\bot, \top$ are negative, and $I, 0$ are positive. On the other hand, assigning a polarity to literals must be an arbitrary choice, since $X, X^\perp$ are introduced alongside by the same rule, but it is common to consider X as positive, and $X^\perp$ as negative [11], and corresponds to the idea that the axiom link is directed from $X^\perp$ to X. According to the above remark, it follows that the formulas of linear logic can be split into two sets N, P of negative and positive formulas respectively.

$$\begin{aligned} P &::= X \mid F \otimes F' \mid F \oplus F' \mid 0 \mid 1 \mid N^\perp \\ N &::= F \parr F' \mid F \,\&\, F' \mid \top \mid \bot \mid P^\perp \end{aligned}$$

where F, F' are formulas of MALL.

If the polarity assigned to a rule is something rooted in the sequent calculus, its extension to formulas is a bit dubious. Therefore, the polarity of formulas should not be considered as a ground feature of linear logic, but more as a tool, that will later be used in order to devise properly the focussed sequent calculus.

These distinctions between asynchronous and synchronous rules lead to a second sequent calculus for linear logic, called **focussed** (or focalised in the literature, but here we will restrain from using the term focalised since it will refer to another property) Before presenting it formally, we expose the general idea.

According to the definition of asynchronous rules, they can all be pushed backwards towards the root of the proof-tree, until they are blocked by a positive connective that happens before in the syntactic formula tree (for instance, one cannot push the & before the $\oplus$ when trying to prove $A \oplus (B \,\&\, C)$). Once this process is finished, we end up with a proof-stree such that starting from the root of the tree and going upward, we will encounter only negative rules until all formulas (except maybe the literals) in the sequent are positive. Now, a proof is focussed, if it chooses one of the positive decomposable formulas, and all the next rules are positive rules decomposing this formula until all the resulting sub-formulas are either negative or literals. Then, one can repeat this routine: a set of negative rules until all formulas in the sequent are positive, following by a choice of a focus, and the decomposition of the focussed formula. The stunning result established by Andreoli [11] is that every cut-free proof of linear logic is equivalent to a focussed one. This gives us a precious insight into the structure of cut-free proofs of linear logic.

Let $\mathcal{P}$ stands for a list of positive formulas, $\mathcal{N}$ be a list of negative formulas, $\mathcal{X}$ a list of negative atomic formulas, P, Q are positive formulas, N, M negative ones. The sequents are of two shapes: either $\vdash \mathcal{P}, \mathcal{N}, \mathcal{X};$ or $\vdash \mathcal{P}, \mathcal{X}; P$. The **focussed sequent calculus** is defined as follows. Note that O is a formula of either positive or negative polarity.

$$\frac{}{\vdash A^{\perp}; A}\ \text{Ax} \qquad \frac{\vdash \mathcal{P}, \mathcal{X}; P}{\vdash \mathcal{P}, \mathcal{X}, P;}\ \text{foc}$$

$$\frac{\vdash \mathcal{P}, \mathcal{N}, A, B;}{\vdash \mathcal{P}, \mathcal{N}, A \parr B;}\ \parr$$

$$\frac{\vdash \mathcal{P}, \mathcal{X}; P \qquad \vdash \mathcal{P}', \mathcal{X}'; Q}{\vdash \mathcal{P}, \mathcal{P}', \mathcal{X}, \mathcal{X}'; P \otimes Q}\ \otimes \qquad \frac{\vdash \mathcal{P}, \mathcal{X}; P \qquad \vdash \mathcal{P}', \mathcal{X}', M;}{\vdash \mathcal{P}, \mathcal{P}', \mathcal{X}, \mathcal{X}'; P \otimes M}\ \otimes$$

$$\frac{\vdash \mathcal{P}, \mathcal{X}, M; \qquad \vdash \mathcal{P}', \mathcal{X}'; P'}{\vdash \mathcal{P}, \mathcal{P}', \mathcal{X}, \mathcal{X}'; M \otimes P'}\ \otimes \qquad \frac{\vdash \mathcal{P}, \mathcal{X}, M; \qquad \vdash \mathcal{P}', \mathcal{X}', N;}{\vdash \mathcal{P}, \mathcal{P}', \mathcal{X}, \mathcal{X}'; M \otimes N}\ \otimes$$

$$\frac{\vdash \mathcal{P}, \mathcal{N}, \mathcal{X}, A; \qquad \vdash \mathcal{P}, \mathcal{N}, \mathcal{X}, B;}{\vdash \mathcal{P}, \mathcal{N}, \mathcal{X}, A \,\&\, B;}\ \&$$

$$\frac{\vdash \mathcal{P}, \mathcal{X}; P}{\vdash \mathcal{P}, \mathcal{X}; P \oplus O}\ \oplus_1 \qquad \frac{\vdash \mathcal{P}, \mathcal{X}; P}{\vdash \mathcal{P}, \mathcal{X}; O \oplus P}\ \oplus_2$$

$$\frac{\vdash \mathcal{P}, M, \mathcal{X};}{\vdash \mathcal{P}, \mathcal{X}; M \oplus O}\ \oplus_1 \qquad \frac{\vdash \mathcal{P}, \mathcal{X}, M;}{\vdash \mathcal{P}, \mathcal{X}; O \oplus M}\ \oplus_2$$

$$\frac{}{\vdash ; 1}\ 1 \qquad \frac{\mathcal{P}, \mathcal{N}, \mathcal{X};}{\mathcal{P}, \mathcal{N}, \mathcal{X}, \bot;}\ \bot \qquad \frac{}{\mathcal{P}, \mathcal{N}, \mathcal{X}, \top;}\ \top$$

A slightly weaker system is presented, called **weakly focused**. Just as in [59], where it was originally presented, we write Π for a sequent that is either empty or consists of a unique positive formula P. The sequents of $\mathsf{MALL_{wfoc}}$ are of the form $\vdash \Gamma; \Pi$, where Γ is a multi-set of formulas. This system is more permissive than $\mathsf{MALL_{foc}}$ as it allows rules to be applied to negative formulas on the left-hand-side of the sequent even when the right-hand-side of the sequent is non-empty. Furthermore, the negative formulas are now pushed on the left-hand-side of the sequent thanks to a "unfoc"-rule, instead of a built-in machinery.

$$\frac{}{\vdash A^{\bot}; A}\ \mathrm{Ax} \qquad \frac{\vdash \Gamma; P}{\vdash \Gamma, P;}\ \mathrm{foc} \qquad \frac{\vdash \Gamma, M;}{\vdash \Gamma; M}\ \mathrm{unfoc}$$

$$\frac{\vdash \Gamma, A, B; \Pi}{\vdash \Gamma, A \parr B; \Pi}\ \parr$$

$$\frac{\vdash \Gamma; A \qquad \vdash \Delta; B}{\vdash \Gamma, \Delta; A \otimes B}\ \otimes$$

$$\frac{\vdash \Gamma, A; \Pi \qquad \vdash \Gamma, B; \Pi}{\vdash \Gamma, A \,\&\, B; \Pi}\ \&$$

$$\frac{\vdash \Gamma; A}{\vdash \Gamma; A \oplus B}\ \oplus_1 \qquad \frac{\vdash \Gamma; B}{\vdash \Gamma; A \oplus B}\ \oplus_2$$

$$\frac{}{\vdash ; 1}\ 1 \qquad \frac{\vdash \Gamma; \Pi}{\vdash \Gamma, \bot; \Pi}\ \bot \qquad \frac{}{\vdash \Gamma, \top; \Pi}\ \top$$

Theorem 2.1. *[11] [59] Every proof of* MALL *is equivalent to a proof of* $\mathsf{MALL_{wfoc}}$ *(resp* $\mathsf{MALL_{Foc}}$*), and* $\mathsf{MALL_{wfoc}}$ *(resp* $\mathsf{MALL_{Foc}}$*) can be seen as a subsystem of* MALL.

2.2.2.1 Global connectives

In this section, we strive to prove that proofs of linear logic can be represented, up to equivalence by two global connectives. We present in the paragraph an intermediate step, with four ones, that will be enough for our purposes. They consist of two positive ones, that encapsulate sequences of $\otimes$ and $\oplus$, and two negative ones, that encapsulate sequences of $\parr$ and &. However, for such a system to work, one must consider formulas up to distributivity isomorphisms.

In MALL, there are four distributivity laws, coming from proofs of linear logic.

$$A \otimes (B \oplus C) \simeq (A \otimes B) \oplus (A \otimes C)$$
$$(A \oplus B) \otimes C \simeq (A \otimes C) \oplus (B \otimes C)$$
$$A \parr (B \,\&\, C) \simeq (A \parr C) \,\&\, (A \parr C)$$
$$(A \,\&\, B) \parr C \simeq (A \parr C) \,\&\, (B \parr C)$$

We write $F_1 \simeq F_2$ to indicate that there are two proofs $\pi_1 : F_1 \to F_2$, and $\pi_2 : F_2 \to F_1$ such that $\pi_1;_{cut}\pi_2 \sim \mathsf{id}_{F_1}$ and $\pi_2;_{cut}\pi_1 \sim \mathsf{id}_{F_1}$. We present these proofs for the first equation, while the others can be designed in a similar way.

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdash A^\perp, A \qquad \vdash B^\perp, B}{\vdash A^\perp, B^\perp, A \otimes B}}{\vdash A^\perp, B^\perp, (A \otimes B) \oplus (A \otimes C)} \qquad \dfrac{\dfrac{\vdash A^\perp, A \qquad \vdash C^\perp, C}{\vdash A^\perp, C^\perp, A \otimes C}}{\vdash A^\perp, C^\perp, (A \otimes B) \oplus (A \otimes C)}}{\vdash A^\perp, B^\perp \mathbin{\&} C^\perp, (A \otimes B) \oplus (A \otimes C)}}{\vdash A^\perp \parr (B^\perp \mathbin{\&} C^\perp), (A \otimes B) \oplus (A \otimes C)}$$

$$\dfrac{\dfrac{\dfrac{\vdash A^\perp, A \qquad \dfrac{\vdash B^\perp, B}{\vdash B^\perp, B \oplus C}}{A^\perp, B^\perp, A \otimes (B \oplus C)}}{A^\perp \parr B^\perp, A \oplus B \oplus C} \qquad \dfrac{\dfrac{\vdash A^\perp, A \qquad \dfrac{\vdash C^\perp, C}{\vdash C^\perp, B \oplus C}}{\vdash A^\perp, C^\perp, A \otimes (B \oplus C)}}{\vdash A^\perp \parr C^\perp, A \otimes (B \oplus C)}}{\vdash (A^\perp \parr B^\perp) \mathbin{\&} (A^\perp \parr C^\perp), A \otimes (B \oplus C)}$$

As these proofs are isomorphisms, they only change the set of morphisms up to (set)-isomorphism. That is, given a formula A, such that, for instance, π_1 can be applied to A, $\pi_1 : A \to A'$, then $\mathcal{C}(A, B) \simeq \mathcal{C}(A', B)$. Therefore we consider the following rewriting system:

$$A \otimes (B \oplus C) \rightsquigarrow (A \otimes B) \oplus (A \otimes C)$$
$$(A \oplus B) \otimes C \rightsquigarrow (A \otimes C) \oplus (B \otimes C)$$
$$A \parr (B \mathbin{\&} C) \rightsquigarrow (A \parr C) \mathbin{\&} (A \parr C)$$
$$(A \mathbin{\&} B) \parr C \rightsquigarrow (A \parr C) \mathbin{\&} (B \parr C)$$
$$A \rightsquigarrow B \;\Rightarrow\; F[A] \rightsquigarrow F[B]$$

This system is confluent and strongly normalizing. Therefore, for any given formula A, there exists a unique A' such that A' is the normal form of A. That is, A' is obtained from A by a sequence of steps from the rewriting system, and one cannot rewrite A' any further. Then given A, B any formula of linear logic, and A', B' their normal form, we have the following isomorphism:

$$\mathcal{C}(A, B) \simeq \mathcal{C}(A', B')$$

as each of the rewriting step corresponds to the application of an isomorphism. In other terms, when dealing with proof invariants, one can restrict to considering only formulas in normal form. This generalises straightforwardly to sequents, where a sequent is considered in normal form if each formula in it is.

A second important set of isomorphisms comes from the associativity of each of the connective of linear logic $\otimes, \parr, \&, \oplus$. For instance:

$$\mathcal{C}(I, A \otimes (B \otimes C)) \simeq \mathcal{C}(I, (A \otimes B) \otimes C).$$

This isomorphism allows us to forget the parentheses, and simply write it as $A \otimes B \otimes C$, without specifying the order in which the $\otimes$ connectives must be considered. This generalises straightforwardly in the case where we consider n-tensored formulas, and we write $\bigotimes_{i=1...n} A_i$ in that case. In terms of proofs, these isomorphisms tell us that the order in which the $\otimes$-rules are applied does not matter. This reasoning applies to the other binary connectives of MALL similarly.

By considering formulas up to these two sets of isomorphisms, we can restrict to formulas of two different forms. A negative formula in normal form, is, once considered up to associativity isomorphisms, shaped as follows: $\&_i(\bigparr_j A_{i,j})$, where each $A_{i,j}$ is either a literal or a positive formula in normal form. On the other hand, a positive formula in normal form is of the shape: $\bigoplus_i(\bigotimes_j A_{i,j})$, where each $A_{i,j}$ is either a literal or a negative formula in normal form. We will say that these formulas are in **distributive-associative normal form**.

Now, let us consider a focused proof of linear logic, where the sequents are in normal form. Then the root of the proof is of the shape $\mathcal{P}, \mathcal{N}, \mathcal{X}$. We see it as a unique formula $(\bigparr \mathcal{P}) \parr (\bigparr \mathcal{N}) \parr (\bigparr \mathcal{X})$, and put it in distributive-associative normal form. That is, it is a formula of the shape $\&_i(\bigparr_j A_{i,j})$. If this ends up being the $\&$ of 0 formula, that is of the shape $\& \emptyset = \top$, then the proof proceeds with a $\top$-rule. Otherwise, a proof of this sequent starts with a global $\&$-connective, that encapsulates a series of $\&$-rules.

$$\dfrac{\vdash \bigparr_j A_{1,j}; \qquad ... \qquad \vdash \bigparr_j A_{i,j}; \qquad ... \qquad \vdash \bigparr_n A_{n,j};}{\vdash \&_i(\bigparr_j A_{i,j});} \&$$

Then, let us focus on one of the branch. The second step consists in decomposing the $\parr$:

$$\dfrac{\vdash A_{i,1}, .., A_{i,j}, .., A_{m,j} :}{\vdash \bigparr_j A_{i,j};} \bigparr$$

And finally, if there is some $\bot$ present among the $A_{i,j}$, then the proof removes them using a global $\bot$-rule, whose shape is as follows:

$$\dfrac{\vdash A_1, .., A_n}{\vdash A_1, ..., A_n, \bot_1, ..., \bot_m} \bot_m$$

This is the end of the asynchronous phase. These three global connectives will be encapsulate into one negative move in the sequel, Chapter 5. At this point the synchronous phase begins. It starts with a choice of a focus, that is, a positive formula.

$$\dfrac{\vdash \Gamma, P;}{\vdash \Gamma; P} \text{Foc}$$

Then, as we consider formulas in distributive-associative normal form, $P = \bigoplus_i(\bigotimes_j M_{i,j} \bigotimes_k X_{i,k}, \bigotimes_l 1_{i,l})$. Then the focused proof at this stage going to consists in three distinct steps. The first one is going to consist in a series of $\oplus$ rule, that we can sum up into one major $\bigoplus$ rule:

$$\frac{\Gamma \vdash P_i}{\Gamma \vdash \bigoplus_i P_i} \oplus$$

Following that, the P_i is going to consists in tensored formulas $P_i = (\bigotimes_j M_{i,j} \bigotimes_k X_{i,k} \bigotimes_l 1_{i,l})$. The following sequence of $\otimes$-rules can be summed up into a single global $\bigotimes$-rule:

$$\frac{\Gamma_1 \vdash B_1 \qquad \ldots \qquad \Gamma_i \vdash B_i \qquad \ldots \qquad \Gamma_n \vdash B_n}{\Gamma_1, .., \Gamma_i, ..., \Gamma_n \vdash \bigotimes_{i=1..n} B_i} \bigotimes$$

Finally, the third part of the synchronous phase is going to consist in dealing with all the branches just created. For instance, if the proof reaches a negative formula on the right-hand-side, then it pushes it on the left-hand-side, thanks to a unfoc-rule.

$$\frac{\Gamma, M;}{\Gamma; M} \text{unfoc}$$

If, on the other hand, it reaches a sequent $\vdash; 1$, then it applies the right 1 rule. Finally, if the sequent reached is of the form $\Gamma; X$, then we must have $\Gamma = X$, and the proof-leaf consists of an axiom-rule.

Therefore, we can consider the following proof-system, called $\mathsf{MALL}_{\mathsf{foc-glob}}$. We decompose the formulas into four sets, depending on what is their principal connective.

$$\begin{aligned}
N &= \bigwith_i O_i \mid \top \mid l \\
O &= \bigparr_i P_i \bigparr_j \bot_j \\
P &= \bigoplus_i R_i \mid 0 \mid l \\
R &= \bigotimes_i N_i \bigotimes_j 1_j \\
l &::= X \mid X^\perp
\end{aligned}$$

where $X \in \mathsf{TVar}$. Note that $\top$ corresponds to $\bigwith \emptyset$, and 0 to $\bigoplus \emptyset$. In the following, $\mathcal{P}$ denotes a list of P-formulas, $\mathcal{N}$ a list of N-formulas, and N, O, R, P formulas as above.

$$\frac{}{\vdash X^\perp; X} \text{Ax} \qquad \frac{\vdash \mathcal{P}; P}{\vdash \mathcal{P}, P;} \text{foc} \qquad \frac{\vdash \mathcal{P}, N;}{\vdash \mathcal{P}; N} \text{unfoc}$$

$$\frac{\vdash \mathcal{P}, P_1, ..., P_m, \bot_1, ..\bot_n;}{\vdash \mathcal{P}, \bigparr_i P_i \bigparr_j \bot_j;} \bigparr$$

$$\frac{\vdash \mathcal{P}_1; N_1 \qquad \ldots \qquad \vdash \mathcal{P}_i; N_i \qquad \ldots \qquad \vdash \mathcal{P}_n; N_n}{\vdash \mathcal{P}_1, ..., \mathcal{P}_n; \bigotimes_i N_i} \bigotimes$$

$$\frac{\vdash \mathcal{P}; R_k}{\vdash \mathcal{P}; \bigoplus_{i=1..n} R_i} \bigoplus_{k,n}$$

$$\frac{\vdash \mathcal{P}, O_1; \quad \ldots \quad \vdash \mathcal{P}, O_i; \quad \ldots \quad \vdash \mathcal{P}, O_n;}{\vdash \mathcal{P}, \&_{i=1..n} O_i;} \&$$

$$\frac{}{\vdash; 1} 1 \qquad \frac{\vdash \mathcal{P};}{\vdash \mathcal{P}, \bot_1, \ldots, \bot_m;} \bot_m \qquad \frac{}{\mathcal{P}, \top;} \top$$

As explained above, the following theorem holds.

Theorem 2.2. *Every proof* $\pi :\vdash \Gamma$ *of* MALL *is equivalent to a proof* $\pi' : A$ *of* $\mathsf{MALL}_{\mathsf{foc-glob}}$*, where* A *is the distributive-associative normal form of* $\bigparr \Gamma$.

This system can be once again reformulated into a system with two global connectives: the positive one, that is a mix of $\bigoplus$ and $\bigotimes$, and its negation, a global negative one. This is the idea underlying the creation of ludics [35], and that originates fully complete alternating game models, such as the one presented in the next chapter 5.

2.2.3 Proof nets

In the next chapters, one will often rely on a geometric characterisation of invariants of linear proofs, called proof nets. The basic structures we rely on are proof-structures. We first consider the MLL case, then the MALL one.

2.2.3.1 Proof structures for MLL

Proof structures form a common tool introduced in [33] to reason about invariants of linear logic proofs. Proofs structures are more general than proof invariants. That is, some proof structures do not correspond to proof invariants, but to all proof invariants corresponds one, or several, proof structures. Proof structures that indeed correspond to proof invariants are refereed to as proof-nets. Therefore, one seeks to find the right characterization, that discriminates proof-structures that are valid, that is, correspond to proofs. Proof structures can be composed, and form a category, of which the proof-nets form a sub-category. However, this topic exceeds the scope of this introduction.

Often, in the literature, one will encounter the definition restricted to MLL without units. The reason behind this choice is that, for this fragment, proof structures that are valid are in one-to-one correspondence to proof invariants of linear logic restricted to this fragment. However, when extended to units, we lose this property. That is, a single proof invariant can actually be encoded as various proof structures: proof structures are then too precise. One can bypass this issue by defining a quotient, using a method called Trimble's empire rewriting [47, 48]. However, for brevity, we will not expose this method, and refer to the above references for its presentation.

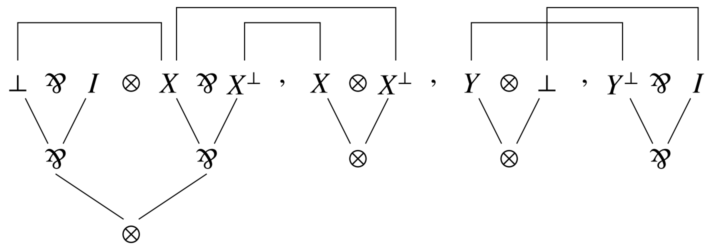


Figure 2.2: A MLL-proof structure

Definition 2.3. *An **MLL sequent** is a multiset of formulas built out of the following grammar :*

$$F, F' ::= X \mid X^{\perp} \mid I \mid \perp \mid F \otimes F' \mid F \parr F',$$

*where $X \in$ TVar. An **MLL^{-} sequent** is a multiset of formulas built out of the following grammar:*

$$F, F' := X \mid X^{\perp} \mid F \otimes F' \mid F \parr F'$$

*where $X \in$ TVar. A sequent is said to be **balanced** if each atomic variable X appears the same number of times as its negation $X^{\perp}$.*

In the sequel, we will refer to MLL^{-} and MLL to speak about the canonical fragment of MALL whose cut-free proofs are those with conclusions lying inside MLL^{-} and MLL respectively.

In the sequel, we identify formulas with their parse trees, and hence see them as tree-graphs. Therefore, a sequent is seen as a forest. The propositional variables $X, Y, ...$ as well as the unit 1 are positive, whereas their negation $X^{\perp}, Y^{\perp}, ...$ and $\perp$ are negative. Given a balanced MLL-sequent, we define a **linking** λ to be a function from its set of negative leaves to its set of positive ones, that is type preserving : $\lambda(X^{\perp}) = X$, and such that it establishes a bijection between its set of occurrences of positive propositional variables and negative ones. This enforces that the sequent be balanced.

Definition 2.4. *A **proof structure** is a graph, made out from a sequent $\vdash \Gamma$, seen as a forest, together with a linking function λ on it, by adding edges between x and $\lambda(x)$. When $x, \lambda(x)$ are propositional variables, such an edge is called an **axiom link**. On the other hand, when $x = \perp$, we call it a **$\perp$-link**.*

Such a proof-structure is presented in the figure 2.2. There is a famous criterion that characterizes precisely the proof structures that arise from the denotation of an actual proof of MLL. It has been first presented by Danos and Regnier in [23] for MLL^{-}. To implement it, we first need to define the notion of switching.

Definition 2.5. *Given a proof structure P, a switching $\mathcal{S}$ is a choice, for each $\parr$-occurrence*

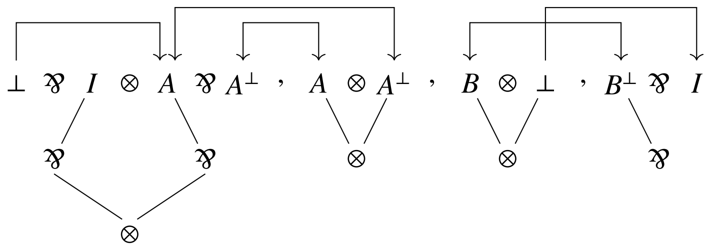


Figure 2.3: An MLL correction graph

in the parse tree, of the left or the right premise. That is, it is a function from the set of $\parr$*-occurrences to the set* $\{\mathsf{l}, \mathsf{r}\}$.

The correction graph consists in, given a proof structure P and a switching $\mathcal{S}$, removing the edge between the premise not chosen by the switching and its conclusion. We say that a proof structure is acyclic (respectively connected) if the correction graphs are connected (respectively acyclic) for all the switchings $\mathcal{S}$ on it.

For instance, the correction graph in figure 2.3 is disconnected and acyclic.

Theorem 2.6. *[23, 48]*

- *To every equivalence class of proofs of* MLL^{-}*, or* MLL^{-}+MIX*, one can assign a unique* MLL^{-}*proof structure (that is, a unique linking). Furthermore, this assignment is faithful and functorial.*
- *An* MLL^{-}*proof structure is a denotation of an* MLL^{-}+ MIX *proof if and only if it is acyclic.*
- *An* MLL^{-}*proof structure is a denotation of an* MLL^{-}*proof it is acyclic and connected, that is, a tree.*
- *To every* MLL *proof structure satisfying the acyclicity criterion, one can canonically assign an* MLL + MIX *proof.*
- *To every* MLL *proof-structure satisfying the acyclicity and the connectedness criterion, one can assign an* MLL *proof.*
- *To every* MLL *proof, one can assign a unique equivalence class of* MLL *proof structures satisfying both criteria. This assignment is functorial, and an isomorphism of categories.*

We will sometimes refer to these conditions (connectedness and acyclicity) as the Danos-Reigner criterion. We say that an MLL^{-} proof structure is a **proof-net** if it satisfies the Danos-Reigner criterion. That is, a proof-net is a proof structure that can be assigned a proof.

2.2.3.2 Proof structures for MALL

In this section, we will restrict our attention to MALL^{-}, the fragment of MALL without units, neither additive nor multiplicative. That is, we focus purely on the propositional part. The grammar of the formulas MALL^- is defined as follows :

$$F ::= X \mid X^\perp \mid F \otimes F' \mid F \parr F' \mid F \oplus F' \mid F \,\&\, F'.$$

where $X \in \mathsf{TVar}$. Historically, two notions of proof structures, and proof-nets, have been developed for MALL^-. The first one, presented by Girard in [36] is based on a notion of graph enriched with booleans, that tracks in what "branch" of a &-link we are. Although we will make use of them in the final chapter 7, they are hard to present, understand, and are mostly dealt with as a technical tool in the scope of this thesis. We will simply say here that their correctness criterion is related to the one of MLL; the notion of switching is much more elaborate, but the final correction graphs must also be acyclic and connected.

The second one, presented by Hughes in [46], satisfies a much simpler presentation, and its exposition might help the reader getting an idea of how invariants of proofs behave in MALL. The correctedness criterion, on the other hand, is a bit obscure, and we will restrict our attention to proof structures. We refer to the literature [46] for more details.

Once again, we see each sequent as a set of parse trees, hence a parse forest. A &-**resolution** on Γ is the result of erasing one argument sub-graph of each &-occurrence. An **additive resolution** is the result of deleting one argument sub-graph of each $\oplus$, &-occurrence. We say that an additive resolution is on a &-resolution, if it can be obtained through deleting argument sub-trees of the $\oplus$-occurrences of this &-resolution.

An **axiom-link** is an edge between complementary literal. A **linking** on a sequent Γ is a set λ of disjoint axiom links such that $\bigcup \lambda$ partitions the set of leaves of an additive resolution of Γ, called $\Gamma \upharpoonright \lambda$. That is, each literal of the additive resolution is under an unique axiom-link.

A MALL-**proof structure** comprises a MALL-sequent, seen as a forest, together with a set of linkings $\Lambda = \{\lambda_1, ..., \lambda_n\}$, such that:

- (P1) For each &-resolution of Γ, there is a unique linking $\lambda \in \Lambda$ such that $\Gamma \upharpoonright \lambda$ is on this &-resolution.
- (P2) Each graph $\Gamma \upharpoonright \lambda$ satisfies the Danos-Reigner criterion: for any switching (choice for each $\parr$ of one of its subtree arguments), the correction graph is a tree.

We usually represent MALL proof structure with all the axiom-links coming from the set of linkings Λ at once, as presented in figure 2.4. As there is only once linking per &-resolution, one can recover the set of linkings by ranging through all &-resolutions.

For instance, there are only two possible &-resolutions on that proof structure, as there is only one &. The set of linkings defines two additive resolutions:

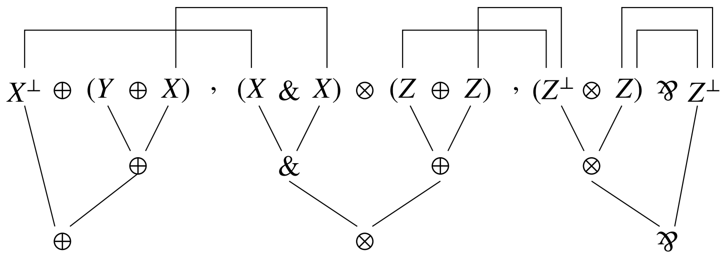


Figure 2.4: An MALL proof structure

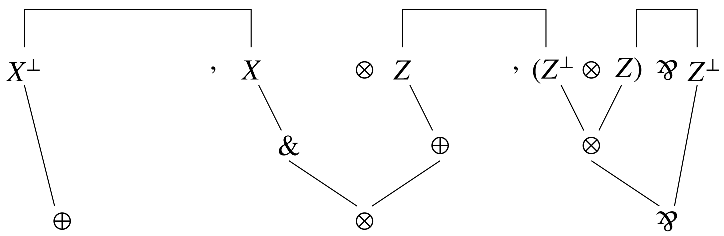


Figure 2.5: First additive resolution

is the first one, and the second is :

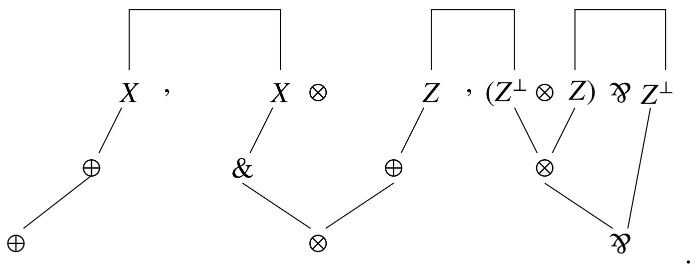


Figure 2.6: Second additive resolution

For a proof-structure of MALL^- to be a proof-net, one also needs the *toggling* condition, that is too technical to be exposed here. Just as the switching condition enforces that sequentialisation is possible for the multiplicative part, the toggling condition is concerned with the additive fragment of the proof. For the next chapters, the relevant aspect of the definition of proof structure is the condition (P1), that can actually be divided into two sub-conditions :

- For each & resolution, the proof chooses exactly one $\oplus$-resolution on this &-resolution.
- On this additive resolution, it defines a unique linking, that is, a unique set of axiom-links that partitions the leaves of this resolution.

Therefore, the proofs structure can be seen as functions f from the domain $\mathbb{D} = \times_{\&\in\Gamma}\{\mathsf{l}_\&, \mathsf{r}_\&\}$, where $\& \in \Gamma$ refers to the set of &-occurrences in Γ, to the set of linkings, such that given an element $k \in \mathbb{D}$, $\Gamma \upharpoonright f(k)$ is a $\oplus$-resolution compatible with k. This path has notably been explored in [2].

2.3 A brief introduction to tensorial logic

On the quest to fully complete game models of linear logic, two major difficulties have been highlighted. The first one concerned the ability of games to cope with polarities of the additives together with composition. Indeed, the initial presentation of a game semantics of linear logic, as devised by Blass [15], leads to the composition being non-associative. This is the so-called Blass problem, highlighted and explained by Abramsky in [5, 3]. The root of the issue lies in the order in which the proof of, for instance, $\vdash A_1 \oplus A_2, B_1 \oplus B_2$, decomposes the left $\oplus$ connector and the right one. Assuming that this order is relevant translates, in term of games, that the strategy has to start by either decomposing the first $\oplus$-occurence, or the second one. This is consistent with the strategy being sequential. However, this leads to a non-associative composition of strategies. To solve this, two solutions were proposed. First, we can consider concurrent, or asynchronous, game semantics where the two moves appear concurrently (the strategies decomposes the two at the same time). This way has been explored in [10, 3]. Or, we rather look at the proof from a focused point of view. Focussing leads to a clear control of the flow of the proof, and notably to a clear control of the flow of cut elimination. This way, the problematic cases that were non-associative are excluded. More on this issue can be found in [3].

If one chooses the focussing way out (and, therefore, keep on working with sequential games), a second problem arises from the difficulty to interpret the relationship between $\otimes$ and $\parr$. For instance, let us consider the following proof:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash A^\perp, A} \qquad \dfrac{}{\vdash B^\perp, B}}{\vdash A^\perp, B^\perp, A \otimes B} \qquad \dfrac{}{\vdash C^\perp, C}}{\vdash A^\perp, B^\perp \otimes C^\perp, A \otimes B, C}}{\vdash A^\perp \parr (B^\perp \otimes C^\perp), (A \otimes B) \parr C}}{\vdash A^\perp \parr (B^\perp \otimes C^\perp), (A \otimes B) \parr C}$$

As $A^\perp \parr (B^\perp \otimes C^\perp) = (A \otimes (B \parr C))^\perp$, it expresses the fact that $A \otimes (B \parr C) \vdash (A \otimes B) \parr C$. One might wonder where does this distributivity law comes from and how to translate it into game semantics. To succeed, the key is to consider that $\otimes$ or $\parr$ does not just put the two games in parallel (as in the concurrent case, or in the multiplicative case *without* additives [5]), but actually might add an initial move to the game that we have to take into account, as discovered by Melliès in [65]. That is, in those games, the multiplicative connectives $\otimes, \parr$ are played, there are moves corresponding to them. But this comes at a price. Now, $(A \otimes 1) \parr \perp$ has two additional moves, corresponding to the two binary tensors, but it is equivalent to A, that does not have such moves. This problem is deeply related to the fact that the negation is self-inverse. The game denoting $((A)^\perp)^\perp$ should be equal to the one denoting A. However, a faithful modelling through games that takes into account polarity would impose to add an initial move to the game for each negation. That is, the game for $((A)^\perp)^\perp$ should have two additional moves compare to the denotation of A. If this can be solved by a quotient, the conclusion of this paragraph is that there is no canonical good notion of sequential games for linear logic, as games rely to much on polarities, and because the negation is self inverse [65].

Tensorial logic has been introduced by Melliès [71], following his discovery of a fully complete model of linear logic [66], using sequential games as support and a quotient (as explained above, see [65]) to circumvent the problems arising from playing additional moves . Those games were somehow more primitive than linear logic itself, as emphasising more the role of polarity, and by being able to distinguish proofs that the sole equational theory of the sequent calculus of linear logic would not. As a result, tensorial logic was exposed as being, somehow, the logic of sequential games.

The formulas of tensorial logic are built out of three connectives $\otimes$, $\oplus$, $\neg$, the units $0, 1, \bot$, and an enumerable set of atomic variables $\mathsf{TVar} = X, Y, \ldots$.

$$F \ ::= \ 0 \ | \ 1 \ | \ \bot \ | \ X \ | F \otimes F \ | \ F \oplus F \ | \ \neg F \ ,$$

where $X \in \mathsf{TVar}$. The sequent calculus of multiplicative additive tensorial logic, or TENS is presented in the figure 2.7 below.

$$\text{Axiom}: \frac{}{A \vdash A} \qquad \text{Cut}: \frac{\Gamma \vdash A \qquad \Delta_1, A, \Delta_2 \vdash B}{\Delta_1, \Gamma, \Delta_2 \vdash B} \qquad \text{Left Exchange}: \frac{\Gamma_1, A, B, \Gamma_2 \vdash C}{\Gamma_1, B, A, \Gamma_2 \vdash C}$$

$$\text{Left Tensor}: \frac{\Gamma_1, A, B, \Gamma_2 \vdash C}{\Gamma_1, A \otimes B, \Gamma_2 \vdash C} \qquad \text{Right Tensor}: \frac{\Gamma \vdash A \qquad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}$$

$$\text{Right Unit}: \frac{}{\vdash 1} \qquad \text{Left Unit}: \frac{\Gamma_1, \Gamma_2 \vdash A}{\Gamma_1, 1, \Gamma_2 \vdash A} \qquad \text{Left } 0: \frac{}{\Gamma, 0 \vdash A}$$

$$\text{Left Negation}: \frac{\Gamma \vdash A}{\Gamma, \neg A \vdash \bot} \qquad \text{Right Negation}: \frac{\Gamma, A \vdash \bot}{\Gamma \vdash \neg A}$$

$$\text{Right} \oplus 1: \frac{\Gamma \vdash A}{\Gamma \vdash A \oplus B} \qquad \text{Right} \oplus 2: \frac{\Gamma \vdash B}{\Gamma \vdash A \oplus B} \qquad \text{Left} \oplus: \frac{\Gamma, A \vdash C \qquad \Gamma, B \vdash C}{\Gamma, A \oplus B \vdash C}$$

Figure 2.7: Sequent calculus of multiplicative additive tensorial logic, TENS

In contrast to linear logic, there is a good notion of games for tensorial logic, as polarities are this time entrenched in the sequent calculus via the negation. If a formula is negated, that is, is of the form $\neg F$, then it is negative, otherwise it is positive. This time, the double negation is not equal to the identity, and can be faithfully modelled through the addition of two moves. That is, the De-Morgan equation $\neg\neg A = A$ does not hold.

2.3.1 Equivalence of proofs in tensorial logic

The equivalence of proofs in tensorial logic is strongly similar to the one in linear logic, except one central difference; the use of $\neg$. In linear logic, due to the one-sided presentation, we are not concerned about where the focus of the proof is: to any formula present in the sequent either a positive or a negative rule might be applied. Here, the split of the sequent in two parts, with the need of a non-involutive operator to switch from one side to the other, is breaking the symmetry. The focus is now the right-hand-side of the sequent, and changing focus comes at a price: the double use of the negation operator. This changes fundamentally the structure of the proof invariants.

To our knowledge, the equivalence relation between proofs of tensorial logic has never been formally presented. However, it is clear from the idea underlying its inception that these shall seemingly be the same as those coming from linear logic, with the distinction coming from the double-sided form of the sequent and the use of a non involutive negation. We first tackle the multiplicative fragment.

left 1 - left 1

$$\frac{\frac{\Gamma \vdash A}{\Gamma, 1_a \vdash A}}{\Gamma, 1_a, 1_b \vdash A} \sim \frac{\frac{\Gamma \vdash A}{\Gamma, 1_b \vdash A}}{\Gamma, 1_a, 1_a \vdash A}$$

left 1 - left ⊗

$$\frac{\frac{\Gamma, A, B \vdash C}{\Gamma, A \otimes B \vdash C}}{\Gamma, A \otimes B, 1 \vdash C} \sim \frac{\frac{\Gamma, A, B \vdash C}{\Gamma, A, B, 1 \vdash C}}{\Gamma, A \otimes B, 1 \vdash C}$$

left 1 - right ⊗

$$\frac{\frac{\Gamma \vdash A}{\Gamma, 1 \vdash A} \qquad \Delta \vdash B}{\Gamma, 1, \Delta \vdash A \otimes B} \sim \frac{\frac{\Gamma \vdash A \qquad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}}{\Gamma, \Delta, 1 \vdash A \otimes B} \sim \frac{\Gamma \vdash A \qquad \frac{\Delta \vdash B}{\Delta, 1 \vdash B}}{\Gamma, 1, \Delta \vdash A \otimes B}$$

left ⊗ - left ⊗

$$\frac{\frac{\Gamma, A, B, C, D, \vdash E}{\Gamma, A \otimes B, C, D, \vdash E}}{\Gamma, A \otimes B, C \otimes D, \vdash E} \sim \frac{\frac{\Gamma, A, B, C, D, \vdash E}{\Gamma, A, B, C \otimes D, \vdash E}}{\Gamma, A \otimes B, C \otimes D, \vdash E}$$

left ⊗ - right ⊗

$$\frac{\frac{\Gamma, A, B \vdash C}{\Gamma, A \otimes B \vdash C} \qquad \Delta \vdash D}{\Gamma, A \otimes B, \Delta \vdash C \otimes D} \sim \frac{\frac{\Gamma, A, B \vdash C \qquad \Delta \vdash D}{\Gamma, A, B, \Delta \vdash C \otimes D}}{\Gamma, A \otimes B, \Delta \vdash C \otimes D}$$

right ⊗ - right ⊗ none

The multiplicative fragment against the negation.

left 1 - left ¬

$$\frac{\frac{\Gamma \vdash A}{\Gamma, 1 \vdash A}}{\Gamma, 1, \neg A \vdash \bot} \sim \frac{\frac{\Gamma \vdash A}{\Gamma, \neg A \vdash \bot}}{\Gamma, \neg A, 1 \vdash \bot}$$

left 1 - right ¬

$$\frac{\frac{\Gamma, A \vdash \bot}{\Gamma, A, 1 \vdash \bot}}{\Gamma, 1, \vdash \neg A} \sim \frac{\frac{\Gamma A \vdash \bot}{\Gamma \vdash \neg A}}{\Gamma, 1 \vdash \neg A}$$

left ⊗ - left ¬

$$\frac{\dfrac{\Gamma, A, B \vdash C}{\Gamma, A \otimes B \vdash C}}{\Gamma, A \otimes B, \neg C \vdash \bot} \sim \frac{\dfrac{\Gamma, A, B \vdash C}{\Gamma, A, B, \neg C \vdash \bot}}{\Gamma, A \otimes B, \neg C \vdash \bot}$$

left ⊗ - right ¬

$$\frac{\dfrac{\Gamma, A, B, C \vdash \bot}{\Gamma, A \otimes B, C \vdash \bot}}{\Gamma, A \otimes B \vdash \neg C} \sim \frac{\dfrac{\Gamma, A, B, C \vdash \bot}{\Gamma, A, B, \bot \vdash \neg C}}{\Gamma, A \otimes B, \bot \vdash \neg C}$$

right ⊗ - left / right ¬ No permutation

We now address the purely additive fragment :

left 0 - left 0 $\dfrac{}{\Gamma, 0_a, 0_b \vdash A}\text{Left } 0_a : \sim \dfrac{}{\Gamma, 0_a, 0_b \vdash A}\text{Left } 0_b :$

left 0 - right ⊕ $\dfrac{\dfrac{}{\Gamma, 0 \vdash A}\text{Left } 0 :}{\Gamma, 0 \vdash A \oplus B} \sim \dfrac{}{\Gamma, 0 \vdash A \oplus B}\text{Left } 0 :$ and similarly for $\oplus_2$.

left 0 - left ⊕

$$\frac{\dfrac{}{\Gamma, 0, A \vdash C}\,0 \qquad \dfrac{\pi_2}{\Gamma, 0, B \vdash C}}{\Gamma, 0, A \,\&\, B \vdash C} \sim \frac{}{\Gamma, 0, A \,\&\, B \vdash C}\,0 \sim \frac{\dfrac{\pi_1}{\Gamma, 0, A \vdash C} \qquad \dfrac{}{\Gamma, 0, B \vdash C}\,0}{\Gamma, 0, A \oplus B \vdash C}$$

left ⊕ - left ⊕

$$\frac{\dfrac{\Gamma, B, C \vdash E \qquad \Gamma, B, D \vdash E}{\Gamma, B, C \oplus D \vdash E} \qquad \dfrac{\Gamma, A, C \vdash E \qquad \Gamma, A, D \vdash E}{\Gamma, A, C \oplus D \vdash E}}{\Gamma, A \oplus B, C \oplus D \vdash E}$$

$$\sim$$

$$\frac{\dfrac{\Gamma, A, C \vdash E \qquad \Gamma, B, C \vdash E}{\Gamma, A \oplus B, C \vdash E} \qquad \dfrac{\Gamma, A, D \vdash E \qquad \Gamma, B, D \vdash E}{\Gamma, A \oplus B, D \vdash E}}{\Gamma, A \oplus B, C \oplus D \vdash E}$$

left ⊕ - right $\oplus_1$

$$\frac{\dfrac{\Gamma, A \vdash C}{\Gamma, A \vdash C \oplus D} \qquad \dfrac{\Gamma, B \vdash C}{\Gamma, B \vdash C \oplus D}}{\Gamma, A \oplus B \vdash C \oplus D} \sim \frac{\dfrac{\Gamma, A \vdash C \qquad \Gamma, B \vdash C}{\Gamma, A \oplus B \vdash C}}{\Gamma, A \oplus B \vdash C \oplus D}$$

and similarly for $\oplus_2$

right ⊕ vs right ⊕ none.

Additive against negation :

left 0 - left ¬

$$\dfrac{\dfrac{}{\Gamma, 0, \vdash A}\text{Left } 0 :}{\Gamma, 0, \neg A \vdash \bot} \sim \dfrac{}{\Gamma, 0, \neg A \vdash \bot}\text{Left } 0 :$$

left 0 - right ¬

$$\dfrac{\dfrac{}{\Gamma, 0, A \vdash \bot}\text{Left } 0 :}{\Gamma, 0 \vdash \neg A} \sim \dfrac{}{\Gamma, 0 \vdash \neg A}\text{Left } 0 :$$

left oplus - left ¬

$$\dfrac{\dfrac{\Gamma, A \vdash C \qquad \Gamma, B \vdash C}{\Gamma, A \oplus B \vdash C}}{\Gamma, A \oplus B, \neg C \vdash \bot} \sim \dfrac{\dfrac{\Gamma, A \vdash C}{\Gamma, A, \neg C \vdash A} \qquad \dfrac{\Gamma, B \vdash C}{\Gamma, B, \neg C \vdash \bot}}{\Gamma, A \oplus B, \neg C \vdash \bot}$$

left oplus - right ¬

$$\dfrac{\dfrac{\Gamma, A, C \vdash \bot \qquad \Gamma, B, C \vdash \bot}{\Gamma, A \oplus B, C \vdash \bot}}{\Gamma, A \oplus B, \vdash \neg C} \sim \dfrac{\dfrac{\Gamma, A, C \vdash \bot}{\Gamma, A \vdash \neg C} \qquad \dfrac{\Gamma, B, C \vdash \bot}{\Gamma, B, \vdash \neg C}}{\Gamma, A \oplus B \vdash \neg C}$$

right oplus - left/right ¬ none

And finally multiplicative against additive.

left 0 - left 1

$$\dfrac{\dfrac{}{\Gamma, 0 \vdash A}\text{Left 0:}}{\Gamma, 0, 1, \vdash A} \sim \dfrac{}{\Gamma, 0, 1 \vdash A}\text{Left } 0 :$$

left 0 - left ⊗

$$\dfrac{\dfrac{}{\Gamma, A, B, 0 \vdash C}\text{Left 0:}}{\Gamma, A \otimes B, 0 \vdash C} \sim \dfrac{}{\Gamma, A \otimes B, 0 \vdash C}\text{Left 0:}$$

left 0 - right ⊗

$$\dfrac{\dfrac{}{\Gamma, 0 \vdash A}\text{Left 0:} \qquad \Delta \vdash B}{\Gamma, \Delta, 0 \vdash A \otimes B} \sim \dfrac{}{\Gamma, \Delta, 0 \vdash A \otimes B}\text{Left } 0 :$$

right ⊕ -left 1

$$\dfrac{\dfrac{\Gamma, \vdash A}{\Gamma, 1 \vdash A}}{\Gamma, 1 \vdash A \oplus B} \sim \dfrac{\dfrac{\Gamma \vdash A}{\Gamma \vdash A \oplus B}}{\Gamma, 1 \vdash A \oplus B}$$

right ⊕ - left ⊗

$$\dfrac{\dfrac{\Gamma, A, B, \vdash C}{\Gamma, A \otimes B \vdash C}}{\Gamma, A \otimes B \vdash C \oplus D} \sim \dfrac{\dfrac{\Gamma, A, B \vdash C}{\Gamma, A, B \vdash C \oplus D}}{\Gamma, A, B \vdash C \oplus D}$$

right ⊕ - right ⊗ none.

left ⊕ - left 1

$$\dfrac{\dfrac{\Gamma, A, \vdash C \qquad \Gamma, B \vdash C}{\Gamma, A \oplus B \vdash C}}{\Gamma, A \oplus B, 1 \vdash C} \sim \dfrac{\dfrac{\Gamma, A, \vdash C}{\Gamma, A, 1 \vdash C} \qquad \dfrac{\Gamma, B \vdash C}{\Gamma, B, 1 \vdash C}}{\Gamma, A \oplus B, 1 \vdash C}$$

left ⊕ - left ⊗

$$\dfrac{\dfrac{\Gamma, A, B, C \vdash E \qquad \Gamma, A, B, D \vdash E}{\Gamma, A, B, C \oplus D \vdash E}}{\Gamma, A \otimes B, C \oplus D \vdash E} \sim \dfrac{\dfrac{\Gamma, A, B, C \vdash E}{\Gamma, A \otimes B, C \vdash E} \qquad \dfrac{\Gamma, A, B, D \vdash E}{\Gamma, A \otimes B, D \vdash E}}{\Gamma, A \otimes B, C \oplus D \vdash E}$$

left ⊕ - right ⊗

$$\dfrac{\dfrac{\pi}{\Gamma \vdash C} \qquad \dfrac{\Delta, A \vdash D \qquad \Delta, B \vdash D}{\Delta, A \oplus B \vdash D}}{\Gamma, \Delta, A \oplus B \vdash C \otimes D} \sim \dfrac{\dfrac{\dfrac{\pi}{\Gamma \vdash C} \qquad \Delta, A \vdash D}{\Gamma, \Delta, A \vdash C \otimes D} \qquad \dfrac{\dfrac{\pi}{\Gamma \vdash C} \qquad \Delta, B \vdash D}{\Gamma, \Delta, B \vdash C \otimes D}}{\Gamma, \Delta, A \oplus B \vdash C \otimes D}$$

2.3.2 Focussing in tensorial logic

Tensorial logic somehow acts as weakly focused linear logic: there is a focus, the right-hand-side of the sequent, and the sequent rules can be both applied to the focus or the left-hand part of the sequent. Just as one can strengthen weakly focused linear logic to get focused linear logic, one can strengthen the sequent calculus of tensorial logic to get focused tensorial logic.

A formula A is said to be in **negative** if $A = \neg A'$. It is **positive** otherwise. We write $\mathcal{P}$ for a list of positive formulas, $\mathcal{N}$ a list of negative formulas, $\mathcal{X}$ a list of positive atomic formulas, P, Q are formulas of either negative or positive polarity. The sequents are of the shape $\mathcal{P}, \mathcal{N}, \mathcal{X} \vdash P$.

The sequent calculus of **focussed tensorial logic** focussed tensorial logic $\mathsf{TENS_{Foc}}$ is as follows:

$$\dfrac{}{A \vdash A}\text{ Ax} \qquad \dfrac{}{\vdash 1}\text{ Right 1}$$

$$\dfrac{\mathcal{P}, \mathcal{N}, \mathcal{X}, \vdash P}{\mathcal{P}, \mathcal{N}, \mathcal{X}, 1 \vdash P}\text{ Left 1} \qquad \dfrac{}{\mathcal{P}, \mathcal{N}, \mathcal{X}, 0 \vdash P}\text{ Left 0}$$

$$\dfrac{\mathcal{N}, \mathcal{X} \vdash P}{\mathcal{N}, \neg P, \mathcal{X} \vdash \bot}\text{ Left }\neg \qquad \dfrac{\mathcal{N}, P, \mathcal{X} \vdash \bot}{\mathcal{N}, \mathcal{X} \vdash \neg P}\text{ Right }\neg$$

$$\dfrac{\mathcal{P}, \mathcal{N}, \mathcal{X}, P, Q \vdash R}{\mathcal{P}, \mathcal{N}, \mathcal{X}, P \otimes Q \vdash R}\text{ Left }\otimes \qquad \dfrac{\mathcal{N}, \mathcal{X} \vdash P \qquad \mathcal{N}', \mathcal{X}' \vdash Q}{\mathcal{N}, \mathcal{N}', \mathcal{X}, \mathcal{X}' \vdash P \otimes Q}\text{ Right }\otimes$$

$$\dfrac{\mathcal{P}, \mathcal{N}, \mathcal{X}, P \vdash R \qquad \mathcal{P}, \mathcal{N}, \mathcal{X}, Q \vdash R}{\mathcal{P}, \mathcal{N}, \mathcal{X}, P \oplus Q \vdash R}\text{ Left }\oplus \qquad \dfrac{\mathcal{N}, \mathcal{X} \vdash P}{\mathcal{N}, \mathcal{X} \vdash P \oplus Q}\text{ Right }\oplus_2$$

$$\dfrac{\mathcal{N}, \mathcal{X} \vdash Q}{\mathcal{N}, \mathcal{X} \vdash P \oplus Q}\text{ Right }\oplus_1 :$$

The focused sequent calculus prevents a rule from being applied on the right-hand-side as long as all the formulas on the left-hand-side of the sequent are not negative. So basically, a proof in focused tensorial logic goes as follows : it starts by decomposing all the formulas on the left-hand-side of the sequent until the left list is only composed of negative formulas and atomic variables. At this point, no more operations can be performed on the left-hand-side of the sequent. This is the asynchronous phase. Now begins the synchronous phase. The proof decomposes the formula on the right-hand-side until it reaches a negative formula. This formula is then brought back to the left-hand-side by the right negation rule. Then, another asynchronous phase begins, that will decompose the newly brought formula on the left-hand-side until the outcomes are all either negative or atomic formulas. Then, begins another synchronous phase, that starts by choosing a new focus, a new formula for the right-hand-side of the sequent, by applying a right negation.

This decomposition of the proof into a sequence of two phases corresponds to the decomposition of the "plays", in terms of games, into sequences of O and P-moves. The O-moves correspond to the asynchronous phases, the P to the synchronous ones.

We have the following theorem, that is a simple extension of its sibling for linear logic, and whose proof is analogous to the one in the case of linear logic, that can be found in [11, 59], and will not be repeated here.

Theorem 2.7. *Every proof of* $\Gamma \vdash A$ *of tensorial logic is equivalent to a proof of* $\Gamma \vdash A$ *in focused tensorial logic.*

Let us note however, that the system is different in nature compared to its focused counterpart in linear logic. Indeed, in linear logic, we could impose that the proof starts without focus, that is, with the right-hand-side of the sequent being empty. However, here, it is not the case. Let us note however that in the case where we start with a empty right-hand-side sequent, then the two systems behave seemingly similarly.

2.3.2.1 Global connectives

Just as in linear logic, we can reorganize the sequent calculus to a much simpler system, by noticing the existence of a normal form, stemming from the following isomorphisms:

$$A \otimes (B \oplus C) \simeq (A \otimes B) \oplus (A \otimes C) \tag{2.1}$$

$$(A \oplus B) \otimes C \simeq (A \otimes C) \oplus (B \otimes C) \tag{2.2}$$

The isomorphism and its inverse of the first equation 2.1 are presented below:

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A \qquad B \vdash B}{A, B \vdash A \otimes B}}{A, B \vdash (A \otimes B) \oplus (A \otimes C)} \qquad \dfrac{\dfrac{A \vdash A \qquad C \vdash C}{A, C \vdash A \otimes C}}{A, C \vdash (A \otimes B) \oplus (A \otimes C)}}{A, B \oplus C \vdash (A \otimes B) \oplus (A \otimes C)}}{A \otimes (B \oplus C) \vdash (A \otimes B \oplus (A \otimes C)}$$

$$\dfrac{\dfrac{A \vdash A \qquad \dfrac{B \vdash B}{B \vdash B \oplus C}}{A \otimes B \vdash A \otimes (B \oplus C)} \qquad \dfrac{A \vdash A \qquad \dfrac{C \vdash C}{C \vdash B \oplus C}}{A \otimes C \vdash A \otimes (B \oplus C)}}{(A \otimes B) \oplus (A \otimes C) \vdash A \otimes (B \oplus C)} \; .$$

Working with these isomorphisms allows us to characterise the sets of morphisms by working with their normal forms, as in the case of linear logic. We define the following rewriting system:

$$A \otimes (B \oplus C) \rightsquigarrow (A \otimes B) \oplus (A \otimes C)$$
$$(A \oplus B) \otimes C \rightsquigarrow (A \otimes C) \oplus (B \otimes C)$$
$$A \rightsquigarrow B \Rightarrow F[A] \rightsquigarrow F[B]$$

and call normal form a formula that cannot be further rewritten. To each formula of tensorial logic is asscociated a unique normal form. Then given two formulas A, B and their associated normal form $\bar{A}$ and $\bar{B}$, we then have $C(A, B) \simeq C(\bar{A}, \bar{B})$, where C is the category of tensorial logic formulas and proofs between them, considered up to equivalence. Furthermore, just as in linear logic, due to the associativity isomorphisms:

$$A \oplus (B \oplus C) \simeq (A \oplus B) \oplus C$$
$$A \otimes (B \otimes C) \simeq (A \otimes C) \otimes C$$

the order in which side to side tensors (and sums) appear in the formula does not matter. Therefore, we consider formulas up to associativity isomorphisms equivalence, and write $\bigotimes$ for a sequence of $\otimes$, $\bigoplus$ for a sequence of $\oplus$. A formula F is said to be in distributive-associative normal form if it originates from the following syntax:

$$\begin{aligned} N &::= \neg F \mid X \mid 1 \\ P &::= \bigoplus_i (\bigotimes_j N_{i,j}) \\ F &::= N \mid P \end{aligned}$$

For instance, $X \otimes X$ is translated in this setting into $(\bigoplus(\bigotimes(X)(X)))$. Note that, despite the perhaps misleading notation, N, P do not precisely correspond to negative and positive formulas, since atomic formulas are positive but, for simplicity, are considered N in the above syntax.

Now, let us study how a proof of focused tensorial logic behaves when the sequent is in normal form. We start with a sequent $\Gamma \vdash P$, where Γ, considered as a $\bigotimes$ of formulas, is considered in normal form: $\Gamma = \bigoplus_i (\bigotimes_j A_{i,j})$. The first step consists in a global left $\bigoplus$-rule:

$$\frac{\bigotimes_j A_{1,j} \vdash P \qquad ... \qquad \bigotimes_j A_{i,j} \vdash P \qquad ... \qquad \bigotimes_n A_{n,j} \vdash P}{\bigoplus_i (\bigotimes_j A_{i,j}) \vdash P} \text{Left} \bigoplus$$

Focussing on one of the branch, the proof now decomposes the $\bigotimes$, through a left global $\bigotimes$ rule.

$$\frac{A_{i,1}, .., A_{i,j}, .., A_{m,j} \vdash P}{\bigotimes_j A_{i,j} \vdash P} \text{Left} \bigotimes$$

Now, if some of the $A_{i,1}$ are 1, then the proof proceeds with a global left 1-rule, of the following shape.

$$\frac{B_1, ..., B_n \vdash P}{B_1, ..., B_n, 1_1, ..., 1_m \vdash P} \text{ Left } 1_m$$

This ends the asynchronous phase, that will be represented through a opponent move. It is then player's turn to play, and so begins the synchronous phase. Now, there are two cases to consider. Either P is $\bot$, and the proof is going to proceeds with the choice of a focus, embodied by a left negation rule. Or it is a negated formula, and the proof proceeds with a right negation. The cases where it is a atomic formula are dealt with below. The general case consists in P being of the form $\bigoplus(\bigotimes D_{i,j})$.

As each B_i is not a tensor nor a sum of sub-formulas, it is either a negation or an atomic formula. In the case where $P = \bot$, the synchronous phase start by picking one of the negated formula B_i, of the shape $\neg C$.

$$\frac{\Gamma \vdash C}{\Gamma, \neg C \vdash \bot} \text{ Left } \neg$$

Now C is of the shape $\bigoplus_i \bigotimes_j (D_{i,j})$. We now are in a similar position as when P was originally of this form. Hence the first step consists in a global right $\bigoplus$ rule:

$$\frac{\Gamma \vdash (\bigotimes D_{i,j})}{\Gamma \vdash \bigoplus_i (\bigotimes_j D_{i,j})} \bigoplus$$

followed by a global right tensor rule.

$$\frac{\Gamma_1 \vdash D_{i,1} \qquad ... \qquad \Gamma_j \vdash D_{i,j} \qquad ... \qquad \Gamma_n \vdash D_{i,n}}{\Gamma_1, .., \Gamma_i, ..., \Gamma_n \vdash \bigotimes B_{i,j}} \bigotimes$$

Finally, we apply the necessary rules for all the branches we created. If the formula $D_{i,j}$ is negated ($D_{i,j} = \neg E$), then we apply a right negation rule.

$$\frac{\Gamma, E \vdash \bot;}{\Gamma \vdash \neg E}$$

If, $\Gamma_i = \emptyset$, and $B_{i,j} = I$, then we apply the right I rule. Finally, in the case where $B_{i,j}$ is a positive atomic formula then so must be Γ_i, and we apply the axiom rule.

So, we can present the following proof-system, called $\mathsf{TENS}_{\mathsf{foc-glob}}$. This time, we decompose the formulas into three sets, depending on what is their principal connective.

$$\begin{aligned} N &= \neg F \mid X \mid 1 \\ P &= \bigoplus_i R_i \mid 0 \\ R &= \bigotimes_i N_i \\ F &::= N \mid P \end{aligned}$$

where $X \in \mathsf{TVar}$, In the following, $\mathcal{N}$ denotes a list of N-formulas without 1, $\mathcal{N}'$ a list of N-formulas, N, R, P, F are formulas as above.

$$\frac{}{X \vdash X}\ \text{Ax} \qquad \frac{\mathcal{N} \vdash F}{\mathcal{N}, \neg F \vdash \bot}\ \text{Left}\ \neg \qquad \frac{\mathcal{N}, F \vdash \bot}{\mathcal{N} \vdash \neg F;}\ \text{Right}\ \neg$$

$$\frac{N_1, ..., N_m \vdash F}{\bigotimes_i N_i \vdash F}\ \text{left}\ \bigotimes$$

$$\frac{\mathcal{N}_1 \vdash N_1 \quad ... \quad \mathcal{N}_i \vdash N_i \quad ... \quad \mathcal{N}_n \vdash N_n}{\mathcal{N}_1, ..., \mathcal{N}_n \vdash \bigotimes_i N_i}\ \text{Right}\ \bigotimes$$

$$\frac{\mathcal{N} \vdash R_k}{\mathcal{N} \vdash \bigoplus_{i=1..n} R_i}\ \text{Right}\ \bigoplus_{k,n}$$

$$\frac{\mathcal{N}, R_1 \vdash F \quad ... \quad \mathcal{N}, R_i \vdash F \quad ... \quad \mathcal{N}, R_n \vdash F}{\mathcal{N}, \bigoplus_{i=1..n} R_i \vdash F}\ \text{Left}\ \bigoplus$$

$$\frac{}{\vdash 1}\ \text{Right 1} \qquad \frac{\mathcal{N} \vdash F}{\mathcal{N}, 1_1, ..., 1_m \vdash F}\ \text{Left}\ 1_m \qquad \frac{}{\mathcal{N}', 0 \vdash F;}\ \text{Left}\ O$$

As explained above, the following theorem holds.

Theorem 2.8. *Every proof $\pi : \Gamma \vdash F$ of* TENS *is equivalent to a proof $\pi' : \bar{\Gamma} \vdash \bar{F}$ of* $\mathsf{MALL}_{\mathsf{foc-glob}}$, *where $\bar{\Gamma}$ is the distributive-associative normal form of $\bigotimes \Gamma$, and $\bar{F}$ the distributive-associative normal form of F.*

This system can be once again reformulated into a system with one global connective, that is a mix of ⊕ and ⊗. A major theorem in the case of tensorial logic is that this proof-system characterises exactly the equivalence classes of proofs.

Theorem 2.9. *Two proofs of* $\mathsf{TENS}_{\mathsf{glob-foc}}$ *are equivalent if and only if they are equal.*

Hence, we are able to precisely characterise the proofs invariants of tensorial logic. The proof of the theorem consists in noticing that no permutation rule can be applied to proofs presented with global connectives. However, this is not the case for linear logic, as different choices of focus entails different proofs invariants in tensorial logic, but not in linear logic. More on this will be presented in the section below.

2.3.2.2 Tensorial linear lambda calculus

The goal of this section is to establish a lambda-calculus for multiplicative tensorial logic. We restrict our attention to multiplicative tensorial logic. The terms of our calculus are as follows:

Types	$TY \ni T, U ::= X \mid 1 \mid \neg T \mid T \otimes U$
	where $X \in \mathsf{TVar}$.
Terms	$TE \ni t, u ::= x \mid \star \mid tu \mid \neg_x.t \mid t \otimes u \mid \mathsf{let}\ z\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ t$
	where $\neg_x$ is a binder that binds x.
Typing context	$\Gamma ::= \emptyset \mid x : T, \Gamma$

We write $x : T, \Gamma$ for $\{x : T\} \cup \Gamma$, with x not already appearing in Γ. The typing rules of our terms are presented in figure 2.8, where we write $\perp$ for the type $\neg 1$.

$$\frac{}{\vdash \star : 1} \qquad \frac{\Gamma \vdash t : T}{\Gamma, x : 1 \vdash t : T}$$

$$\frac{}{x : X \vdash x : X} \qquad \frac{\Gamma \vdash t : T \qquad x : T, \Delta \vdash u : U}{\Gamma, \Delta \vdash u[t/x] : U}$$

$$\frac{\Gamma, x : U \vdash t : \perp}{\Gamma \vdash \neg x.t : \neg U} \qquad \frac{\Gamma \vdash t : T}{\Gamma, f : \neg T \vdash ft : \perp}$$

$$\frac{\Gamma \vdash t : T \qquad \Delta \vdash u : U}{\Gamma, \Delta \vdash t \otimes u : T \otimes U} \qquad \frac{\Gamma, x : T, y : U \vdash v : V}{\Gamma, z : T \otimes U \vdash \mathsf{let}\ z\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ v}$$

Figure 2.8: Formation rules for tensorial linear lambda calculus

These correspond precisely to the terms of the Linear Lambda Calculus LLC [91], where we have restricted the terms of type $T \multimap U$ to the case where $U = \perp$. In that case, we write $\neg T$ for $T \multimap \perp$. Indeed, one can see tensorial logic as the fragment of intuitionistic linear logic where the isomorphism between $\mathcal{C}(A \otimes B, C)$ and $\mathcal{C}(A, B \multimap C)$ holds only when $C = \perp$.

Just as in LLC, the term formation is constrained by the linearity of the typing judgement. In particular :

$$x_1 : A_1, ..., x_k : A_k \vdash t : A$$

implies that, each x_i that is not of type a tensor product of units, occurs exactly once freely within t.

Furthermore, we write $\mathsf{let}\ u\ \mathsf{be}\ (x \otimes y \otimes z)\ \mathsf{in}\ v$ for either $\mathsf{let}\ u\ \mathsf{be}\ (x \otimes w)\ \mathsf{in}\ \mathsf{let}\ w\ \mathsf{be}\ (y \otimes z)\ \mathsf{in}\ v$ or its right associative analogous. To furthermore shorten the notation, we write $\neg(x_1, ..., x_n)$. for $\neg x.(\ \mathsf{let}\ x\ \mathsf{be}\ (x_1, ..., x_n)\ \mathsf{in}\ .)$.

We define the free variables of a term as follows:

$$
\begin{array}{ll}
\mathsf{bv}(x) = \emptyset & \mathsf{fv}(x) = x \\
\mathsf{bv}(\bullet) = \emptyset & \mathsf{fv}(\bullet) = \emptyset \\
\mathsf{bv}(tu) = \mathsf{bv}(t) \cup \mathsf{bv}(u) & \mathsf{fv}(tu) = \mathsf{fv}(t) \cup \mathsf{fv}(u) \\
\mathsf{bv}((\neg(x_1, ..., x_n).t) = \{x_1, ..., x_n\} \cup \mathsf{bv}(t) & \mathsf{fv}(\neg(x_1, .., x_n).t) = \mathsf{fv}(t) \setminus \{x_1, .., x_n\} \\
\mathsf{bv}(t \otimes u) = \mathsf{bv}(t) \cup \mathsf{bv}(u) & \mathsf{fv}(t \otimes u) = \mathsf{fv}(t) \cup \mathsf{fv}(u) \\
\mathsf{bv}(\mathsf{let}\ z\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ t) = \mathsf{bv}(t) \cup \{x, y\} & \mathsf{fv}(\mathsf{let}\ z\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ t) = \{z\} \cup \mathsf{fv}(t) \setminus \{x, y\}
\end{array}
$$

The rules for β-reduction are:

$$
\begin{array}{rcl}
(\neg_x.t)u & \longrightarrow_\beta & t[u/x] \\
\mathsf{let}\ t \otimes u\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ v & \longrightarrow_\beta & v[t/x, u/y].
\end{array}
$$

where the substitution is defined as follows. We define $t[u/x]$ by induction on the structure of the term t:

- $x[u/x] = u$
- $y[u/x] = y$
- $(t_1 \otimes t_2)[u/x] = (t_1[u/x]) \otimes (t_2[u/x])$
- $t_1 t_2[u/x] = t_1[u/x] t_2[u/x]$
- $(\neg y.t)[u/x] = \neg y.(t[u/x])$ if $y \neq t$ and $y \notin \mathsf{fv}(u)$.
- $(\ \mathsf{let}\ y\ \mathsf{be}\ y_1 \otimes y_2\ \mathsf{in}\ t)[u \otimes v/y] = t[u/y_1, v/y_2]$
- $(\ \mathsf{let}\ y\ \mathsf{be} y_1 \otimes y_2\ \mathsf{in}\ t)[u/x] =\ \mathsf{let}\ y\ \mathsf{be} y_1 \otimes y_2\ \mathsf{in}\ (t[u/x])$ if $x \neq y, y_1, y_2$ and $y, y_1, y_2 \notin \mathsf{fv}(u)$.

The conflictual cases $x = y$ or $x = y_1, y_2$ are resolved using α-equivalent terms. We give below an example of a term derivation. We can check that the λ-term:
$\neg(x, w, f, g, h).h(x \otimes \neg u.(g(w \otimes \neg v.(f(u \otimes v)))))) : \neg(X \otimes Y \otimes (\neg(Z \otimes W) \otimes (\neg(Y \otimes \neg W)) \otimes (\neg(X \otimes \neg Z)))$
is well-typed.

$$
\begin{array}{c}
u : Z \vdash u : Z \qquad v : W \vdash v : W \\ \hline
u : Z, v : W \vdash u \otimes v : Z \otimes W \\ \hline
u : Z, v : W, f : \neg(Z \otimes W) \vdash f(u \otimes v) : \bot \\ \hline
u : Z, f : \neg(Z \otimes W) \vdash \neg v.(f(u \otimes v)) : \neg W \qquad w : Y :\vdash w : Y \\ \hline
w : Y, u : Z, f : \neg(Z \otimes W) \vdash w \otimes \neg v.(f(u \otimes v)) : Y \otimes \neg W \\ \hline
w : Y, u : Z, f : \neg(Z \otimes W), g : \neg(Y \otimes \neg W) \vdash g(w \otimes \neg .v(f(u \otimes v))) : \bot \\ \hline
w : Y, f : \neg(Z \otimes W), g : \neg(Y \otimes \neg W) \vdash \neg u.(g(w \otimes \neg v.(f(u \otimes v)))) : \neg Z \qquad x : X \vdash x : X \\ \hline
x : X, w : Y, f : \neg(Z \otimes W), g : \neg(Y \otimes \neg W) : x \otimes \neg u.(g(w \otimes \neg v.(f(u \otimes v)))) : X \otimes \neg Z \\ \hline
x : X, w : Y, f : \neg(Z \otimes W), g : \neg(Y \otimes \neg W), h : \neg(X \otimes \neg Z) \vdash h(x \otimes \neg u.(g(w \otimes \neg v.(f(u \otimes v))))) \\ \hline
y : X \otimes Y \otimes (\neg(Z \otimes W) \otimes (\neg(Y \otimes \neg W)) \otimes (\neg(X \otimes \neg Z)) \vdash \mathsf{let}\ y\ \mathsf{be}(x, w, f, g, h)\ \mathsf{in}\ h(x \otimes \neg u.(g(w \otimes \neg v.(f(u \otimes v))))) \\ \hline
\vdash \neg(x, w, f, g, h).h(x \otimes \neg u.(g(w \otimes \neg v.(f(u \otimes v)))))) : \neg(X \otimes Y \otimes (\neg(Z \otimes W) \otimes (\neg(Y \otimes \neg W)) \otimes (\neg(X \otimes \neg Z)))
\end{array}
$$

2.4 Linear logic and tensorial logic: a translation

Relying on polarities, one can devise several translations of linear logic into tensorial logic. The games for linear logic defined in [66] come from a translation from linear logic into tensorial logic named " focalised translation" in [71] where it was originally presented.

Using if necessary the equations ruling $(.)^\perp$, each formula of linear logic is equal to one where the $(.)^\perp$ connective is only applied to atomic propositional variable. Using this form, we recall that we can split the formulas of linear logic into two sets of formulas,

$$
\begin{array}{rcl}
P & = & X \mid F \otimes F' \mid F \oplus F' \mid 0 \mid 1 \\
N & = & X^{\perp} \mid F \parr F' \mid F \,\&\, F' \mid \top \mid \bot \,;
\end{array}
$$

where F, F' denotes any formula of MALL. This way, One can notice, that, as expected, the negation of a positive formula is a negative formula and conversely. Given a positive formula of linear logic P, its translation into tensorial logic $(P)^F$ is defined as follows, where $* \in \{\oplus, \otimes\}$:

$$
\begin{array}{rcl}
P = X \Rightarrow (P)^F & = & X \\
P = O \Rightarrow (P)^F & = & 0 \\
P = 1 \Rightarrow (P)^F & = & 1 \\
P = P_1 * P_2 \Rightarrow (P)^F & = & (P_1)^F * (P_2)^F \\
P = N_1 \otimes P_2 \Rightarrow (P)^F & = & \neg(N_1^{\perp})^F \otimes (P_2)^F \\
P = P_1 \otimes N_2 \Rightarrow (P)^F & = & (P_1)^F \otimes \neg(N_2^{\perp})^F \\
P = N_1 \otimes N_2 \Rightarrow (P)^F & = & \neg(N_1^{\perp})^F \otimes \neg(N_2^{\perp})^F
\end{array}
$$

Proposition 2.10.

- *There is an syntactic equivalence between the cut-free proofs $\vdash \Gamma; \Pi$ in weakly focused linear logic and the proof of $\neg(\mathcal{P})^F, (\mathcal{N}^{\perp})^F, \mathcal{X}^{\perp} \vdash (\Pi)^F$ in tensorial logic, where $\mathcal{P}$ is the subset of positive formulas of Γ, $\mathcal{X}$ its subset of negative atomic formulas, $\mathcal{N}$ its subset of negative formulas that are not in $\mathcal{X}$. Furthermore $(\Pi)^F = \bot$ if (Π) is empty, $(P)^F$ in the case where $(\Pi) = P$ is positive, and $\neg(M^{\perp})^F$ in the case where $\Pi = M$ is negative.*
- *There is a syntactic equivalence between the cut-free proofs of $\vdash \mathcal{P}, \mathcal{N}, \mathcal{X};$ in focused linear logic and the proofs of $\neg(\mathcal{P})^F, (\mathcal{N}^{\perp})^F, \mathcal{X}^{\perp} \vdash \bot$ in focused tensorial logic.*
- *There is a syntactic equivalence between the cut-free proofs of $\vdash \mathcal{P}, \mathcal{N}, \mathcal{X};$ in focused linear logic with global connectives and the proofs of $\neg(\mathcal{P})^F, (\mathcal{N}^{\perp})^F, \mathcal{X}^{\perp} \vdash \bot$ in focused tensorial logic with global connectives.*

This theorem can be summed up in the following table.

$$
\begin{array}{ccc}
\mathsf{MALL} \searrow & & \\
\mathsf{MALL}_{\mathsf{wfoc}} & \simeq & \mathsf{TENS} \\
\downarrow & & \downarrow \\
\mathsf{MALL}_{\mathsf{foc}} & \simeq & \mathsf{TENS}_{\mathsf{foc}} \\
\downarrow & & \downarrow \\
\mathsf{MALL}_{\mathsf{foc-glob}} & \simeq & \mathsf{TENS}_{\mathsf{foc-glob}}
\end{array}
$$

The proposition is at the core of the definability result for games for linear logic. If one can establish a definability result for games for tensor logic, then to each strategy will correspond a proof in tensorial logic and, finally, a proof of linear logic. However, to two strategies might correspond equivalent proofs. This issue has to be tackled via a quotient on strategies.

Proof. The proof is done by induction on the rules of the proofs. We remind that Π denotes either a unique formula of a empty-sequent of MALL. $\vdash A^{\perp}; A$ of $\mathsf{MALL}_{\mathsf{wfoc}}$ is translated into $((A^{\perp})^{\perp})^F \vdash (A)^F = A \vdash A$, that is, the axiom rule of tensor logic.

Weakly Focused Linear Logic	Tensorial Logic
$\dfrac{}{\vdash A^{\perp}; A}\,\text{Ax}$	$\dfrac{}{A \vdash A}\,\text{Ax}$
$\dfrac{\vdash \mathcal{P}, \mathcal{N}, \mathcal{X}; P}{\vdash \mathcal{P}, \mathcal{N}, \mathcal{X}, P;}\,\text{Foc}$	$\dfrac{\neg(\mathcal{P})^F, (\mathcal{N}^{\perp})^F, \mathcal{X}^{\perp} \vdash (P)^F}{\neg(\mathcal{P})^F, \neg(P)^F, (\mathcal{N}^{\perp})^F, \mathcal{X}^{\perp} \vdash \perp}\,\text{Left } \neg$
$\dfrac{\mathcal{P}, \mathcal{N}, M, \mathcal{X};}{\mathcal{P}, \mathcal{N}, \mathcal{X}; M}\,\text{Unfoc}$	$\dfrac{\neg(\mathcal{P})^F, (\mathcal{N}^{\perp})^F, (M^{\perp})^F, (\mathcal{X})^{\perp} \vdash \perp}{\neg(\mathcal{P})^F, (\mathcal{N}^{\perp})^F, (\mathcal{X})^{\perp} \vdash \neg(M^{\perp})^F}\,\text{Right } \neg$
$\dfrac{\vdash \mathcal{P}, \mathcal{N}, M, N, \mathcal{X}; \Pi}{\vdash \mathcal{P}, \mathcal{N}, M \parr N, \mathcal{X}; \Pi}\,\parr$	$\dfrac{\neg(\mathcal{P})^F, (\mathcal{N}^{\perp})^F, (M^{\perp})^F, (N^{\perp})^F, \mathcal{X}^{\perp} \vdash (\Pi)^F}{\neg(\mathcal{P})^F, (\mathcal{N}^{\perp})^F, (M^{\perp} \otimes N^{\perp})^F, \mathcal{X}^{\perp} \vdash (\Pi)^F}\,\text{Left } \otimes$ since $((M \parr N)^{\perp})^F = (M^{\perp} \otimes N^{\perp})^F = ((M^{\perp})^F \otimes (N^{\perp})^F))$
$\dfrac{\vdash \mathcal{P}, \mathcal{N}, P, Q, \mathcal{X}; \Pi}{\vdash \mathcal{P}, \mathcal{N}, P \parr Q, \mathcal{X}; \Pi}\,\parr$	$\dfrac{\neg(\mathcal{P})^F, (\mathcal{N}^{\perp})^F, \neg(P)^F, \neg(Q)^F \mathcal{X}^{\perp} \vdash (\Pi)^F}{\neg(\mathcal{P})^F, (\mathcal{N}^{\perp})^F, \neg(P)^F \otimes \neg(Q)^F, \mathcal{X}^{\perp} \vdash (\Pi)^F}\,\text{Left } \otimes$ since $((P \parr Q)^{\perp})^F = (P^{\perp} \otimes Q^{\perp})^F = (\neg(P)^F \otimes \neg(Q)^F)$

$$\dfrac{\vdash \mathcal{P}, \mathcal{N}, \mathcal{X}; Q \qquad \vdash \mathcal{P}', \mathcal{N}', \mathcal{X}; P}{\vdash \mathcal{P}, \mathcal{P}', \mathcal{N}, \mathcal{N}' \mathcal{X}, \mathcal{X}'; P \otimes Q}\,\otimes$$

$$\dfrac{\neg(\mathcal{P})^F, (\mathcal{N}^{\perp})^F, \mathcal{X}^{\perp} \vdash (P)^F \qquad \neg(\mathcal{P}')^F, ((\mathcal{N}')^{\perp})^F, \mathcal{X}'^{\perp} \vdash (Q)^F}{\neg(\mathcal{P})^F, \neg(\mathcal{P}')^F, (\mathcal{N}^{\perp})^F, (\mathcal{N}'^{\perp})^F, \mathcal{X}^{\perp}, \mathcal{X}'^{\perp} \vdash (P)^F \otimes (Q)^F}\,\text{Right } \otimes$$

since $(P \otimes Q)^F = P^F \otimes Q^F$.

$$\dfrac{\vdash \mathcal{P}, \mathcal{N}, \mathcal{X}; M \qquad \vdash \mathcal{P}', \mathcal{N}', \mathcal{X}; N}{\vdash \mathcal{P}, \mathcal{P}', \mathcal{N}, \mathcal{N}', \mathcal{X}, \mathcal{X}'; M \otimes N}\,\otimes$$

$$\dfrac{\neg(\mathcal{P})^F, (\mathcal{N}^{\perp})^F, \mathcal{X}^{\perp} \vdash \neg(M^{\perp})^F \qquad \neg(\mathcal{P}')^F, (\mathcal{N}'^{\perp})^F, \mathcal{X}'^{\perp} \vdash \neg(N^{\perp})^F}{\neg(\mathcal{P})^F, \neg(\mathcal{P}')^F, (\mathcal{N}^{\perp})^F, (\mathcal{N}'^{\perp})^F, \mathcal{X}^{\perp}, \mathcal{X}'^{\perp} \vdash \neg(M^{\perp})^F \otimes \neg(N^{\perp})^F}\,\otimes$$

since $(M \otimes N)^F = (\neg(M^{\perp})^F \otimes \neg(N^{\perp})^F)$.

$\dfrac{\vdash \mathcal{P}, \mathcal{N}, \mathcal{X}; \Pi}{\vdash \mathcal{P}, \mathcal{N}, \perp, \mathcal{X}; \Pi}\,\perp$	$\dfrac{\neg(\mathcal{P})^F, (\mathcal{N}^{\perp})^F, \mathcal{X}^{\perp} \vdash (\Pi)^F}{\neg(\mathcal{P})^F, (\mathcal{N}^{\perp})^F, 1, \mathcal{X}^{\perp} \vdash (\Pi)^F}\,\text{Left } 1$ since, as $\perp$ is negative, $(\perp^{\perp})^F = (1)^F = 1$.

$$\frac{}{\vdash 1}\ 1 \qquad\qquad \frac{}{\vdash ; 1}\ 1$$

$$\frac{\vdash \mathcal{P}, \mathcal{N}, P, \mathcal{X}; \Pi \qquad \vdash \mathcal{P}, \mathcal{N}, Q, \mathcal{X}; \Pi}{\vdash \mathcal{P}, \mathcal{N}, P \mathbin{\&} Q, \mathcal{X}; \Pi}\ \&$$

$$\frac{\neg(\mathcal{P})^F, \neg(P^F), (\mathcal{N}^\perp)^F, \mathcal{X}^\perp \vdash \Pi \qquad \neg(\mathcal{P})^F, \neg(Q^F), (\mathcal{N}^\perp)^F, \mathcal{X}^\perp \vdash \Pi}{\neg(\mathcal{P})^F, \neg(P^F) \oplus \neg(Q^F), (\mathcal{N})^F, \mathcal{X}^\perp \vdash (\Pi)^F}\ \text{Left} \oplus$$

since $((P \mathbin{\&} Q)^\perp)^F = (P^\perp \oplus Q^\perp)^F = \neg(P^F) \oplus \neg(Q^F)$

$$\frac{\vdash \mathcal{P}, \mathcal{N}, A, \mathcal{X}; \Pi \qquad \vdash \mathcal{P}, \mathcal{N}, B, \mathcal{X}; \Pi}{\vdash \mathcal{P}, \mathcal{N}, M \mathbin{\&} N, \mathcal{X}; \Pi}\ \&$$

$$\frac{\neg(\mathcal{P})^F, (\mathcal{N}^\perp)^F, (M^\perp)^F, \mathcal{X}^\perp \vdash \Pi \qquad \neg(\mathcal{P})^F, (\mathcal{N}^\perp)^F, (N^\perp)^F, \mathcal{X}^\perp \vdash \Pi}{\neg(\mathcal{P})^F, (\mathcal{N})^F, (M^\perp)^F \oplus (N^\perp)^F, \mathcal{X}^\perp \vdash \Pi}\ \text{Right} \oplus_1$$

since $((M \mathbin{\&} N)^\perp)^F = (M^\perp \oplus N^\perp)^F = (M^\perp)^F \oplus (N^\perp)^F$

$$\frac{\vdash \mathcal{P}, \mathcal{N}, \mathcal{X}; P}{\vdash \mathcal{P}, \mathcal{N}, \mathcal{X}; P \oplus Q}\ \oplus_1 \qquad \frac{\neg(\mathcal{P})^F, (\mathcal{N}^\perp)^F, \mathcal{X}^\perp \vdash P^F}{\neg(\mathcal{P})^F, (\mathcal{N}^\perp)^F, \mathcal{X}^\perp \vdash P^F \oplus Q^F}\ \text{Right} \oplus_1$$

since $(P \oplus Q)^F = P^F \oplus Q^F$

$$\frac{\vdash \mathcal{P}, \mathcal{N}, \mathcal{X}; P}{\vdash \mathcal{P}, \mathcal{N}, \mathcal{X}^\perp; P \oplus M}\ \oplus_1 \qquad \frac{\neg(\mathcal{P})^F, (\mathcal{N}^\perp)^F, \mathcal{X}^\perp \vdash P^F}{\neg(\mathcal{P})^F, (\mathcal{N}^\perp)^F, \mathcal{X}^\perp \vdash P^F \oplus \neg(M^\perp)^F}\ \text{Right} \oplus_1$$

since $(P \oplus M)^F = P^F \oplus \neg(M^\perp)^F$

$$\frac{\vdash \mathcal{P}, \mathcal{N}, \mathcal{X}; M}{\vdash \mathcal{P}, \mathcal{N}, \mathcal{X}; M \oplus P}\ \oplus_1 \qquad \frac{\neg(\mathcal{P})^F, (\mathcal{N}^\perp)^F, \mathcal{X}^\perp \vdash \neg(M^\perp)^F}{\neg(\mathcal{P})^F, (\mathcal{N}^\perp)^F, \mathcal{X}^\perp \vdash \neg(M^\perp)^F \oplus \neg P^F}\ \text{Right} \oplus_1$$

since $(M \oplus P)^F = \neg(M^\perp)^F \oplus P^F$

$$\frac{\vdash \mathcal{P}, \mathcal{N}, \mathcal{X}; M}{\vdash \mathcal{P}, \mathcal{N}, \mathcal{X}; M \oplus N}\ \oplus_1 \qquad \frac{\neg(\mathcal{P})^F, (\mathcal{N}^\perp)^F, \mathcal{X}^\perp \vdash \neg(M^\perp)^F}{\neg(\mathcal{P})^F, (\mathcal{N}^\perp)^F, \mathcal{X}^\perp \vdash \neg(M^\perp)^F \oplus \neg(N^\perp)^F}\ \text{Right} \oplus_1$$

since $(M \oplus N)^F = \neg(M^\perp)^F \oplus \neg(N^\perp)^F$

$$\frac{}{\vdash \mathcal{P}, \mathcal{N}, X, \top; \Pi}\ \top \qquad \frac{}{\neg(\mathcal{P})^F, (\mathcal{N}^\perp)^F, \mathcal{X}^\perp, 0 \vdash \Pi}\ \text{Left } 0$$

The focused, and global focused systems are dealt with on a equal footing. □

2.4.1 Reverse translation and quotient

Tensorial logic is essentially linear logic with non-involutive negation. Therefore, there is also a canonical translation from tensorial into linear logic. We denote by $(.)^I$ this mapping, and it is defined by induction on formulas by:

$$(0)^I = 0 \qquad (1)^I = 1 \qquad (\bot)^I = \bot$$
$$(A \otimes B)^I = (A)^I \otimes (B)^I \qquad (A \oplus B)^I = A^I \oplus B^I$$
$$(\neg A)^I = ((A)^I)^\bot$$

The mapping $(.)^I$ extends to proofs. It maps a proof of tensorial logic into a proof of $\mathsf{MALL}_{\mathsf{wfoc}}$. If we denote by **TensLog** the pre-category (that is, we are not concerned about composition) of tensorial logics, which has formulas of tensorial logic as objects and proofs as morphisms between them, and **MALLwfoc** its sibling for weakly focused linear logic, then we have a pair of mappings :

$$\mathbf{TensLog} \underset{(.)^F}{\overset{(.)^I}{\rightleftarrows}} \mathbf{MALLwfoc}$$

The mapping $(.)^I$ can be lifted through $[.]$ (that is, the map that to each proof associates its equivalence class) and hence leads to a functor. That is, given two proofs π, π' of tensorial logic, $\pi \sim \pi' \Rightarrow (\pi)^I \sim (\pi')^I$. Equivalently, there is a map $(.)^{[I]}$ from the category of invariant of proofs of tensor logic, to the category of invariants of proofs of linear logic, such that $([\pi]_{\mathsf{TensLog}})^{[I]} = [(\pi)^I]_{\mathsf{MALL}}$. However, this is not the case for $(.)^F$, as it does not factor through $[.]_{\mathsf{MALL}}$. That is, it does not respect the equivalences of proofs of linear logic: two proofs of linear logic that are equivalent might be mapped to non-equivalent proofs of tensorial logic. We will display an example below.

These mappings somehow define an embedding, that is $((.)^F)^I = \mathsf{id}$. On the other hand, we do not have $((.)^I)^F = \mathsf{id}$, since the mapping $(.)^I$ will forget about any possible double negations. For instance $((\neg\neg A)^I)^F = ((A)^I)^F$.

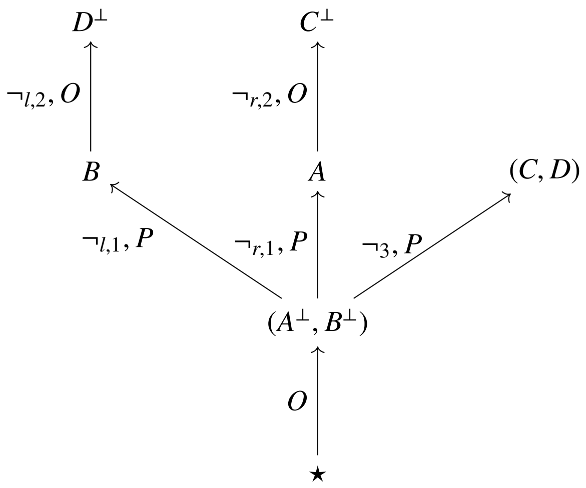
Let us display why some non-equivalent proofs of tensorial logic might be mapped to equivalent proofs of linear logic. For instance, let us consider the following two proofs of the same proposition of tensorial logic.

$$\begin{prooftree}
\AxiomC{}
\UnaryInfC{$A \vdash A$}
\AxiomC{}
\UnaryInfC{$B \vdash B$}
\AxiomC{}
\UnaryInfC{$C \vdash C$}
\AxiomC{}
\UnaryInfC{$D \vdash D$}
\BinaryInfC{$C, D \vdash C \otimes D$}
\RightLabel{$\neg_3$}
\UnaryInfC{$C, D, \neg(C \otimes D) \vdash \bot$}
\RightLabel{$\neg_{l,2}$}
\UnaryInfC{$C, \neg(C \otimes D) \vdash \neg D$}
\BinaryInfC{$B, C\neg(C \otimes D) \vdash B \otimes \neg D$}
\RightLabel{$\neg_{l,1}$}
\UnaryInfC{$B, C\neg(C \otimes D), \neg(B \otimes \neg D) \vdash \bot$}
\RightLabel{$\neg_{r,2}$}
\UnaryInfC{$B, \neg(C \otimes D), \neg(B \otimes \neg D) \vdash \neg C$}
\BinaryInfC{$A, B, \neg(C \otimes D), \neg(B \otimes \neg D) \vdash A \otimes \neg C$}
\RightLabel{$\neg_{r,1}$}
\UnaryInfC{$A, B, \neg(C \otimes D), \neg(B \otimes \neg D), \neg(A \otimes \neg C) \vdash \bot$}
\end{prooftree}$$

and

$$
\begin{prooftree}
\AxiomC{}
\UnaryInfC{$B \vdash B$}
\AxiomC{}
\UnaryInfC{$A \vdash A$}
\AxiomC{}
\UnaryInfC{$C \vdash C$}
\AxiomC{}
\UnaryInfC{$D \vdash D$}
\BinaryInfC{$C, D \vdash C \otimes D$}
\RightLabel{$\neg_3$}
\UnaryInfC{$\neg(C \otimes D), D, C, \vdash \bot$}
\RightLabel{$\neg_{r,2}$}
\UnaryInfC{$\neg(C \otimes D), D \vdash \neg C$}
\BinaryInfC{$A, \neg(C \otimes D), D \vdash A \otimes \neg C$}
\RightLabel{$\neg_{r,1}$}
\UnaryInfC{$A, \neg(C \otimes D), \neg(A \otimes \neg C), D \vdash \bot$}
\RightLabel{$\neg_{l,2}$}
\UnaryInfC{$A, \neg(C \otimes D), \neg(A \otimes \neg C) \vdash \neg D$}
\BinaryInfC{$A, B, \neg(C \otimes D), \neg(A \otimes \neg C) \vdash (B \otimes \neg D)$}
\RightLabel{$\neg_{l,1}$}
\UnaryInfC{$A, B, \neg(C \otimes D), \neg(B \otimes \neg D), \neg(A \otimes \neg C) \vdash \bot$}
\end{prooftree}
$$

These proofs differ in the order in which they will explore each branch. The first one will start exploring $\neg(A \otimes \neg C)$, whereas the second will focus on $(B \otimes \neg C)$. To make it clear, we expose below the (simplified) arena for the proofs. At this stage, one only needs to know that an arena is a tree whose edges are labelled "player" or "opponent", and the moves are equipped with a notion of cell, that corresponds to propositional variables.



The two proofs above correspond to two different strategies, as the first one reacts to the initial O-move by playing the P-move $\neg_{l,1}$, whereas the second reacts by playing $\neg_{r,1}$. However, if we translate the first proof into linear logic we obtain:

$$
\begin{prooftree}
\AxiomC{}
\UnaryInfC{$A \vdash A$}
\AxiomC{}
\UnaryInfC{$B \vdash B$}
\AxiomC{}
\UnaryInfC{$C \vdash C$}
\AxiomC{}
\UnaryInfC{$D \vdash D$}
\BinaryInfC{$C, D \vdash C \otimes D$}
\UnaryInfC{$C, D, (C^\perp \parr D^\perp) \vdash \bot$}
\UnaryInfC{$C, (C^\perp \parr D^\perp) \vdash D^\perp$}
\BinaryInfC{$B, C, (C^\perp \parr D^\perp) \vdash B \otimes D^\perp$}
\UnaryInfC{$B, C, (C^\perp \parr D^\perp), (B^\perp \otimes D^\perp) \vdash \bot$}
\UnaryInfC{$B, (C^\perp \parr D^\perp), B^\perp \parr D \vdash C^\perp$}
\BinaryInfC{$A, B, (C^\perp \parr D^\perp), B^\perp \parr D \vdash A \otimes C^\perp$}
\UnaryInfC{$A, B, (C^\perp \parr D^\perp), (B^\perp \parr D), (A^\perp \parr C) \vdash \bot$}
\end{prooftree}
$$

that we can simplify, by forgetting the back and forth around $\vdash$, to form:

$$\dfrac{\dfrac{}{\vdash A^\perp, A} \qquad \dfrac{\dfrac{}{\vdash B^\perp, B} \qquad \dfrac{\dfrac{}{\vdash C^\perp, C} \qquad \dfrac{}{\vdash D^\perp, D}}{\vdash C, D, C^\perp \otimes D^\perp}}{\vdash B^\perp, C \otimes D, C^\perp, B \otimes D^\perp}}{\vdash A^\perp, B^\perp, C \otimes D, B \otimes D^\perp, A \otimes C^\perp}$$

for the first one. Doing an equal translation, we get

$$\dfrac{\dfrac{}{\vdash B^\perp, B} \qquad \dfrac{\dfrac{}{\vdash A^\perp, A} \qquad \dfrac{\dfrac{}{\vdash C^\perp, C} \qquad \dfrac{}{\vdash D^\perp, D}}{\vdash C, D, C^\perp \otimes D^\perp}}{\vdash A^\perp, C \otimes D, D^\perp, A \otimes C^\perp}}{\vdash A^\perp, B^\perp, C \otimes D, B \otimes D^\perp, A \otimes C^\perp}$$

for the second one. But these two proofs are equivalent in linear logic. The reason why was highlighted in the previous section 2.2.1.1. The same argument explains why $(.)^F$ is not a functor. Indeed, the two above proofs of linear logic are equivalent, but are mapped by $(.)^F$ to two different proofs of tensorial logic. Therefore, the mapping $(.)^F$ does not respect the invariants of linear logic; it is only a syntactic translation.

This is (partly) related to the Blass problem, as explained in [65]. In tensorial logic, the order in which the tensors are played is made precise. However, in linear logic, the two player moves $\neg_{l,1}, \neg_{r,1}$ happen concomitantly. Therefore, there are two ways to circumvent this problem (from a semantical point of view). Either we consider that the two moves happen concomitantly, that is, the strategy is now a concurrent strategy, being able to play several moves at once. Or, we force the strategy to choose an order, even arbitrary, and then we forget about it by imposing a quotient on strategies, that relates two strategies whose order on those moves differ.

2.5 Categorical models

Along this thesis, we will often refer to the categorical structure of our models, and their properties. For a complete introduction to categories, we refer to [58] or [91]. The purpose of this section is to briefly remind what are categorical models of linear and tensorial logic. To keep it short, we do not give a full description, and refer the reader to [67] [71] for more. To start, we briefly remind the definition of a monoidal category.

Definition 2.11. *A category $\mathbf{C}$ is said to be* **monoidal** *if there exists a functor $\otimes : \mathbf{C} \times \mathbf{C} \to \mathbf{C}$, a unit object I, two unit natural isomorphisms $\rho : A \otimes I \simeq A$ and $\lambda : I \otimes A \simeq A$, and a natural associativity isomorphism $\alpha_{a,b,c} : A \otimes (B \otimes C) \simeq (A \otimes B) \otimes C$ satisfying certain coherence diagrams, which we omit.*

Models of linear logic and tensorial logic consist of **symmetric** monoidal categories, that are monoidal categories equipped with a natural associativity isomorphism $s_{A,B} : A \otimes B \to B \otimes A$, such that $s^{-1}_{A,B} = s_{B,A}$, and subject to some conditions that we do not present here. If both categories behave the same way with regards to the tensor, they present a schism when it comes to the negation.

We start with tensorial logic. A categorical model of multiplicative tensorial logic is referred to as a dialogue category. A proof $\pi : \Gamma = F_1, ..., F_n \vdash A$ of tensorial logic is interpreted as a morphism $F_1 \otimes ... \otimes F_n \to A$.

Definition 2.12. *A* ***dialogue category*** *C is a symmetric monoidal category together with a tensorial negation $\neg : C \to C^{op}$ such that there is a family of bijections natural in A, B, C :*

$$C(A \otimes B, \neg C) \simeq C(A, \neg(B \otimes C))$$

In particular, writing $\bot$ for $\neg I$, this entails the following:

$$C(A \otimes B, \bot) \simeq C(A, \neg B).$$

A dialogue category defines a model of multiplicative additive tensorial logic if it has coproducts that distribute over the monoidal products. Given a dialogue category, an easy way to obtain such coproducts is to perform the family construction, that consists in taking lists of objects as new objects. More details can be found in [71] [8]. On the other hand, models of multiplicative linear logic turn out to be star-autonomous. A dialogue category is star-autonomous only when the double-negation monad is the identity. Star-autonomous categories notably have an additional property of closure. We say that a symmetric monoidal category is **closed** when, for every object B the functor $\_ \otimes B$ admits a right adjunct denoted $B \multimap \_$. It then satisfies the following correspondence:

$$C(A \otimes B, C) \simeq C(A, B \multimap C)$$

Therefore, the left adjunct of the identity $A \multimap B \to A \multimap B$ is a morphism $(A \multimap B) \otimes A \to B$ called $\mathsf{eval}_{A,B}$. We can then obtain a morphism $A \to (A \multimap B) \multimap B$, using first the symmetry $(A \multimap B) \otimes A \simeq A \otimes (A \multimap B)$ and then the left to right isomorphism. This morphism is called the currying of the $\mathsf{eval}_{A,B}$ map. A proof $\pi : F_1, ..., F_n \vdash G_1, .., G_n$ of linear logic is interpreted as a morphism $F_1 \otimes ... \otimes F_n \to G_1 ⅋ .... ⅋ G_n$.

Definition 2.13. *A* ***star-autonomous*** *category is a symmetric monoidal category C with a dualising object $\bot$ such that the currying of the evaluation map $\mathsf{eval}_{A,\bot}$ is an isomorphism:*

$$(A \multimap \bot) \multimap \bot \simeq A.$$

In a star-autonomous category, we do notably have $(A)^{\perp} \simeq A \multimap \perp$. There are alternative presentations of star-autonomous categories, but all of them are equivalent. Among them, one of the most relevant consists defining them through a double monoidal structure. Indeed, star-autonomous have two monoidal structures, one coming from the tensor $\otimes$ and the second one coming from its negation. The original one models the tensor $\otimes$ from the logic, whereas the second one models the par $\parr$. If both tensors are interpreted the same way (that is, $\otimes = \parr$), then the star-autonomous category is said to be degenerate. Compact closed categories correspond precisely to the degenerate case, that is, they are star-autonomous categories with $\otimes = \parr$. We give the formal definition below.

Definition 2.14. *A* ***compact closed category*** *is a symmetric closed monoidal category in which every object A is assigned a dual $A^{\perp}$, a unit $\eta_A : I \to A^{\perp} \otimes A$ and a co-unit $\epsilon_A : A \otimes A^{\perp} \to I$ such that the following equalities hold :*

$$A \xrightarrow{\mathsf{Id}_A \otimes \eta_A} A \otimes A^{\perp} \otimes A \xrightarrow{\epsilon_A \otimes \mathsf{id}_A} A = \mathsf{id}_A$$
$$A^{\perp} \xrightarrow{\eta_A \otimes \mathsf{id}_A} A^{\perp} \otimes A \otimes A^{\perp} \xrightarrow{\mathsf{id}_A \otimes \epsilon_A} A^{\perp} = \mathsf{id}_A$$

It turns out that the assignment $A \to A^{\perp}$ can be turned into a contravariant functor. To each morphism $f : A \to B$, $f^{\perp} : B^{\perp} \to A^{\perp}$ is :

$$B^{\perp} \xrightarrow{\mathsf{id}_{B^{\perp}} \otimes \eta_A} B^{\perp} \otimes A^{\perp} \otimes A \xrightarrow{\mathsf{id}_{B^{\perp} \otimes A^{\perp}} \otimes f} B^{\perp} \otimes A^{\perp} \otimes B \xrightarrow{\mathsf{exchange}} B \otimes B^{\perp} \otimes A^{\perp} \xrightarrow{\eta_B \otimes \mathsf{id}_{A^{\perp}}} A^{\perp}$$

Therefore, in the sequel, we will simply present compact closed categories as monoidal categories together with a negation functor. It should be clear from context what the families η_A and ϵ_A are.

If star-autonomous categories form a sound model for multiplicative linear logic, one needs more structure for the additive part. To model it, coproducts and products are required for every two objects of the category, together with a initial and final object for the additive units. However it turns out that a star-autonomous category can not have one without the other: the negation of a product becoming a coproduct, and reversely. Therefore, it is enough to require only one of the two. By convention, we say that a categorical model of MALL is a star-autonomous category with products.

2.5.1 Free categories

Given a category C in **Cat**, its free S-category will be precisely defined in terms of the following universal property. Given U_S :**S-Cat** $\to$ **Cat**, the forgetful functor, and C a category, the free S-category of C consists of an object C_S of **S-Cat**, such that there exists a canonical functor $\epsilon_C : C \to U_S(C_s)$ that makes C a subcategory of $U_S(C_s)$, and furthermore such that for every S-category $\mathcal{D}$, and every functor $h : C \to U_S(\mathcal{D})$ there exists an arrow $h^{\flat} : C_s \to \mathcal{D}$ such that $U_S(h^{\flat}) \circ \epsilon_C = h$. This arrow $h^{\flat}$ is not necessarily unique, but is unique up to structure-preserving isomorphism. Diagrammatically:

$$\mathbf{Cat} \xleftarrow[U_S]{\qquad\qquad} \mathbf{S\text{-}cat}$$

$$\begin{array}{ccccc} & & U(\mathcal{D}) & \qquad\qquad & \mathcal{D} \\ & {}^{h}\nearrow & \uparrow{\scriptstyle U(h^\flat)} & & {\scriptstyle h^\flat}\uparrow \\ C & \xrightarrow[\epsilon_C]{} & U(C_S) & & C_S \end{array}$$

The free category could be explicitly described by means of algebraic combinations of generators and relations, but the interesting cases arise when a description can be given by a direct definition. Our interest will range from S being the free symmetric monoidal category to S being compact closed. We will furthermore give an explicit description of a compact category, and then star-autonomous categories with products and co-products. Unfortunately, those will not be the free-categories.

Chapter 3

Simple Nominal Models

Models of linear logic are often referred to as static, in the sense that they do not involve a notion of evolution or time. Unfortunately, no full completeness result has been achieved with statics models without using some additional categorical structure like di-natural transformations [16, 84], or similar 2-cateogorical tools [5] [60], in order to deal with atomic types. We here present basic adaptations of some of the most popular static models of linear logic to the nominal world, and highlight why nominal models are ideal to deal with resources and linearity. Our goal is to shift from a 2-categorical setting to a first-order one. In order to do so, one must be able to give to each formula, as denotation, a single object in the category. Therefore, we do not deal with abstract structures, but really concrete ones where each object can be formed from basic building blocks, corresponding to atomic formulas and units, and a finite number of categorical operations, that are the analogues of the connectives of the logic. We chose to model atomic variables by sets of names, that seem to be suitable candidates to represent the "elementary particles of logic" [67].

We construct gradually more and more complex categories that correspond to bigger fragments of linear logic. Our basic category will be a discrete category, where the only morphisms are the identities. We start by presenting its free symmetric monoidal category, and from it construct a compact closed category, that we briefly compare to the free one. This is the simplest model of multiplicative linear logic one might look for. To enrich it with additive structure, we move on to nominal relations. Finally, to keep the model from being degenerate, we introduce a notion of coherence that allows us to distinguish $\otimes$ from $\parr$. At this stage, one could either present coherence or hypercoherence spaces, and we choose the former as it refines the latter.

We recall that the denotation function is designated by $[\![.]\!]_{C,}$ from $\mathcal{L}$- formulas and proofs to C, where $\mathcal{L}$ is a fragment of linear logic and C is a category. Generally, C and $\mathcal{L}$ will be clear from the context, and we will simply write $[\![.]\!]$.

Definition 3.1. *The category* VAR *is the discrete category with objects* $X \in \mathsf{TVar}$*, and whose only morphisms are the identities.*

The logic associated with the category VAR is the "axiomatic" fragment:

$$\frac{}{X \vdash X}\text{Ax} \quad X \in \mathsf{TVar}$$

At this stage, the function $[\![.]\!]$ maps each type variable to its associated canonical object, and each proof $X \vdash X$ to the identity morphism.

In this section, we build a simple nominal model of propositional linear logic, using sorted names to represent atomic types. We construct it gradually. We sum up in this table the different categories we present, what logic they model, and what is their relationship to the logic and its invariants.

Category	Logic	Relation
VAR	Axiomatic Fragment	Free
NomLinList	$\otimes$ + Exchange Fragment	Free
NomLinPol	Compact closed logic	Fully complete
NomLinPol	MLL	Sound, degenerate
NomLinRel	$(\otimes, \oplus)$	Sound
NomLinRelPol	Compact closed logic with biproduct	Sound
NomLinRelPol	MALL	Sound, degenerate
NomHypCoh	MALL	Sound

Figure 3.1: List of categories of Chapter 3.

In the above table, we wrote "Free" to indicate that the category is the free S-category on VAR, where S is the appropriate categorical structure corresponding to the logic. That is, a category is **Free** if there is a correspondence between the morphisms of it and the invariants of proofs of the suitable logic.

3.1 Names and the free symmetric monoidal category

3.1.1 An introduction to nominal sets

We briefly recall some key notions of sorted nominal set theory, for a more complete introduction we advise looking at [81] or [82]. Let us denote $\mathsf{TVar} = X, Y, ...$ a countable infinite set of atomic types and let us fix a countable infinite family $(\mathbb{A}_X)_{X \in \mathsf{TVar}}$ of pairwise disjoint, countable infinite sets of **names**. We write $\mathsf{Perm}(\mathbb{A}_X)$ for the group of finite permutations of $\mathbb{A}_X$, that are, permutations that change only a finite number of elements We will denote names by $a, b, c....$ and name permutations by π. Also, (a, b) is the permutation swapping a and b. We will call $\mathbb{A}$ the set of all names : $\mathbb{A} = \biguplus_X \mathbb{A}_X$, and consider $\mathsf{Perm}(\mathbb{A}) = \bigoplus_X \mathsf{Perm}(\mathbb{A}_X)$, where $\biguplus$ is the disjoint union, and $\bigoplus$ the direct product. We call **name permutations** the elements of $\mathsf{Perm}(\mathbb{A})$.

Proposition 3.2. $(\mathsf{Perm}(\mathbb{A}), \circ, \mathsf{id})$ *forms a group. The inverse of each element is its inverse as a function.*

The proof is straightforward. We recall below the axioms of group actions.

Definition 3.3. *Given a group* $(\mathcal{G}, \circ, 1)$ *and a set* S*, a (left) group action from* $\mathcal{G}$ *to* S *is a function* $\mathcal{G} \times S \to S$ *denoted* $(g, x) \to (g \cdot x)$ *such that:*

- $g \cdot (g' \cdot x) = (g \circ g') \cdot x.$
- $1 \cdot x = x.$

In the future we will only consider left group actions and hence simply denote them group actions.

Definition 3.4. *A* $\mathsf{Perm}(\mathbb{A})$*-set* S *is a set* $|S|$ *together with a group action* $\mathsf{Perm}(\mathbb{A}) \times |S| \to |S|$.

We say that a subset $A \subseteq \mathbb{A}$ **supports** an element $x \in \mathcal{S}$, if $\forall \pi \in \mathsf{Perm}(\mathbb{A}).(\forall a \in A.\pi(a) = a) \Rightarrow \pi \cdot x = x$.

Definition 3.5. *A* ***nominal set*** $\mathcal{S}$ *is a* $\mathsf{Perm}(\mathbb{A})$*-set such that each element of* $\mathcal{S}$ *has a finite supporting set.*

A key property of finite supporting sets is that they intersect. That is, given x an element of a nominal set, $S, T \subseteq \mathsf{fin}\mathbb{A}$ such that S, T support x then $S \cap T$ supports x as well (see [90, p. 17] for a proof). Therefore, we can write $\nu(x)$ for the minimal supporting set of x, and call it the **support** of x. In the case where the element x is finite, the support is exactly the set of names it contains.

Given an element x in a nominal set, we denote $[x]$ its **orbit**, defined by $[x] = \{\pi \cdot x \mid \pi \in \mathsf{Perm}(\mathbb{A})\}$. Given two elements x, y of $\mathcal{S}$, we write $x \# y$ if they have disjoint support ($\nu(x) \cap \nu(y) = \emptyset$. We say that two elements x, y are **equivalent**, written $x \simeq y$ if there is a permutation π such that $\pi \cdot x = y$. Equivalently, two elements are equivalent if they have same orbit. Given a function between nominal sets $f : S \to T$, we define $(\pi \cdot f)$ by $(\pi \cdot f)(x) = \pi \cdot (f(\pi^{-1} \cdot x))$. This way, we equip the set of functions from S to T with a group action, given it the structure of a $\mathsf{Perm}(\mathbb{A})$-set. An element x is **equivariant** if $\forall \pi \in \mathsf{Perm}(\mathbb{A}). \pi \cdot x = x$. Equivalently, an element is equivariant if it has empty support. In particular, a function between two nominal sets is **equivariant** if $f(\pi \cdot x) = \pi \cdot f(x)$ for all x, π. In this case, one can establish that $\nu(f(x)) \subseteq \nu(x)$.

Definition 3.6. *Nominal sets and equivariant functions between them form a category, called* NSet.

We sometimes write nominal instead of equivariant. For instance, a nominal subset of a nominal set is a subset closed under nominal permutations.

Definition 3.7. *An element x of a nominal set has* ***strong support*** *if $\pi \cdot x = x \Rightarrow \pi \# x$.*

Alternatively, we say that the element x is strongly supported. One should note that the property above is actually an equivalence $\pi \cdot x = x \Leftrightarrow \pi \# x$, since the right to left direction is automaticly true. Strong support has been introduced in [90], where it was noted that it was a necessary condition to model adequately nominal programming languages. A simple example of an element with non strong support is the set $\{a, b\}$, since $(a, b) \cdot \{a, b\} = \{a, b\}$, though $\nu((a, b)) = \nu(\{a, b\})$. On the other hand, the list $a.b$ has strong support.

Definition 3.8. *A nominal set with strong support is a nominal set such that each of its element enjoys being strongly supported.*

In the sequel, the careful reader might notice that all the nominal sets we present have strong support, since they massively rely on lists.

3.1.2 Building the free symmetric monoidal category

Given a small discrete category, that is, a category where the only morphisms are the identities, one can construct the free symmetric monoidal category out of it using sorted names. The category NomLinList, that we present below, will be our first construction. In order to cope with the unit of the monoidal category, we introduce a distinctive element $\bullet$ of empty support. We use the word **atome** (in the literature, atom is a synonym for names, and this is why we preferred a french spelling) to refer to either a name or to $\bullet$. We overload the previous notation, and let a, b, c and variants range over the set of atomes. We say that two atomes are **name-distinct** (or simply distinct) if they have non-intersecting support. That is, two atomes a, b are distinct only if $a \# b$. If a list of atomes contains only name-distinct atoms, we say that this list is **separated**. We say that a nominal set is **1-orbit** if it possesses only one equivalence class relatively to name permutations, that is, for every two elements of the set x, y, it follows that $x \simeq y$. Therefore, given a 1-orbit nominal set S, we have $\forall x \in S. S = [x]$, that is, S is perfectly defined by each of its elements.

The objects of NomLinList will be set of separated lists L built out of the following grammar:

$$L ::= \bullet \;\mid\; a \;\mid\; L_1.L_2$$

where $a \in \mathbb{A}$ and $L_1 \# L_2$. We write $L_1.L_2$ for list concatenation and we abuse notation by writing "$\bullet$" and "a" for the one element list containing $\bullet$ and a respectively.

Definition 3.9. *The category* NomLinList *has:*

- *non-empty 1-orbit sets of finite non-empty separated lists of atomes as objects*
- *functions ϕ that are equivariant and linear as morphisms, in the sense that each resource in its antecedent should appear in its image.*
    - *ϕ is equivariant : $\pi \cdot (\phi(x)) = \phi(\pi \cdot x)$*

– *ϕ is linear :* $\nu(x) = \nu(\phi(x))$

The composition of morphisms is the standard composition of set-functions, and the identity function acts as the identity morphism. Both the equivariance condition and the linearity one compose, and the idendity morphism is equivariant and linear. For each $X \in \mathsf{TVar}$, we set $[\![X]\!] = \mathbb{A}_X$. This assigns to X a pool of names (resources) of sort X. We furthermore define $[\![I]\!] = \{\bullet\}$.

$\mathsf{NomLinList}$ is equipped with a monoidal product, that originates from a mix between separated product and concatenation of lists. We name it **separated concatenation**, denote it $\star$ and define it by the following operation :

$$X_1 \star X_2 = \{x_1.x_2 \mid x_1 \in X_1, x_2 \in X_2, x_1 \# x_2\}$$

As expected, we set $[\![A \otimes B]\!] = [\![A]\!] \star [\![B]\!]$. Note that any object of $\mathsf{NomLinList}$ is **well typed**, that is, it can be inductively generated from $I, [\![X]\!](X \in \mathsf{TVar})$ and the binary operation $\star$ defined below. Separated concatenation prevents a resource from appearing several times in an list, hence the linear condition really enforces linearity: there are no morphisms of type $X \to X \otimes X$, nor of type $X \to I$. Furthermore, the use of sorted names prevents incoherent typing, such as morphisms $X \to Y$. Note also that any element of empty support would do as a unit for the monoidal product, therefore, we take $I = \{\bullet\}$ for it, where $\nu(\bullet) = \emptyset$. The morphism $\rho_A : A \otimes I \to A$ is defined by $\rho_A(l.\bullet) = l$, where l is an element of A. $\lambda : I \otimes A \to A$ is defined dually, and the natural associativity isomorphism is the identity. The monoidal product acts like the cartesian product on morphisms, that is:

$$\phi \star \psi(L_1.L_2) = \phi(L_1).\psi(L_2).$$

From the equivariance of ϕ and ψ follows that $\phi(L_1)\#\psi(L_2)$ and therefore the monoidal product is well defined on functions.

Therefore, the category $\mathsf{NomLinList}$ is a model of the monoidal fragment of linear logic, defined below:

$$\frac{}{A \vdash A}\,\text{Ax} \qquad \frac{\Gamma \vdash A \qquad A, \Delta \vdash B}{\Gamma, \Delta \vdash B}\,\text{Cut}$$

$$\frac{\Gamma_1, A, B, \Gamma_2 \vdash C}{\Gamma_1, B, A, \Gamma_2 \vdash C}\,\text{Left exchange}$$

$$\frac{\Gamma \vdash A}{\Gamma, 1 \vdash A}\,\text{Left 1} \qquad \frac{}{\vdash 1}\,1$$

$$\frac{\Gamma \vdash A \qquad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}\,\text{Right}\otimes \qquad \text{Left}\otimes\,\frac{\Gamma, A, B \vdash C}{\Gamma, A \otimes B \vdash C}$$

Figure 3.2: The monoidal + exchange fragment of Linear Logic

3.1.2.1 Free semi-strict symmetric monoidal category

Definition 3.10.
- *A symmmtric monoidal category is semi-strict if the associator $\alpha : (A \otimes B) \otimes C \to (A \otimes B) \otimes C$ is the identity.*
- *A monoidal category is* ***strict*** *if the isomorphisms $\alpha_{A,B,C} : (A \otimes B) \otimes C \to A \otimes (B \otimes C)$, λ_A, ρ_A are identities (for all objects A, B, C of the category).*

With NomLinList defined, we can move on to prove it is the free semi-strict symmetric monoidal category over VAR. To simplify things, we do not write explicitly the forgetful functor U. The functor $\epsilon : \mathsf{VAR} \to \mathsf{NomLinList}$ is defined by mapping each variable X onto the set $\mathbb{A}_X$ viewed as a 1-orbit set of lists of length 1, and the identities to identity maps of NomLinList. Let $\mathcal{D}$ be a symmetric monoidal category, and F a functor $F : \mathsf{VAR} \to \mathcal{D}$. The symmetric monoidal functor $F^\flat : \mathsf{NomLinList} \to \mathcal{D}$ is defined on objects as follows:

- $F^\flat(I) = I_\mathcal{D}$
- $F^\flat(\mathbb{A}_X) = F(X)$, when $X \in TVar$.
- $F^\flat(A_1 \star A_2) = F^\flat(A_1) \otimes F^\flat(A_2)$

To define its action on morphisms, we need the following proposition, that can be found in [58], chapter XI.

Proposition 3.11. *Each symmetric monoidal category is equivalent (via strong monoidal functors) to a strict monoidal category.*

We recall that two categories are equivalent if there is a fully faithful functor from one to the other. Here, it is required that, furthermore, this functor is a strong monoidal one, that is $F(A \otimes B) \simeq F(A) \otimes F(B)$ and $F(I) \simeq I$, where I are the monoidal units of the categories.

More precisely, a semi-strict symmetric monoidal category is strongly equivalent to its strict subcategory, defined to be the restriction of the category to strict objects. An object X is strict if it is either I, or writing $X = X_1 \otimes ... \otimes X_n$, where each X_i is irreducible (that is, cannot be written as $X_i = Y_1 \otimes Y_2$), then $\forall i. X_i \neq I$. The strict subcategory forms a strict monoidal category, where given two objects X, Y, $X \otimes_{\mathsf{strict}} Y = X \otimes Y$ if $X, Y \neq I$, and $X \otimes_{\mathsf{strict}} Y = X$ if $Y = I$ or $X \otimes_{\mathsf{strict}} Y = Y$ if $X = I$.

Writing $F : C \to C_{\mathsf{strict}}$ and $G : C_{\mathsf{strict}} \to C$ for the functors of monoidal categories reflecting the equivalence, we define $\chi : \mathsf{Id}_C \to G \circ F$ the monoidal natural isomorphism coming from the equivalence. Then χ maps an object formed by tensoring atomic objects and units to the same object without the units (for instance, it sends $A \otimes I \otimes B$ to $A \otimes B$). Then, every morphism $f : A \to B$ factors as a morphism: $f : A \xrightarrow{\chi_A} A_{\mathsf{strict}} \xrightarrow{f_{\mathsf{strict}}} B_{\mathsf{strict}} \xrightarrow{(\chi_B)^{-1}} B$.

The standard construction of the free strict symmetric monoidal category can be found in [1], and is reminded here. Given a category C, the free strict symmetric monoidal category $\mathsf{SSMC}(C)$ adjoined to C has objects finite lists of objects of C and the morphisms are described as follows. $f : (A_1, ..., A_n) \to (B_1, ..., B_n)$ consists of a permutation $s_n \in \Sigma_n$, where Σ_n is the group of permutations of $\{1, .., n\}$, and a family $\{f_i \mid i \in [1, n]\}$ of morphisms of C such that $f_i : A_i \to B_{s_n(i)}$.

The monoidal product consists of the concatenation of lists, and the unit is the empty list. The composition of morphisms is defined through the combination of set-theoretic composition of permutations and composition of morphisms of C. Here, as we based our constructions on the category VAR that has only identities as morphisms, we can forget about the family f_i. The morphisms of SSMC(VAR), that we refer to simply as SSMC in the sequel, are simply permutations s_n such that for all $i \in [1, n]$, $A_i = B_{s_n(i)}$.

NomLinList and SSMC are equivalent, but not isomorphic. Indeed, in the list construction, the objects $A \otimes I$ and A are identical, whereas they differ in the nominal construction. However, SSMC corresponds precisely to the strict subcategory of NomLinList. Consequently, the action of morphisms of NomLinList on objects is uniquely defined by their permutation of objects coming from the symmetry of NomLinList. This can be presented as follows: for any objects $A = \bigotimes_i X_i$, $B = \bigotimes_i Y_i$, and any morphism $f : A \to B$, f_{strict} can clearly be seen as a permutation, by attributing to each atomic variable a natural number indicating its position in the formula:

$$\begin{array}{ccccccc} X & \otimes & Y & \otimes & Z & \otimes & W \\ 1 & & 2 & & 3 & & 4 \end{array}$$

and then noticing that f_{strict} will send a name in position i to a position j. For instance, the morphism f_{strict} below:

$$\begin{array}{ccccccccccccccc} A & \otimes & B & \otimes & C & \otimes & D & \to & W & \otimes & X & \otimes & Y & \otimes & Z \\ (a & \cdot & b & \cdot & c & \cdot & d) & \to & (c & \cdot & a & \cdot & b & \cdot & d) \end{array}$$

is equivalent to the permutation $(1, 2, 3)$. Therefore, to each morphism f of NomLinList can be assigned a permutation, by looking at the sole position of atomic variables inside the formula (that is, we do not pay attention to units), and how it acts on their associated names. For instance, to the morphism f:

$$\begin{array}{ccccccccccccccc} 1 & & & & & & 2 & \to & & & 1 & & & & 2 \\ X & \otimes & I & \otimes & I & \otimes & Y & \to & I & \otimes & Y & \otimes & I & \otimes & X \\ (a & \cdot & \bullet & \cdot & \bullet & \cdot & b) & \to & (\bullet & \cdot & b & \cdot & \bullet & \cdot & a) \end{array}$$

one can assign the permutation $(1, 2)$. The number associated to each type variable will be called its **location**. Equivalently, we also speak about the location of a name in the list. Moreover, composition of morphisms works as composition of permutations. These permutations can be built from the symmetry morphisms $s_{X,Y}$ and the identities by tensoring and composing them. Therefore, they are present in any symmetric monoidal category.

Based on this result, we can uncover the action of $F^\flat$ on a morphism $f : A \to B$. On **strict objects**, that, is, objects built from the atomic variables and the tensor product, without using the unit, its image $F^\flat(f)$ is the exact same permutation, this time applied to:

$$F^\flat(A_1) \otimes ... \otimes F^\flat(A_n) \to F^\flat(B_1) \otimes ... \otimes F^\flat(B_n).$$

Given $f : A \to B$, and its decomposition $\chi_A; f_{\mathsf{strict}}; \chi_B^{-1}$, its image $F^\flat(f) : F^\flat(A) \to F^\flat(B)$, is given by $\chi_{F^\flat(A)}; F^\flat(f_{\mathsf{strict}}); \chi^{-1}_{F^\flat(B)}$. One can straightforwardly check that $F^\flat(\mathsf{id}_A) = \mathsf{id}_{F^\flat(A)}$, and $F^\flat(f; g) = F^\flat(f); F^\flat(g)$, as the two structural morphisms χ^{-1}, χ in the middle will cancel themselves out.

Note that in order to obtain the free symmetric monoidal category, one should work with products instead of lists. This way, we would have keep the parentheses inside the elements. For instance, an object of $X \otimes (Y \otimes Z)$ would have been under the form $(x, (y, z))$ instead of being a simple list (x, y, z). Therefore, by working with lists, we obtain an *almost* perfect abstract representation of proofs, as we work up to associativity equivalence.

3.2 Traced monoidal category

Symmetric monoidal categories are models of the ⊗-part of linear logic. To incorporate the whole multiplicative structure, one needs star-autonomous categories. Compact closed categories are the simplest instance of such categories. Given a traced monoidal category, there is a standard (and free) completion of it to a compact closed category. We would like to apply this construction to our former category, in order to get a model of multiplicative linear logic. Therefore, we need to check that the category previously defined can be equipped with a trace, and, if so, present its definition.

Definition 3.12. *A* ***traced monoidal category*** *is a symmetric monoidal category equipped with a family of functions* $\mathsf{Tr}^U_{A,B} : C(A \otimes U, B \otimes U) \to C(A, B)$ *for all objects* A, B, U *of* C *satisfying the following properties:*

- *Given* $f : A \otimes U \to B \otimes U$ *and* $g : A' \to A$ *then:* $\mathsf{Tr}^U_{A,B}(f) \circ g = \mathsf{Tr}^U_{A',B}(f \circ (g \otimes \mathsf{Id}_U))$ *(Naturality 1).*
- *Given* $f : A \otimes U \to B \otimes U$ *and* $g : B \to B'$*, then* $g \circ \mathsf{Tr}^U_{A,B}(f) = \mathsf{Tr}^U_{A,B'}((g \otimes \mathsf{id}_U) \circ f)$ *(Naturality 2).*
- *Given* $f : A \otimes U \to B \otimes U'$ *and* $g : U' \to U$ *then:* $\mathsf{Tr}^U_{A,B}((\mathsf{Id}_B \otimes g) \circ f) = \mathsf{Tr}^{U'}_{A,B}(f \circ (\mathsf{Id}_A \otimes g))$ *(Dinaturality 3).*
- *Given* $f : A \otimes I \to B \otimes I$ *then* $\mathsf{Tr}^I_{A,B}(f) = f$*, and given* $g : A \otimes U \otimes V \to B \otimes U \otimes V$ *then:* $\mathsf{Tr}^{U \otimes V}_{A,B}(g) = \mathsf{Tr}^U_{A,B}(\mathsf{Tr}^V_{A \otimes U, B \otimes U}(g))$ *(Vanishing).*
- *Given* $f : A \otimes U \to B \otimes U$ *and* $g : W \to Z$ *then:* $g \otimes \mathsf{Tr}^U_{A,B}(f) = \mathsf{Tr}^U_{W \otimes A, Z \otimes B}(g \otimes f)$ *(Superposing).*
- *Given* $s_{A,B} : A \otimes B \to B \otimes A$ *the family of symmetry functions that comes from the symmetry of the monoidal product, then* $\mathsf{Tr}^U_{A,A}(s_{A,A}) = \mathsf{Id}_A$ *(Yanking).*

Note that the three first conditions only state that $\mathsf{Tr}^U_{A,B}$ is natural in its three parameters U, A, B. Surprisingly, even if one can easily uncover a trace for our category, it turns out that the resulting category is not the free traced category (see [1] for more details). The trace is not easily describable in terms of names and functions, therefore, we use the parallel established in the previous section between our morphisms and permutations.

To define the trace of $f : A \otimes U \to B \otimes U$, we focus on the associated permutation. We assume there are m locations in U, and n in A, B. The trace of a permutation is defined through the feedback operator. To $f : A \otimes U \to B \otimes U$, seen as a permutation, we define the permutation associated to $g = \mathsf{tr}(f)$ as follows. We take $g = g' \restriction \{1, .., n\}$ where g' is defined by: [1]

$$g'(i) = f(i) \text{ if } f(i) \leq n$$
$$g'(i) = g'(f(i) \text{ otherwise}$$

Since f is a finite, respecting types, permutation, we can easily check that the above is well-defined.

As proven in [1], the trace defined above indeed forms a trace on the category SSMC, that is, on the sub-strict category of $\mathsf{NomLinList}$. We can prove that this forms a trace on $\mathsf{NomLinList}$ using categorical arguments, by relying on the equivalence between the category $\mathsf{NomLinList}$ and its strict subcategory. The trace exposed above can be described in these new terms:

$$\mathsf{Tr}^U_{A,B,\ \mathsf{NomLinList}}(f) = \chi_A; (G \circ \mathsf{Tr}^{F(U)}_{F(A),F(B),\mathsf{SSMC}} \circ F(f)); \chi_B^{-1}$$

Lemma 3.13. *The family of functions* $\mathsf{Tr}^U_{A,B,\ \mathsf{NomLinList}} : C(A \otimes U, B \otimes U) \to C(A, B)$ *that to each* $f : A \otimes U \to B \otimes U$ *assigns* $\chi_A; (G \circ \mathsf{Tr}^{F(U)}_{F(A),F(B),\mathsf{SSMC}} \circ F(f)); \chi_B^{-1}$ *is a trace in* $\mathsf{NomLinList}$.

Proof. We start by focussing on naturality 1. Let $f : A \otimes U \to B \otimes U$, and $g : A' \to A$. Then we have the following equalities. Note that, in our case, F is a strict functor of monoidal categories,

[1] A equivalent definition is given through the feedback operator:

$$\text{Given } i \leq n,\ g(i) = \begin{cases} f(i) \text{ if } f(i) \leq n \\ \mathsf{feedback} f(i) \text{ otherwise} \end{cases}$$
$$\text{Given } i > n,\ \mathsf{feedback}(i) = \begin{cases} f(i) \text{ if } f(i) \leq n \\ \mathsf{feedback}(f(i)) \text{ otherwise} \end{cases}$$

that is $F(f \otimes g) = F(f) \otimes F(g)$.

$$
\begin{aligned}
g; \mathsf{Tr}^{U}_{A,B\ \mathsf{NomLinList}}(f) &= g; \chi_A; (G(\mathsf{Tr}^{F(U)}_{F(A),F(B),\mathsf{SSMC}}(F(f))); \chi_B^{-1} && (3.1)\\
&= \chi_{A'}; (G \circ F)(g); (G \circ \mathsf{Tr}^{F(U)}_{F(A),F(B),\mathsf{SSMC}}(F(f))); \chi_B^{-1} && (3.2)\\
&= \chi_{A'}; (G \circ (F(g); \mathsf{Tr}^{F(U)}_{F(A),F(B),\mathsf{SSMC}}(F(f))); \chi_B^{-1} && (3.3)\\
&= \chi_{A'}; (G \circ \mathsf{Tr}^{F(U)}_{F(A'),F(B),\mathsf{SSMC}}((F(g) \otimes \mathsf{id}_{F(U)}); F(f))); \chi_B^{-1} && (3.4)\\
&= \chi_{A'}; (G \circ \mathsf{Tr}^{F(U)}_{F(A'),F(B),\mathsf{SSMC}}((F(g \otimes \mathsf{id}_U); F(f))); \chi_B^{-1} && (3.5)\\
&= \chi_{A'}; (G \circ \mathsf{Tr}^{F(U)}_{F(A'),F(B),\mathsf{SSMC}}((F(g \otimes \mathsf{id}_U); f))); \chi_B^{-1} && (3.6)\\
&= \mathsf{Tr}^{U}_{A',B,\ \mathsf{NomLinList}}((g \otimes \mathsf{id}_U); f) && (3.7)
\end{aligned}
$$

where the passage from (3.1) to (3.2) follows the naturality of χ, from (3.2) to (3.3) the functoriality of G, from (3.3) to (3.4) we use the property (1) of the trace, from (3.4) to (3.5) the strict monoidality of F, from (3.5) to (3.6) the functoriality of F, and finally the definition of the trace in NomLinList for the last bit.

The two other naturality properties are proven in a similar manner. We now turn to vanishing. Given $f : A \otimes I \to B \otimes I$, then let us compute the trace:

$$
\begin{aligned}
\mathsf{Tr}^{I}_{A,B,\ \mathsf{NomLinList}}(f) &= \chi_A; G \circ \mathsf{Tr}^{F(I)}_{F(A),F(B),\ \mathsf{SSMC}}(F(f)); \chi_B^{-1} && (3.8)\\
&= \chi_A; G \circ \mathsf{Tr}^{I}_{F(A),F(B),\ \mathsf{SSMC}}(F(f)); \chi_B^{-1} && (3.9)\\
&= \chi_A; G \circ F(f); \chi_B^{-1} && (3.10)\\
&= f && (3.11)
\end{aligned}
$$

Similarly, we compute $\mathsf{Tr}^{U \otimes V}_{A,B,\ \mathsf{NomLinList}}(f)$. To do that, we first have to notice that for every morphism $g : A \to B \in \mathsf{SSMC}$, there is a morphism $f : A' \to B' \in \mathsf{NomLinList}$ such that $F(f) = g$. As a result, $F(\chi_{A'}; G(g); \chi_{B'}^{-1}) = F(\chi_{A'}; G \circ F(f); \chi_{B'}^{-1}) = F(f) = g$.

$$
\begin{aligned}
\mathsf{Tr}^{U \otimes V}_{A,B,\ \mathsf{NomLinList}}(f) &= \chi_A; G \circ \mathsf{Tr}^{F(U \otimes V)}_{F(A),F(B),\ \mathsf{SSMC}}(F(f)); \chi_B^{-1} && (3.12)\\
&= \chi_A; G \circ \mathsf{Tr}^{F(U) \otimes F(V)}_{F(A),F(B),\ \mathsf{SSMC}} F(f) \chi_B^{-1} && (3.13)\\
&= \chi_A; G \circ (\mathsf{Tr}^{F(U)}_{F(A),F(B),\ \mathsf{SSMC}}(\mathsf{Tr}^{F(V)}_{F(A) \otimes F(U),F(B) \otimes F(U),\ \mathsf{SSMC}}(F(f)))) \chi_B^{-1} && (3.14)\\
&= \chi_A; G \circ \mathsf{Tr}^{F(U)}_{F(A),F(B),\ \mathsf{SSMC}}(F(\chi_{A \otimes U}; G(\mathsf{Tr}^{F(V)}_{F(A \otimes U),F(B \otimes U)}(F(f)); \chi^{-1}_{B \otimes U})); \chi_B^{-1} && (3.15)\\
&= \mathsf{Tr}^{U}_{A,B,\mathsf{NomLinList}}(\mathsf{Tr}^{V}_{A,B,\mathsf{NomLinList}}(f)) && (3.16)
\end{aligned}
$$

The superposing property relies on the monoidality of the natural isomorphism χ.

$$
\begin{aligned}
g \otimes \mathsf{Tr}^U_{A,B,\ \mathsf{NomLinList}}(f) &= g \otimes (\chi_A ; G \circ \mathsf{Tr}^{F(U)}_{F(A),F(B),\ \mathsf{SSMC}}(F(f)); \chi_B^{-1}) && (3.17)\\
&= (\chi_W ; (G \circ F)(g); \chi_Z^{-1}) \otimes (\chi_A ; G \circ \mathsf{Tr}^{F(U)}_{F(A),F(B),\ \mathsf{SSMC}}(F(f)); \chi_B^{-1}) && (3.18)\\
&= \chi_{W\otimes A} ; G \circ (F(g)) \otimes (G \circ \mathsf{Tr}^{F(U)}_{F(A),F(B),\ \mathsf{SSMC}}(F(f))); \chi^{-1}_{Z\otimes B} && (3.19)\\
&= \chi_{W\otimes A} ; G \circ (F(g) \otimes \mathsf{Tr}^{F(U)}_{F(A),F(B),\ \mathsf{SSMC}}(F(f))); \chi^{-1}_{Z\otimes B} && (3.20)\\
&= \chi_{W\otimes A} ; G \circ (\mathsf{Tr}^{F(U)}_{F(W)\otimes F(A),F(Z)\otimes F(B),\ \mathsf{SSMC}}(F(g \otimes f))); \chi^{-1}_{Z\otimes B} && (3.21)\\
&= \chi_{W\otimes A} ; G \circ (\mathsf{Tr}^{F(U)}_{F(W\otimes A),F(Z\otimes B),\ \mathsf{SSMC}}(F(g \otimes f))); \chi^{-1}_{Z\otimes B} && (3.22)\\
&= \mathsf{Tr}^U_{W\otimes A,Z\otimes B,\ \mathsf{NomLinList}}(g \otimes f) && (3.23)
\end{aligned}
$$

Finally, the last property, yanking, is straightforward. □

As a final note, it turns out that this category is not the free traced monoidal category. Indeed, the free traced monoidal category "counts the loops" that have been erased, as proven in [1].

3.3 Polarities and compact closed category

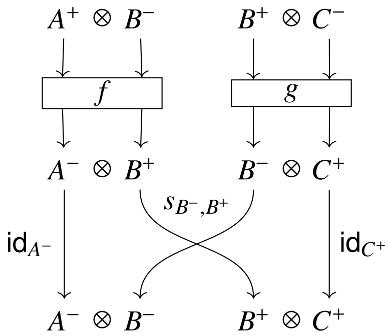
Having established that the category NomLinList can be equipped with a trace, we can build the free compact closed category of it following a famous construction by Joyal, Street, and Verity [54]. We recall that the definition of a compact closed category might be found in 2.5. We present below the original Int construction, that will be later slightly refined for our purposes.

In the sequel, we will slightly generalise the definition of the trace. Given given $f : A_1 \otimes U \otimes A_2 \to B_1 \otimes U \otimes B_2$, we define $\mathsf{Tr}^U_{A_1,A_2,B_1,B_2}(f) : A_1 \otimes A_2 \to B_1 \otimes B_2$ to be $\mathsf{Tr}^U_{A_1\otimes A_2, B_1\otimes B_2}((\mathsf{id}_{A_1} \otimes s_{A_2,U}); f; (\mathsf{id}_{B_1} \otimes s_{U,B_2}))$.

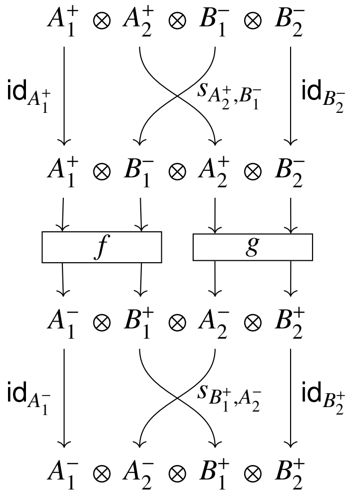
Definition 3.14. *Given a traced symmetric monoidal category C, its free compact closure* Int(C) *can be described by the following :*

- *Objects of* Int(C) *are pairs of objects $A = (A^-, A^+)$ of C.*
- *Morphisms $f : A \to B$ are morphisms $g : A^+ \otimes B^- \to A^- \otimes B^+$ of C*
- *Composition is defined through parallel composition + tracing . Given $f : A \to B$ and $g : B \to C$, one defines:*

$$f; g = \mathsf{Tr}^{B^-\otimes B^+}_{A^+,C^-,A^-,C^+}((f \otimes g); (\mathsf{id}^+_A \otimes s_{B^-,B^+} \otimes \mathsf{id}_{C^+}))$$



- *The tensor product is given by the following formula on objects:* $(A^- \otimes A^+) \otimes (B^-, B^+) = (A^- \otimes B^-, A^+ \otimes B^+)$*; and given two morphisms* $f_1 : (A_1^-, A_1^+) \to (B_1^-, B_1^+)$*,* $f_2 : (A_2^-, A_2^+) \to (B_2^-, B_2^+)$*, we define their tensor product according the below diagram:*



Furthermore, the unit of the tensor product is (I, I)*.*

- *The negation is defined by* $(A^-, A^+)^\perp = (A^+, A^-)$ *on objects. Given a morphism* $f : (A^-, A^+) \to (B^-, B^+)$*, seen as a morphism* $f : A^+ \otimes B^- \to A^- \otimes B^+$*, the morphism* $f^\perp : (B^+, B^-) \to (A^+, A^-)$*, seen as* $f^\perp : B^- \otimes A^+ \to B^+ \otimes A^-$ *is defined by* $f^\perp = s_{B^-,A^+}; f; s_{B^+,A^-}$*. This defines a contravariant functor.*

Each formula is modelled by spitting it into its negative and positive fragment, whereas occurrences of positive or negative atomic formulas can appear throughout the formula. For instance, to interpret the formula $Y \otimes Y^\perp \otimes Z \otimes W^\perp$ in this category, we would have to reorder and divide the formula in two distinct fragments of opposite polarity $((Y^\perp \otimes W^\perp, Y \otimes Z))$. Therefore, we make use of the key property that in $\mathsf{Int}(C)$, negation commutes with tensor. That is:

$$(A \otimes B)^\perp = A^\perp \otimes B^\perp.$$

We call **irreducible objects**, objects A of $\mathsf{NomLinList}$ that can not be further decomposed into $A = A_1 \otimes A_2$. As a consequence of the above remark, any object of $\mathsf{Int}(\mathsf{NomLinList})$ will be isomorphic to one of the form :

$$\bigotimes A_i \otimes \bigotimes B_i^\perp$$

where each A_i, B_i is an irreducible object of $\mathsf{NomLinList}$, seen as an object of $\mathsf{Int}(\mathsf{NomLinList})$. Therefore, it is enough to assign polarity to irreducible objects only.

As a result, we can give a second description of the free compact closed category on ($\mathsf{NomLinList}, \mathsf{Tr}$) by only assigning polarity to irreducible objects. We will call this category $\mathsf{NomLinPol}$. We do that by the means of polarised atomes, that are pairs (a, p) where p is either $+1$ or -1, and where a is an atome. Indeed, 1-orbit sets of lists of length 1 correspond precisely to irreducible objects. The objects of $\mathsf{NomLinPol}$ are nominal sets of polarised and separated lists subject to some more properties stated below. Elements of the sets are built out of the following grammar:

$$L \;::=\; (a, p) \;\mid\; (\bullet, p) \;\mid\; L_1.L_2$$

where $a \in \mathbb{A},\; p \in \{-1, 1\},\;$ and $L_1 \#_{\mathsf{pol}} L_2$ where $\#_{\mathsf{pol}}$ is defined below. Such an element is called a **polarised separated list**. The p next to each atome is called the **polarity** of the atome. Given a list L, we define by $\mathsf{Pos}(L)$ the restriction of L to its atomes of positive polarity, and $\mathsf{Neg}(L)$ its restriction to the negative ones. Formally, these two functions are defined as follows, where ε denotes the empty list, and a ranges over atomes:

$$\begin{array}{llll} \mathsf{Pos}(\varepsilon) = \varepsilon & \mathsf{Pos}(a, 1) = a & \mathsf{Pos}(a, -1) = \varepsilon & \mathsf{Pos}(L_1.L_2) = \mathsf{Pos}(L_1).\mathsf{Pos}(L_2) \\ \mathsf{Neg}(\varepsilon) = \varepsilon & \mathsf{Neg}(a, 1) = \varepsilon & \mathsf{Neg}(a, -1) = a & \mathsf{Neg}(L_1.L_2) = \mathsf{Neg}(L_1).\mathsf{Neg}(L_2) \end{array}$$

We write $L_1 \#_{\mathsf{pol}} L_2$ if $(\mathsf{Pos}(L_1) \# \mathsf{Pos}(L_2)) \wedge (\mathsf{Neg}(L_1) \# \mathsf{Neg}(L_2))$, and note $\star_{\mathsf{pol}}$ the associated polarised separated concatenation operation :

$$A \star_{\mathsf{pol}} B = \{L_1.L_2 \mid L_1 \in A, L_2 \in B, L_1 \#_{\mathsf{pol}} L_2\}.$$

We write $L_1 \star_{\mathsf{pol}} L_2$ as a shorthand for $L_1.L_2$ knowing $L_1 \#_{\mathsf{pol}} L_2$. We furthermore define the operation $(.)^\perp$ on lists, that consists in inverting all polarities.

$$(a, p)^\perp = (a, -p) \qquad (\bullet, p)^\perp = (\bullet, -p) \qquad (L_1.L_2)^\perp = L_1^\perp.L_2^\perp$$

We extend this operation to sets : $A^\perp = \{L^\perp \mid L \in A\}$.

Furthermore, to avoid sets of the form $\{(a, -1).(a, 1) \mid a \in \mathbb{A}_X\}$, that do not correspond to a denotation of a formula of MLL, it is necessary to add a condition that stipulates that $X \simeq \mathsf{Neg}(X) \times \mathsf{Pos}(X)$. This basically enforces that there is no relationship between the negative and positive names occurring within the list. Note that if X is non-empty, then $\mathsf{Pos}(X)$, $\mathsf{Neg}(X)$ are never empty.

Definition 3.15. *The objects of* NomLinPol *are non-empty nominal sets A of non-empty polarised separated lists such that $A \simeq \mathsf{Neg}(A) \times \mathsf{Pos}(A)$, and both $\mathsf{Neg}(A), \mathsf{Pos}(A)$ are 1-orbit sets of lists. The morphisms $A \to B$ are nominal linear functions f:*

$$f : \mathsf{Pos}(A) \star \mathsf{Neg}(B) \to \mathsf{Neg}(A) \star \mathsf{Pos}(B).$$

The composition of morphisms is defined by using the trace structure just as presented in the Int *construction.* NomLinPol *is a compact closed category :*

- *The monoidal product $A \otimes B$ is modelled by $\star_{pol}$ on objects, and by cartesian product on morphisms. The unit I is $I = \{(\bullet, 1)\}$.*
- *The negation is modelled by $(.)^{\perp}$ on objects, and as defined by the* Int*-construction on morphisms.*

Note that the monoidal product is well-defined since if $A \simeq \mathsf{Neg}(A) \times \mathsf{Pos}(A)$, and $B \simeq \mathsf{Pos}(B) \times \mathsf{Neg}(B)$, then $A \star_{\mathsf{pol}} B \simeq \mathsf{Neg}(A \star_{\mathsf{pol}} B) \times \mathsf{Pos}(A \star B)$, since $\mathsf{Neg}(A \star_{\mathsf{pol}} B) \simeq \mathsf{Neg}(A) \star \mathsf{Neg}(B)$.

Let us note that our choice of a monoidal unit is far from innocent. Indeed, decomposing I into its negative and positive part, we have $I = \{\bullet\} \times \{\varepsilon\}$. This corresponds to the idea that we want I to be somehow positive. We could also have chosen $I = \{(\bullet, -1)\}$, $I = \{(\bullet, -1).(\bullet, 1)\}$, or, finally, $I = \{\varepsilon\}$ if we had allowed sets of empty lists in the objects of our category. However, these choices would have not been appropriate for a future modelling of a star-autonomous category, where we expect I to be the positive unit, and $\perp = I^{\perp}$ the negative one. In categorical terms, we have chosen a unit $I = \{(e, 1).(d, -1))\}$ where e is a non strict unit: $e \otimes A \simeq A$ but $e \otimes A \neq A$, and d is a strict one: $d \otimes A = A$.

We can define a denotation by assigning to the atomic type X the set $[\![X]\!] = \{(a, 1) \mid a \in \mathbb{A}_X\}$ giving polarity 1 to the atomes of X, $[\![1]\!] = I = \{(\bullet, 1)\}$. Furthermore, we obviously set $[\![A \otimes B]\!] = [\![A]\!] \otimes [\![B]\!]$, and $[\![A^{\perp}]\!] = [\![A]\!]^{\perp}$. As the objects of the category are those freely generated by negation and tensor from VAR, NomLinPol is the free compact closed category on the traced monoidal category (NomLinList, Tr).

A sequent calculus has been presented in [86] for the collapsed version of multiplicative linear logic, where the two tensors $\otimes$ and $⅋$ are merged, giving rise to a logic, named multiplicative compact closed linear logic , whose sound models are precisely the compact closed categories. NomLinPol is a model of multiplicative compact closed linear logic, whose sequent calculus is presented in the figure 3.3 below.

Due to the symmetry between the right rules and the left rules, a simplified version can be given, where the left exchange is obtained as a combination: left negation - right exchange - right negation, and similarly for the left $\otimes$.

A proof $\pi : A_1, .., A_n \vdash B_1, ..., B_n$ is modelled by a morphism $\phi : [\![A_1 \otimes .... \otimes A_n]\!] \to [\![B_1 \otimes ... \otimes B_n]\!]$, such that if there is an axiom link between two occurrences $X^{\perp}, X$ in π, then

$$\frac{}{A \vdash A}\text{Ax} \qquad \frac{\Gamma_1 \vdash \Delta_1, A \qquad A, \Gamma_2 \vdash \Delta_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\text{Cut}$$

$$\frac{\Gamma_1, A, B, \Gamma_2 \vdash \Delta}{\Gamma_1, B, A, \Gamma_2 \vdash \Delta}\text{Left exchange} \qquad \frac{\Gamma \vdash \Delta_1, A, B, \Delta_2}{\Gamma \vdash \Delta_1, B, A, \Delta_2}\text{Right exchange}$$

$$\frac{\Gamma, A \vdash \Delta}{\Gamma \vdash A^{\perp}, \Delta}\text{Left negation} \qquad \frac{\Gamma \vdash A, \Delta}{\Gamma, A^{\perp} \vdash \Delta}\text{Right negation}$$

$$\frac{\Gamma \vdash \Delta}{\Gamma, 1 \vdash \Delta}\text{Left 1} \qquad \frac{}{\vdash 1}\text{Right 1}$$

$$\frac{\Gamma_1 \vdash \Delta_1, A \qquad \Gamma_2 \vdash \Delta_2, B}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2, A \otimes B}\text{Right} \otimes \text{-1} \qquad \frac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \otimes B, \Delta}\text{Right} \otimes \text{-2}$$

$$\frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \otimes B \vdash \Delta}\text{Left} \otimes$$

Figure 3.3: Sequent calculus for multiplicative compact closed linear logic

ϕ will send the name in the location associated to the occurrence $X^{\perp}$ to the location of the occurrence of X.

3.3.1 On the free compact closed category

The category NomLinPol is the free compact closed category on the traced monoidal category (NomLinList, Tr), but not the free compact closed category on VAR, as, as explained above, (NomLinList, Tr) is not the free traced monoidal category on NomLinList. Indeed, the free traced category FTC has scalars (morphisms $f : I \to I$) that reflect on loops erased when composing / tracing the morphisms, whereas NomLinList has a collapsed homset NomLinList$(I \to I)$; the only morphism in it is the identity.

More precisely the objects of the free traced monoidal category are obviously the same as NomLinList, but now the morphisms are pairs (f, S), where S is a set of natural numbers indexed by TVar. Each of them reflect the loops of type $X, (X \in \mathsf{TVar})$ that have been erased when tracing. For instance, the following equality holds. Given $f : A \otimes B \otimes B \otimes A \to A \otimes B \otimes B \otimes A$, given by $f = (\mathsf{id}_A \star s_{B,B} \star \mathsf{id}_A)$ then $\mathsf{Tr}_{A,A}^{B \otimes B \otimes A}(f) = (\mathsf{id}_A, 1_B, 1_A) : A \to A$. Indeed, one loop has been erased between B-objects, and one between A-objects.

This is related to the fact that the compact closed linear logic does not enjoy a full cut-elimination procedure. A proof is called normal when the only cuts are loops, that are, proofs of the following form:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A}}{\vdash A^{\perp}, A}}{\vdash A^{\perp} \otimes A} \qquad \dfrac{\dfrac{\dfrac{}{A \vdash A}}{A^{\perp}, A \vdash}}{A^{\perp} \otimes A \vdash}}{\vdash}\text{Cut}$$

A loop cannot be eliminated by a cut elimination algorithm, as there are no cut-free proofs of $\vdash$.

Therefore, the normal form of a proof in normal-form can only be faithfully modelled through a morphism of NomLinPol that represents the cut-free part, and a sequence of numbers labelled with TVar representing the loops. For instance, the proof $I \to I$ above should be modelled by $(\mathsf{id}_I, 1_A)$. Therefore, NomLinPol is fully complete for the compact closed logic, but not faithfully so. Several proofs with different invariants might be sent to the same morphism.

3.4 Nominal relations

3.4.1 Nominal linear relations

One major drawback of the previous construction is that it does not have products and coproducts. We can tackle this problem by moving from functions to relations. The category of relations is one of the simplest known models of linear logic. We carefully adapt it to take care of polarities and names.

Definition 3.16. *A **nominal linear relation** $\mathcal{R} : A \to B$ is a nominal subset of $A \times B$ such that if $(a, b) \in \mathcal{R}$ then $\nu(a) = \nu(b)$.*

In the future, we will write $a\mathcal{R}b$ for $(a, b) \in \mathcal{R}$. Linear nominal relations compose as relations and the identity relation is linear nominal. Consequently, they organise themselves as a category. We shall specify a full subcategory of it, whose objects correspond to denotations of formulas of MALL. We denote this sub-category NomLinRel. Its objects are nominal sets of non-empty separated, **annotated** lists of atomes, where the adjective annotated highlights the fact that lists will be built using patterns. More precisely, the elements L of the sets are built from the following grammar :

$$L \ := \ a \ | \ \bullet \ | \ \mathsf{inl}(L) \ | \ \mathsf{inr}(L) \ | \ L_1.L_2$$

where $a \in A$, and $L_1 \# L_2$.

Note that the previously enforced condition that the sets are 1-orbit, or non-empty is dropped. The category NomLinRel can be equipped with a monoidal product, written $\star$. The monoidal product of relations is usually defined via the cartesian product of sets, but to accommodate names and lists we rather use the separated concatenation $\star$, just as in NomLinList.

- $A \otimes B = A \star B = \{L_1.L_2 \mid L_1 \in A, L_2 \in B, L_1 \# L_2\}$
- $\mathcal{R}_1 \otimes \mathcal{R}_2 = \mathcal{R}_1 \star \mathcal{R}_2 = \{(a.b, c.d) \ | \ (a, c) \in \mathcal{R}_1, (b, d) \in \mathcal{R}_2, a\#b \wedge c\#d\}$

The unit of the monoidal product is the set $I = \{\bullet\}$. Once again, the use of the separated product as a monoidal product prevents the category from having non-linear morphisms, similarly as NomLinList. In the sequel, given $L = L_1.L_2$ in $A \otimes B$, we will write $\upharpoonright A$ for the projection on A, and $\upharpoonright B$ for the projection on B, that is $L \upharpoonright A = L_1$ and $L \upharpoonright = L_2$.

The inl, inr constructors are called **patterns**, and will allow us to form coproducts, denoted by $\oplus$.

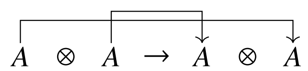
- $A \oplus B = \mathsf{inl}(A) \uplus \mathsf{inr}(B) = \{\mathsf{inl}(L) \mid L \in A\} \uplus \{\mathsf{inr}(L) \mid L \in B\}$.
- $\mathcal{R}_1 \oplus \mathcal{R}_2 = \mathsf{inl}(\mathcal{R}_1) \uplus \mathsf{inr}(\mathcal{R}_2)$

where, given $\mathcal{R} : A \to B$, we define $\mathsf{inl}(\mathcal{R}) = \{\mathsf{inl}(L_1).L_2 \mid L_1.L_2 \in \mathcal{R}, L_1 \in A, L_2 \in B\}$, and respectively for inr. Furthermore, we write $\uplus$ to express the union of disjoint sets.

The unit of the coproduct 0 is the empty set $\emptyset$. $\oplus$ is not a simple tensor product, it actually defines a biproduct: $A \oplus B$ is the biproduct of A and B. We remind here that a biproduct is an object that can act both as a coproduct of A,B, as a product of A, B, and does both in a compatible way that we will not describe here.

The category (NomLinRel, $\otimes, \oplus$) is a monoidal category with coproduct. Furthermore, the tensor product distributes over the coproduct: $A \otimes (B \oplus C) \simeq (A \otimes B) \oplus (A \otimes C)$, in the sense that there is an isomorphism between them. As a result, (NomLinRel, $\otimes, \oplus$) defines a model of the $(\otimes, \oplus)$ fragment of MALL, where $[\![1]\!] = I$, $[\![X]\!] = \mathbb{A}_X, [\![0]\!] = 0_{\mathsf{NomLinRel}}$. One can easily see that NomLinList is a subcategory of NomLinRel, and we write $\mathcal{I}$ for the inclusion functor NomLinList $\xrightarrow{\mathcal{I}}$ NomLinRel. We say that an object of the category is **well-typed** if it can be built out of the $[\![X]\!], I, 0$ using the two previously described operations. This time, as we dropped the restriction that the objects are 1-orbit, some objects might not be well-typed. A typical example of such an object is $\mathbb{A}$. The subcategory of well-typed objects of NomLinRel corresponds to the biproduct completion of NomLinList. In the sequel, we will restrict NomLinRel to its full subcategory of well-typed objects.

Each element of a morphism of NomLinRel defines a set of axiom links, that is, a set-function from the atomic literals on the domain to the codomain. For instance, an element $(a_1.a_2, a_2.a_1)$ of $X \otimes X \to X \otimes X$ defines the following set of axiom-links.

$\sigma :$ 

In contrast to Rel, the category of sets and relations, NomLinRel is not monoidal closed, notably due to the linear condition on the relations. Indeed, given a relation $A \to B$, if the category was closed there would be a corresponding "name" $\mathcal{R}' : I \to A \multimap B$, and, as a result, for all x in $A \multimap B$, if $\bullet \mathcal{R} x$ then $\nu(x) = \nu(\bullet) = \emptyset$. Hence the only elements of $A \multimap B$ would be lists of atomes of empty support. To solve this, we want to apply, as we did in the previous section, the Int construction. Therefore, we first have to address its traced structure.

Just as its sibling NomLinList, tracing NomLinRel is not straightforward. For instance, taking $\mathcal{R}$ as above, defined by $(a_1.a_2)\mathcal{R}(a_2.a_1)$. Then tracing $\mathsf{tr}^{A_2}_{A_1,A_1}(\mathcal{R})$ (where A_2 are the right hand side A's), requires us to equate a_2 and a_1, but the element $(a_1.a_1, a_1.a_1)$ is not part of the relation.

3.4.1.1 Category of lax nominal relations

In order to bypass the above restriction, we close linear relations under strict substitutions before composing them. We define **strict substitutions** to be name substitutions of the form $[a_n/b_n]...[a_1/b_1]$ where for each i, there is a type variable X_i such that $a_i, b_i \in \mathbb{A}_{X_i}$. Similarly as for nominal permutations, we write "$\cdot$" for the action of strict substitutions on elements. We call Ξ the set of strict substitutions, and often use the lowercase e to refer to one of its elements. The composition of strict substitutions e_1, e_2 consists in concatenating their sequences, and is therefore written $e_1.e_2$. We devise several properties of strict substitutions and define them formally in the appendix 9.4. We then have $e_1.e_2 \cdot x = e_1 \cdot (e_2 \cdot x)$. We also introduce the following notation. Given a well-typed set A of annotated name-distinct lists, we write $\hat{A}$ for the corresponding set of annotated lists, such that the name-distinct property is dropped. Formally, $\hat{A}$ is defined as follows by induction on the structure of A.

$$\begin{array}{ll} \widehat{\mathbb{A}_X} = \mathbb{A}_X & \widehat{\{\bullet\}} = \{\bullet\} \\ \widehat{A_1 \star A_2} = \hat{A}_1 \times \hat{A}_2 & \widehat{A \uplus B} = \hat{A} \uplus \hat{B} \end{array}$$

That way, if an element $x \in A$, then the set $\hat{x} = \{e \cdot x \mid e \in \Xi\} \subseteq \hat{A}$. For $\mathcal{R} : A$ a linear relation, we write $\hat{\mathcal{R}} : \hat{A}$ for the closure under strict substitutions of $\mathcal{R}$, that is:

$$\widehat{\mathcal{R}} = \{e \cdot r \mid r \in \mathcal{R}, e \in \Xi\}.$$

Proposition 3.17. $\hat{(.)}$ *defines a symmetric monoidal functor from the category* NomLinRel *to the category of nominal linear, non-separated, relations. We call this category* LaxNomLinRel*. It has nominal sets of annotated lists as objects, and linear relations as morphisms.*

To prove the property, we will use this following lemma of strict substitutions, that is proven appendix 9.4.

Lemma 3.18. *Let x an element of nominal set, and $e \in \Xi$. Then $e \cdot x = e' \cdot (\pi \cdot x)$, where if $e' = [a_n/b_n]...[a_1/b_1]$, and, writing e'_i for $[a_i/b_i]...[a_1/b_1]$, then $a_{i+1}, b_{i+1} \in \nu(e'_i \cdot (\pi \cdot x))$ and where π is a nominal permutation.*

Given an element x and $e \in \Xi$, we refer to the (e', π) in the above lemma as a canonical form of e given x. Equivalently, we say that e is in canonical form for x, if (e, id) is a canonical form of e given x. The lemma basically says that the action of a strict substitution on an element consists of two distinct actions: a name-permutation followed by a name-merging.

Proof of Proposition 3.17. Given $\mathcal{R} : A \to B$ and $\mathcal{Q} : B \to C$, we need to prove that:

$$\widehat{\mathcal{R}; \mathcal{Q}} = \widehat{\mathcal{R}}; \widehat{\mathcal{Q}}.$$

First, we consider the left to right inclusion. Let $u \in \widehat{\mathcal{R};Q}$. Then $\exists e \in \Xi, \exists u' \in \mathcal{R};Q.u = e\cdot u'$. Furthermore $\exists u'' \in A\times B\times C$ such that $u'' \upharpoonright A\times B \in \mathcal{R}$, $u'' \upharpoonright B\times C \in Q$, and $u'' \upharpoonright \hat{A}\times\hat{C} = u'$. Hence $e\cdot u'' \upharpoonright \hat{A}\times\hat{C} = u$ and $e\cdot u'' \upharpoonright \hat{A}\times\hat{B} \in \widehat{\mathcal{R}}$, $e\cdot u'' \upharpoonright \hat{B}\times\hat{C} \in \widehat{Q}$, hence $u \in \hat{\mathcal{R}};\hat{Q}$, that is, it belongs in the second set. Now let us prove the right to left inclusion. Formally, let $u' \in \widehat{\mathcal{R}};\widehat{Q}$. It entails that there exist $u \in \hat{A}\times\hat{B}\times\hat{C}$ such that $u \upharpoonright \hat{A}\times\hat{B} \in \widehat{\mathcal{R}}$, $u \upharpoonright \hat{B}\times\hat{C} \in \widehat{Q}$, and $u \upharpoonright A\times C = u'$. Then let us, as above, use canonical forms for the strict substitutions and use the fact that both $\mathcal{R}$ and Q are closed under nominal permutations. Hence $u \upharpoonright \hat{A}\times\hat{B} = e_1\cdot u_1$, $u_1 \in \mathcal{R}$ and e_1 is in canonical form for u_1. Furthermore $u \upharpoonright \hat{B}\times\hat{C} = e_2\cdot u_2$, $u_2 \in Q$ and e_2 is in canonical form for u_2. Let us call $x = u_1 \upharpoonright B$ and $y = u_2 \upharpoonright B$. As there exists strict substitutions that equate them, it entails that the lists have same length and same type. Hence, thanks to the fact that the lists are separated, they are nominally equivalent, that is, there exists a permutation π such that $\pi\cdot u_1 \upharpoonright B = u_2 \upharpoonright B$. Now, by using the fact that every permutation can be encoded as a strict substitution for a given specific element (property refprop:permsubstitution), we write $\overline{\pi^{-1}}$ for the strict substitution that encodes π^{-1} for $\pi\cdot u_1$. Hence $e_2{\cdot}u_2 \upharpoonright B = e_1{\cdot}(u_1 \upharpoonright B) = e_1.\overline{\pi^{-1}}{\cdot}\pi{\cdot}(u_1 \upharpoonright B) = e_1.\overline{\pi^{-1}}{\cdot}(u_2 \upharpoonright B)$. As $\nu(u_2 \upharpoonright B) = \nu(u_2 \upharpoonright C)$, $e_2\cdot u_2 \upharpoonright C = e_1.\overline{\pi}^{-1}u_2 \upharpoonright C$. Let $u'' = e_1.\overline{\pi^{-1}}\cdot(\pi\cdot u_1 \mid_B u_2)$, where $|_B$ indicates that we merge the two B parts of the lists, that are equal. Therefore, as $\pi\cdot u_1 \in \mathcal{R}$, we can conclude that $u'' \upharpoonright A\times C$ belongs in $\widehat{\mathcal{R};Q}$, and moreover $u'' = u$.

Also, one can prove that $\mathsf{id}_{\hat{A},\mathsf{LaxNomLinRel}} = \widehat{\mathsf{id}_{A,\mathsf{NomLinRel}}}$. The inclusion $\widehat{\mathsf{id}_{A,\mathsf{NomLinRel}}} \subseteq \mathsf{id}_{\hat{A},\mathsf{LaxNomLinRel}}$ is automatic, whereas the other consists in proving that any element of $\mathsf{id}_{\hat{A},\mathsf{LaxNomLinRel}}$ can be obtained from an element of $\mathsf{id}_{A,\mathsf{NomLinRel}}$ by means of substitutions.

The monoidality comes from $\widehat{A\star B} = A\times B$ by definition, and similarly on morphisms: $\widehat{\mathcal{R}\star Q} = \mathcal{R}\times Q$. □

LaxNomLinRel is not an appropriate category to model morphisms of linear logic, as there might be morphisms $A\otimes A \to A$ or $A \to A\otimes A$. On the other hand, to relax the separated condition will prove useful in the future to deal with tracing, and therefore, with composition when dealing with the compactification of this category through the Int-construction.

3.4.2 Trace structure on NomLinRel

Given a relation $\mathcal{R} : A\otimes U \to B\otimes U \in \mathsf{NomLinRel}$, let us define $\mathsf{Tr}^U_{A,B}(R)$ by:

$$\mathsf{Tr}^U_{A,B}(\mathcal{R}) = \{r \upharpoonright \hat{A}\times\hat{B} \mid r \in \hat{\mathcal{R}} \wedge r \upharpoonright \hat{A}\times\hat{B} \in A\times B \wedge r \upharpoonright \hat{U}\times\hat{U} \in \widehat{\mathsf{id}_U}\}. \tag{3.24}$$

The use of strict substitutions allows us to equate some names in the relation. However, the part in $\hat{B}\times\hat{C}$ must still have the property that the atomes are name-distinct on each projection, condition that is enforced through $r \upharpoonright \hat{B}\times\hat{C} \in B\times C$. One can see that NomLinList is a subcategory of NomLinRel. We write $\mathcal{I}$ for the inclusion functor, each morphism of NomLinList is sent by $\mathcal{I}$ to its graph, seen as a relation. We would like to prove that the trace defined in NomLinRel extends the one of NomLinList.

Proposition 3.19. *Let* $\phi : A \otimes U \to B \otimes U \in \mathsf{NomLinList}$*, then:*

$$\mathcal{I}(\mathsf{Tr}^U_{A,B,\ \mathsf{NomLinList}}(\phi)) = \mathsf{Tr}^U_{A,B,\ \mathsf{NomLinRel}}(\mathcal{I}(\phi)) \quad .$$

Proof. We first show that the inclusion $\mathsf{Tr}^U_{A,B,\ \mathsf{NomLinRel}}(\mathcal{I}(\phi)) \subseteq \mathcal{I}(\mathsf{Tr}^U_{A,B,\ \mathsf{NomLinList}}(\phi))$. Let $x \in \mathcal{I}(\phi)$, and consider any y such that $\exists e \in \Xi.\ y = e \cdot x,\ \ y \restriction \hat{U} \times \hat{U} \in \widehat{\mathsf{id}_U}$, and $y \restriction \hat{A} \times \hat{B} \in A \times B$. Then, without loss of generality, we can look at a canonical form of e given x, use the fact that $\mathcal{I}(\phi)$ is closed under permutations, and consider that the only action of e is to equate names present in x.

Now let us consider, that, as defined in Section 3.1.2.1 above, we attribute a number to each location, and we will now compute the function ψ attributed to $y \restriction A \times B$. So let us consider a location l in A, and the associated name y_l in y, corresponding to x_l in x. As $e \cdot x \restriction \hat{A} = y \restriction \hat{A}$, $e \cdot x \restriction \hat{A} \in A$, and A is separated, then this entails there is a $\pi \in \mathsf{Perm}(\mathbb{A})$ such that $(\pi \cdot y) \restriction A = x \restriction A$. Furthermore, applying a permutation to y does not change the function ψ it defines. Hence one can assume without loss of generality that for all locations l in A, $x_l = y_l$. We decompose x, y into x_1, x_2, y_1, y_2, where $x_1 = x \restriction A \star U$ (respectively $y_1 = y \restriction \hat{A} \times \hat{U}$), $x_2 = x \restriction A \star U$ (resp $y_2 = y \restriction \hat{B} \times \hat{U}$).

Let l in A, and consider that $\phi(l) = l' \in B$. Then, as $x_{1,l} = x_{2,l'}$ and as y is obtained from x by strict substitutions, then $y_{2,l'} = y_{1,l}$. Therefore, if $\phi(l) \in B$ then $\psi(l) = \phi(l)$. Writing n for the number of locations of A (and hence, B as well), this sums up to:

$$\text{Given } i \leq n,\ \psi(i) = \phi(i) \text{ if } \phi(i) \leq n.$$

Now, let us suppose that $\phi(l) \in U$. Then $y_{1,l} = y_{2,\phi(l)}$. But, as $y \restriction \hat{U} \times \hat{U} \in \widehat{\mathsf{id}_U}$, we also have that $y_{2,\phi(l)} = y_{1,\phi(l)}$. As $y_{2,\phi(\phi(l)} = y_{1,\phi(l)}$, we obtain $y_{1,l} = y_{2,\phi\circ\phi(l)}$. Then, either $y_{2,\phi\circ\phi(l)}$ is in B or we repeat the routine. That is, if $i > n$, then $\psi(i) = \mathsf{feedback}(\phi)(i)$, where:

$$\mathsf{feedback}(\phi)(i) = \begin{cases} \phi(i) \text{ if } \phi(i) \leq n \\ \mathsf{feedback}(\phi)(\phi(i)) \text{ otherwise.} \end{cases}$$

Therefore, $\psi(i) = \mathsf{Tr}(\phi)(i)$, and $\mathsf{Tr}^U_{A,B,\mathsf{NomLinPol}}(\mathcal{I}(\phi)) \subseteq \mathcal{I}(\mathsf{Tr}^U_{A,B,\mathsf{NomLinList}}(\phi))$.

We now tackle the reverse inclusion. Let $y \in \mathcal{I}(\mathsf{Tr}^U_{A,B,\ \mathsf{NomLinList}}(\phi))$, let us show that this y belongs in $\mathsf{Tr}^U_{A,B,\ \mathsf{NomLinPol}}(\mathcal{I}(\phi))$. So we start by picking an $x \in \mathcal{I}(\phi)$, such that $y \restriction A = x \restriction A$. We build a strict substitution in n-steps, corresponding to the n locations of A.

Given $i \leq n$, let $x_{1,i}$ the i^{th} element of x_1. Suppose that $\phi(i) = j \leq n$, then $e_i = \varepsilon$ (the empty substitution). On the other hand, suppose that $\phi(i) = j > n$. Then we set $e_i = [x_{1,j}/x_{1,i}]$. That way, after applying e_i to x, resulting in $\tilde{x}$, we get $\tilde{x}_{1,j} = \tilde{x}_{1,i} = \tilde{x}_{2,j}$, and therefore $x \restriction l \in \mathsf{id}_l$

where l corresponds to the occurrence of the literal in the j^{th} position of $A \otimes U$. Inductively, if $\phi(j) = k > n$, we set $e_i ::= [x_{1,k}/x_{1,j}].e_i$. We repeat this process, until $\phi(l) = m \leq n$.

At this stage, one needs an important property of traces of permutations: each location l of U is used at most once when tracing (see [1] proposition 1). That is, if $\exists i \leq n$, such that $\exists k \geq 1$ $\phi^k(i) = l$ and $\phi^1(i), \phi^2(i), ..., \phi^{k-1}(i) > n$, then this i is unique. As a result, if a name x_l such that l in U is in one of the support of an e_i, then this i is unique.

Now, let us consider all names whose locations are in U and that are not in the support of any $e_i, 1 \leq i \leq n$. Then we consider the substitution e_{n+1} whose only action is to send all those names to a fresh name d. Finally, we consider $e = e_1.e_2. \ ... \ .e_n.e_{n+1}$. Then $e \cdot x \restriction A \to B = y$. This can be checked as in the above part of the proof. Furthermore, $e \cdot x \restriction U \to U \in \widehat{\mathsf{id}_U}$. We conclude that $y \in \mathsf{Tr}^U_{A,B,\mathsf{NomLinRel}}(\mathcal{I}(\phi))$. □

This property allows us to prove that tracing a linear relation indeed results in a linear relation.

Proposition 3.20. *Let* $\mathcal{R} : A \otimes U \to B \otimes U$*. Let* $x \in \mathsf{Tr}^U_{A,B}(\mathcal{R})$*, then* $\nu(x \restriction A) = \nu(x \restriction B)$*.*

Proof. The proof is based on the above proposition 3.19. Let $e \cdot x$ an element that appears as witness of the trace, and hence such that $x \in \mathcal{R}$. Then let us consider ϕ the morphism of NomLinList associated to x. Then $e \cdot x$ corresponds, as explained above, to tracing ϕ. Therefore, $e \cdot x \restriction A \to B$ belongs in a graph of a function $\psi \in \mathsf{NomLinList}(A \to B)$ and therefore has same support in A and B. □

We can now state with confidence that this definition of trace seems appropriate, as it is a simple extension of the one previously defined in the case without biproduct, and acts in a compatible way with linear nominal relations. Only remains to prove that this family of functions formally defines a trace.

Proposition 3.21. *The family of functions* $\mathsf{Tr}^U_{A,B}$ *defined in 3.24 forms a trace of the symmetric monoidal category* NomLinRel*.*

Proof. We start the proof with the naturality of the trace. We treat only one of the three cases, as the two others could be dealt with along the same lines. Let $\mathcal{R} : A \otimes U \to B \otimes U$ and $Q : A' \to A$. Then one must check that :

$$\mathsf{Tr}^U_{A,B}(\mathcal{R}) \circ Q = \mathsf{Tr}^U_{A',B}(\mathcal{R} \circ (Q \otimes \mathsf{id}_U)).$$

The left-hand-side term unfolds to:

$$\{u \upharpoonright A' \times B \mid u \in A' \times \hat{A} \times \hat{U} \times \hat{B} \times \hat{U}), u \upharpoonright A' \times \hat{A} \in Q, u \upharpoonright (\hat{A} \times \hat{U}) \times (\hat{B} \times \hat{U}) \in \widehat{\mathcal{R}}, u \upharpoonright \hat{U} \times \hat{U} \in \widehat{\mathsf{id}_U}, u \upharpoonright \hat{A} \times \hat{B} \in A \times B\}$$

whereas the second can only be easily presented if we use the functoriality of $\widehat{(.)}$, that is, $\widehat{\mathcal{R}; Q} = \widehat{\mathcal{R}}; \widehat{Q}$:

$$\widehat{\mathcal{R}; Q} = \{u \upharpoonright \hat{A} \times \hat{C} \mid u \in \hat{A} \times \hat{B} \times \hat{C} \mid u \upharpoonright \hat{A} \times \hat{B} \in \widehat{\mathcal{R}}, u \upharpoonright \hat{B} \times \hat{C} \in \widehat{Q}\}$$

as well as $\widehat{Q \star \mathsf{id}_U} = \widehat{Q} \times \widehat{\mathsf{id}_U}$, which follows from the monoidality of the functor $\widehat{(.)}$ which is proven in proposition 3.17. Using it, the second term unfolds to:

$$\{w \upharpoonright A' \times B \mid w \in (\hat{A}' \times \hat{U}_1) \times (\hat{A} \times \hat{U}_2) \times (\hat{B} \times \hat{U}_3), w \upharpoonright \hat{A}' \times \hat{A} \in \widehat{Q}, w \upharpoonright (\hat{A} \times \hat{U}_2) \times (\hat{B} \times \hat{U}_3) \in \widehat{\mathcal{R}}, w \upharpoonright \hat{U}_1 \times \hat{U}_3 \in \widehat{\mathsf{id}_U}, w \upharpoonright \hat{U}_2 \times \hat{U}_3 \in \widehat{\mathsf{id}_U}, w \upharpoonright \hat{A}' \times \hat{B} \in A \times B\}$$

In the second term, as $w \upharpoonright \hat{U}_1 \times \hat{U}_3 \in \widehat{\mathsf{id}_U}$ and $w \upharpoonright \hat{U}_2 \times \hat{U}_3 \in \widehat{\mathsf{id}_U}$, one can devise that $u \upharpoonright \hat{U}_1 \times \hat{U}_2 = u \upharpoonright \hat{U}_2 \times \hat{U}_3 = u \upharpoonright \hat{U}_1 \times \hat{U}_3$. The two terms only differ by the presence, in the second one, of $U_1 \times U_2$, which is insignificant (as $w \upharpoonright \hat{U}_1 \times \hat{U}_2 \in \widehat{\mathsf{id}_U}$), and the fact that $w \upharpoonright \hat{A}' \times \hat{A} \in \widehat{Q}$ whereas $u \upharpoonright A' \times A \in Q$ in the first one. At this stage, it is useful to note the following property. Let $\mathcal{P} : A \to B$ a nominal linear relation. Suppose $u \in \widehat{\mathcal{P}}$ and $u \upharpoonright \widehat{A} \in A$. Then $u \in \mathcal{P}$. This is simply proven by noticing that as $u \upharpoonright \widehat{A} \in A$, then no names have been merged. We can apply this property in our case, as $w \upharpoonright \widehat{A'} \in A'$. As a result, $w \upharpoonright \hat{A}' \times \hat{A} \in Q$ and the two sets are equal.

Next, we check the first part of vanishing property, namely that

$$\mathsf{Tr}^I_{B,C}(\mathcal{R}) = \mathcal{R}$$

which translates into:

$$\{u \upharpoonright B \times C \mid u \in \hat{\mathcal{R}} \times \mathsf{id}_I, u \upharpoonright \hat{B} \times \hat{C} \in B \times C\} = \{u \in R\}$$

The right to left inclusion is straightforward. For the left to right one, we use the property that $r \in \widehat{\mathcal{R}} \wedge r \upharpoonright \hat{B} \in B \Rightarrow r \in \mathcal{R}$, as explained above. This entails the left to right inclusion.

The second part of the vanishing property is consists in proving that given $\mathcal{R} : A \otimes U \otimes V \to B \otimes U \otimes V$, then $\mathsf{Tr}^{U \otimes V}_{A,B}(\mathcal{R}) = \mathsf{Tr}^U_{A,B}(\mathsf{Tr}^V_{A \otimes U, B \otimes U}(\mathcal{R}))$. This first term can be described by:

$$\{r \upharpoonright A \times B \mid r \in \hat{\mathcal{R}}, r \upharpoonright \hat{A} \times \hat{B} \in A \times B, r \in \hat{U} \times \hat{V} \times \hat{U} \times \hat{V} \in \widehat{\mathsf{id}_{U \otimes V}}\}$$

The second term unfolds in two steps. First, we consider $\mathsf{Tr}^V_{A \otimes U, B \otimes U}(\mathcal{R})$.

$$\mathsf{Tr}^V_{A \otimes U, B \otimes U}(\mathcal{R})) = \{u \upharpoonright (A \star U) \times (B \star U) \mid u \in \hat{R}, u \upharpoonright (\hat{A} \times \hat{U}) \times (\hat{B} \times \hat{U}) \in (A \star U) \times (B \star U), u \upharpoonright \hat{V} \times \hat{V} \in \widehat{\mathsf{id}_V}\}.$$

Then we can consider its closure $\widehat{\mathsf{Tr}^V_{A\otimes U,B\otimes U}(\mathcal{R})}$,

$$\widehat{\mathsf{Tr}^V_{A\otimes U,B\otimes U}(\mathcal{R})} = \{u \upharpoonright (\hat{A}\times\hat{U})\times(\hat{B}\times\hat{U}) \mid u\in\hat{R}, u\upharpoonright\hat{V}\times\hat{V}\in\widehat{\mathsf{id}_V}\}$$

in order to devise the whole second term.

$$\begin{aligned}\mathsf{Tr}^U_{A,B}(\mathsf{Tr}^V_{A\otimes U,B\otimes U}(\mathcal{R})) &= \{u\upharpoonright A\times B \mid u\in\widehat{\mathsf{Tr}^V_{A\otimes U,B\otimes U}(\mathcal{R})}, u\upharpoonright\hat{A}\times\hat{B}\in A\times B, u\upharpoonright\hat{U}\times\hat{U}\in\widehat{\mathsf{id}_U}\}\\ &= \{u\upharpoonright A\times B\mid u\in\hat{R}, u\upharpoonright\hat{V}\times\hat{V}\in\widehat{\mathsf{id}_V}, u\upharpoonright\hat{U}\times\hat{U}\in\widehat{\mathsf{id}_U}, u\upharpoonright\hat{A}\times\hat{B}\in A\times B\}\\ &= \{u\upharpoonright A\times B\mid u\in\hat{R}, u\upharpoonright(\hat{U}\times\hat{V})\times(\hat{U}\times\hat{V})\in\widehat{\mathsf{id}_{U\star V}}, u\upharpoonright\hat{A}\times\hat{B}\in A\times B\}\\ &= \mathsf{Tr}^{U\otimes V}_{A,B}(\mathcal{R})\end{aligned}$$

We now deal with superposing:

$$\mathsf{Tr}^U_{C\otimes A,D\otimes B}(\mathcal{Q}\otimes\mathcal{R}) = \mathcal{Q}\otimes\mathsf{Tr}^U_{A,B}(\mathcal{R}),$$

for $\mathcal{R} : A\otimes U\to B\otimes U$, and $\mathcal{Q} : C\to D$. This translates into:

$$\begin{aligned}&\{u\upharpoonright(C\star A)\times(D\star B)\mid u\in\widehat{\mathcal{Q}\star\mathcal{R}}, u\upharpoonright\hat{U}\times\hat{U}\in\widehat{\mathsf{id}_U}, u\upharpoonright(\hat{C}\times\hat{A})\times(\hat{D}\times\hat{B})\in(C\star A)\times(D\star B)\}\\ &\stackrel{?}{=}\{w\upharpoonright(C\star A)\times(D\star B)\mid w\in\mathcal{Q}\star\widehat{\mathcal{R}}, w\upharpoonright\hat{A}\times\hat{B}\in A\times B, w\upharpoonright\hat{U}\times\hat{U}\in\widehat{\mathsf{id}_U}\}\end{aligned}$$

The fact that the second set is included in the first one is the easiest inclusion to prove. It follows from $\mathcal{Q}\star\widehat{\mathcal{R}}\subseteq\widehat{\mathcal{Q}\star\mathcal{R}}$. Let us prove the reverse inclusion. As $u\in\widehat{\mathcal{Q}\star\mathcal{R}}\wedge u\upharpoonright\hat{C}\times\hat{D}\in C\times D$ then $u\upharpoonright C\times D\in\mathcal{Q}\wedge u\in\mathsf{id}_C\times\hat{\mathcal{R}}$, however, nothing imposes that $u\in\mathsf{id}_C\star\hat{\mathcal{R}}$, the names in the id_C and $\hat{\mathcal{R}}$ part of u could be the same. However, as $u\upharpoonright(\hat{C}\times\hat{A})\times(\hat{D}\times\hat{B})\in(C\star A)\times(D\times B)$, one gathers that $\nu(u\upharpoonright(\hat{A}\times\hat{U})\times(\hat{B}\times\hat{U}))\cap\nu(u\upharpoonright C\times D) = \nu(u\upharpoonright\hat{U}\times\hat{U})\cap\nu(u\upharpoonright C\times D)$. Let us suppose that this set is not empty, and let a be a name appearing in it. Then, by definition, this name does not appear in $u\upharpoonright\hat{A}\times\hat{B}$. Therefore, by applying a permutation (a,c) to $u\upharpoonright(\hat{A}\times\hat{U})\times(\hat{B}\times\hat{U})$, where c is fresh, we can remove the name a from the intersection. That is, let $u' = (u\upharpoonright C\times D)\times((a,c)\cdot u\upharpoonright(\hat{A}\times\hat{U})\times(\hat{B}\times\hat{U}))$. Then $u'\upharpoonright(\hat{C}\times\hat{A})\times(\hat{D}\times\hat{B}) = u\upharpoonright(\hat{A}\times\hat{C})\times(\hat{B}\times\hat{D})$, $u'\upharpoonright(\hat{A}\times\hat{U})\times(\hat{B}\times\hat{U})\in\widehat{\mathcal{R}}$, and $u'\upharpoonright\hat{U}\times\hat{U}\in\widehat{\mathsf{id}_U}$. Furthermore, $\nu(u'\upharpoonright(\hat{A}\times\hat{U})\times(\hat{B}\times\hat{U}))\cap\nu(u'\upharpoonright C\times D) = \nu(u\upharpoonright(\hat{A}\times\hat{U})\times(\hat{B}\times\hat{U}))\cap\nu(u\upharpoonright C\times D)\setminus\{a\}$. Since the support of u is finite, we can repeat the procedure for every name in the intersection. Finally, we obtain that an element u'' such that $u''\upharpoonright(\hat{C}\times\hat{A}\times\hat{D}\times\hat{B}) = u\upharpoonright(\hat{C}\times\hat{A})\times(\hat{D}\times\hat{B})$, $u''\in\mathcal{Q}\star\widehat{\mathcal{R}}$, and such that u'' satisfies all the conditions of the second set. Hence the two sets are equal.

Finally, we must prove the yanking property:

$$\mathsf{Tr}^A_{A,A}(s_{A,A}) = \mathsf{id}_A$$

Let us remind that $s_{A,A}$ is the isomorphism coming from the symmetry of the monoidal category, $s_{A,A} = \{(u.v, v.u) \mid u, v \in A, u\#v\}$. Hence

$$\mathsf{Tr}^A_{A,A} = \{(u, w) \in A \times A \mid \exists (v, x) \in \widehat{A} \times \widehat{A}.(u.v, w.x) \in \widehat{s_{A,A}}.(v, x) \in \widehat{\mathsf{id}_A}\}$$

As $(v, x) \in \widehat{\mathsf{id}_A}$, $v = x$. Furthermore, as $(u.v, w.v) \in \widehat{s_{A,A}}$, $u = v$, and $v = w$. As a result:

$$\mathsf{tr}^A_{A,A}(s_{A,A}) = \{(u, u) \mid u \in A\} = \mathsf{id}_A$$

□

3.4.3 Nominal polarised linear relations

As the category NomLinRel is monoidal traced, one can define a compact closed category by polarising the category as before. Let us expose that formally. The objects of NomLinRelPol will be nominal sets of annotated, polarised and separated lists. Elements of the sets are defined from the following grammar.

$$L := (a, p) \mid (\bullet, p) \mid \mathsf{inl}(L) \mid \mathsf{inr}(L) \mid L_1.L_2$$
$$\text{where } a \in \mathbb{A}, \text{ and } p \in \{-1, 1\}, \text{ and } L_1\#_{\mathsf{pol}}L_2$$

The p next to each atome is called the **polarity** of the atome. Given a list L, we define by $\mathsf{Pos}(L)$ the restriction of L to its atomes of positive polarity, and $\mathsf{Neg}(L)$ its restriction to the negative ones, where $\mathsf{Pos}(\mathsf{inl}(L)) = \mathsf{inl}(\mathsf{Pos}(L))$, $\mathsf{Pos}(\mathsf{inr}(L)) = \mathsf{inr}(\mathsf{Pos}(L))$, and similarly for Neg. We write $L_1\#_{\mathsf{pol}}L_2$ if $(\mathsf{Pos}(L_1)\#\mathsf{Pos}(L_2)) \wedge (\mathsf{Neg}(L_1)\#\mathsf{Neg}(L_2))$, and note $\star_{\mathsf{pol}}$ the associated polarised separated concatenation operation:

$$A \star_{\mathsf{pol}} B = \{L_1.L_2 \mid L_1 \in A, L_2 \in B, L_1\#_{\mathsf{pol}}L_2\}.$$

We write $L_1 \star_{\mathsf{pol}} L_2$ as a shorthand for $L_1.L_2$ knowing $L_1\#_{\mathsf{pol}}L_2$. We furthermore define the operation $(.)^\perp$ on lists, that consists in inverting all polarities.

$$(a, p)^\perp = (a, -p) \qquad (\bullet, p)^\perp = (\bullet, -p)$$
$$\mathsf{inl}(L)^\perp = \mathsf{inl}(L^\perp) \qquad \mathsf{inr}(L)^\perp = \mathsf{inr}(L^\perp) \qquad (L_1.L_2)^\perp = L_1^\perp.L_2^\perp$$

We extend this operation to sets : $A^\perp := \{L^\perp \mid L \in A\}$. Just as its seminal category, the morphisms $\mathcal{R} : A \to B$ of the category are relations. More precisely a **nominal linear polarised relation** $\mathcal{R} : A \to B$ is a nominal relation $\mathcal{R} \subseteq A^\perp \star_{\mathsf{pol}} B$ such that each element is linear between negative and positive elements. That is, $\forall x \in \mathcal{R}$, $\nu(\mathsf{Pos}(x)) = \nu(\mathsf{Neg}(x))$. The identity $\mathsf{id}_A : A \to A$ is the identity relation $L^\perp.L \in A^\perp \star_{\mathsf{pol}} A$. Note that the $\star_{\mathsf{pol}}$ extends directly to morphisms, looking at them as subsets. Given two relations $\mathcal{R}_1 : A_1 \to B_1$, and $\mathcal{R}_2 : A_2 \to B_2$, we define $R_1 \star_{\mathsf{pol}} R_2 : A_1 \star_{\mathsf{pol}} A_2 \to B_1 \star_{\mathsf{pol}} B_2$ by:

$$\mathcal{R}_1 \star_{\mathsf{pol}} \mathcal{R}_2 = \{L_{A_1}.L_{A_2}.L_{B_1}.L_{B_2} \mid L_1 = L_{A_1}.L_{B_1} \in R_1, L_2 = L_{A_2}.L_{B_2} \in \mathcal{R}_2,\ L_1 \#_{\mathsf{pol}} L_2\}$$

Note that actually, $L_1 \# L_2$, as $\nu(\mathsf{Neg}(L_1)) = \nu(\mathsf{Pos}(L_1))$ and similarly for L_2. Hence $R_1 \star_{\mathsf{pol}} \mathcal{R}_2 = \mathcal{R}_1 \star \mathcal{R}_2$.

Just as in the case of NomLinRel, the patterns inl, inr allow us to form coproducts, denoted by $\oplus$:

- $A \oplus B = \mathsf{inl}(A) \uplus \mathsf{inr}(B)$.
- $\mathcal{R}_1 \oplus \mathcal{R}_2 = \mathsf{inl}(\mathcal{R}_1) \uplus \mathsf{inr}(\mathcal{R}_2)$

where we remind that $\uplus$ expresses the fact that the union is disjoint.

Finally, given a set A, we define the two following sets:

$$\mathsf{Pos}(A) = \{\mathsf{Pos}(L) \mid L \in A\}$$
$$\mathsf{Neg}(A) = \{\mathsf{Neg}(L) \mid L \in B\}.$$

One can then easily notice that $A \subseteq \mathsf{Neg}(A) \times \mathsf{Pos}(A)$, although the inclusion is strict in general. That is, there is a injective function $A \to \mathsf{Neg}(A) \times \mathsf{Pos}(A)$, that maps a list L to $(\mathsf{Neg}(L), \mathsf{Pos}(L))$. Through this inclusion, a nominal linear polarised relation $\mathcal{R} : A \to B$ lifts to a nominal linear relation $\mathcal{R}' : \mathsf{Pos}(A) \star \mathsf{Neg}(B) \to \mathsf{Neg}(A) \star \mathsf{Pos}(B)$. Indeed, given an element $L \in \mathcal{R}$, then $\nu(\mathsf{Neg}(L)) = \nu(\mathsf{Pos}(L))$, and $L \in A^{\perp} \star_{\mathsf{pol}} B$. Hence $\mathsf{Neg}(L) = \mathsf{Pos}(L \upharpoonright A).\mathsf{Neg}(L \upharpoonright B)$, $\mathsf{Pos}(L) = \mathsf{Neg}(L \upharpoonright A).\mathsf{Pos}(L \upharpoonright B)$ and furthermore $\mathsf{Pos}(L \upharpoonright A) \# \mathsf{Neg}(L \upharpoonright B)$, $\mathsf{Neg}(L \upharpoonright A) \# \mathsf{Pos}(L \upharpoonright B)$.

Just as functions in NomLinPol did not compose as set-functions, nominal linear polarised relations do not compose as relations. The composition is defined via the trace, just as in the previous section 3.3. Let us consider two morphisms $\mathcal{R} : A \to B$ and $Q : B \to C$, seen as nominal linear relations $\mathcal{R}' : \mathsf{Pos}(A) \star \mathsf{Neg}(B) \to \mathsf{Neg}(A) \star \mathsf{Pos}(B)$ and $Q' : \mathsf{Pos}(B) \star \mathsf{Neg}(C) \to \mathsf{Neg}(B) \star \mathsf{Pos}(C)$. Their composition is defined by taking their trace along $\mathsf{Neg}(B) \times \mathsf{Pos}(B)$.

$$\mathcal{R}'; Q' = \mathsf{Tr}^{\mathsf{Neg}(B) \star \mathsf{Pos}(B)}_{\mathsf{Pos}(A),\mathsf{Neg}(C),\mathsf{Neg}(A),\mathsf{Pos}(C)}(\mathsf{id}_{\mathsf{Neg}(A)} \star s_{\mathsf{Pos}(B),\mathsf{Neg}(B)} \star \mathsf{id}_{\mathsf{Pos}(C)}) \circ (\mathcal{R}' \star Q')$$

where we recall that s is the swapping morphism coming from the symmetry. This unfolds to:

$$\begin{aligned}
\mathcal{R}'; Q' = \{&u \upharpoonright \mathsf{Pos}(A) \star \mathsf{Neg}(C) \times \mathsf{Neg}(A) \star \mathsf{Pos}(C) \mid u \in \widehat{\mathcal{R}'} \times \widehat{Q'},\\
&u \upharpoonright (\widehat{\mathsf{Neg}(B)} \times \widehat{\mathsf{Pos}(B)}) \times (\widehat{\mathsf{Pos}(B)} \times \widehat{\mathsf{Neg}(B)}) \in \widehat{s_{\mathsf{Pos}(B),\mathsf{Neg}(B)}},\\
&u \upharpoonright (\widehat{\mathsf{Pos}(A)} \times \widehat{\mathsf{Neg}(C)}) \times (\widehat{\mathsf{Neg}(A)} \times \widehat{\mathsf{Pos}(C)}) \in (\mathsf{Pos}(A) \star \mathsf{Neg}(C)) \times (\mathsf{Neg}(A) \star \mathsf{Pos}(C))\}\\
= \{&u \upharpoonright (\mathsf{Pos}(A) \star \mathsf{Neg}(C)) \times (\mathsf{Neg}(A) \star \mathsf{Pos}(C)) \mid u \in \widehat{\mathcal{R}'} \times \widehat{Q'},\\
&u \upharpoonright (\widehat{\mathsf{Neg}(B)} \times \widehat{\mathsf{Pos}(B)}) = u \upharpoonright (\widehat{\mathsf{Pos}(B)} \times \widehat{\mathsf{Neg}(B)}),\\
&u \upharpoonright (\widehat{\mathsf{Pos}(A)} \times \widehat{\mathsf{Neg}(C)}) \times (\widehat{\mathsf{Neg}(A)} \times \widehat{\mathsf{Pos}(C)}) \in (\mathsf{Pos}(A) \star \mathsf{Neg}(C)) \times (\mathsf{Neg}(A) \star \mathsf{Pos}(C))\}\}
\end{aligned}$$

As $u \in \widehat{\mathcal{R}'} \times \widehat{Q'}$, u comes from the injection $D \to \mathsf{Neg}(D) \times \mathsf{Pos}(D)$, where $D = A^\perp \times B \times B^\perp \times C$. Therefore, it is possible to backtrack u to an element of D. Doing so we get:

$$\begin{aligned}
\mathcal{R}; Q &= \{u \upharpoonright A \star_{\mathsf{pol}} C \mid u \in \widehat{A^\perp} \times \widehat{B_1} \times \widehat{B_2^\perp} \times \widehat{C}, \\
&\qquad u \upharpoonright \widehat{A^\perp} \times \widehat{B_1} \in \widehat{\mathcal{R}}, u \upharpoonright \widehat{B_2^\perp} \times \widehat{C} \in \widehat{Q}, (u \upharpoonright \widehat{B_1})^\perp = u \upharpoonright \widehat{B_2^\perp}, u \upharpoonright \widehat{A} \times \widehat{C} \in A \star_{\mathsf{pol}} C\} \\
&= \{u \in A \star_{\mathsf{pol}} C \mid \exists r_1 \in \widehat{\mathcal{R}}, r_2 \in \widehat{Q}, (r_1 \upharpoonright \widehat{B})^\perp = r_2 \upharpoonright \widehat{B}, r_1 \upharpoonright \hat{A} = u \upharpoonright A, r_2 \upharpoonright \hat{C} = u \upharpoonright C\}
\end{aligned}$$

Furthermore, as the composition of $\mathcal{R}'; Q'$ corresponds to tracing nominal linear relations, it results in a nominal linear polarised relation. That is, for each element u of $\mathcal{R}'; Q'$, we have that $\nu(u \upharpoonright \mathsf{Pos}(A) \star \mathsf{Neg}(C)) = \nu(u \upharpoonright \mathsf{Neg}(A) \star \mathsf{Pos}(C))$. Therefore, backtracking u into an element of $A \star_{\mathsf{pol}} C$, we get that $\nu(\mathsf{Neg}(u)) = \nu(\mathsf{Pos}(u))$, that is, the relation $\mathcal{R}; Q$ hence defined is linear polarised. The closure under permutation follows straightforwardly from the invariance under permutation of the definitions. Therefore $\mathcal{R}; Q$ is indeed a morphism $A \to C$ of NomLinRelPol.

To sum up, the paradigm for composition is: closure under strict substitutions + relational composition + projections on "good" elements. Just as above, we restrict the objects to the well-typed ones. We define $I = \{(\bullet, 1)\}$, $[\![X]\!] = \{(a, 1) \mid a \in \mathbb{A}_X\}$, $0 = \emptyset$, and present formally NomLinRelPol in the following definition.

Definition 3.22. NomLinRelPol *is the category that has as objects the smallest set such that* $I, [\![X]\!], 0 \in \mathsf{Obj}(\mathsf{NomLinRelPol})$ *and*

$$\begin{aligned}
A, B \in \mathsf{Obj}(\mathsf{NomLinRelPol}) &\Rightarrow A \star_{\mathsf{pol}} B \in \mathsf{Obj}(\mathsf{NomLinRelPol}) \\
A, B \in \mathsf{Obj}(\mathsf{NomLinRelPol}) &\Rightarrow A \oplus B \in \mathsf{Obj}(\mathsf{NomLinRelPol}) \\
A \in \mathsf{Obj}(\mathsf{NomLinRelPol}) &\Rightarrow (A)^\perp \in \mathsf{Obj}(\mathsf{NomLinRelPol})
\end{aligned}$$

Hence objects of NomLinRelPol *are nominal sets of polarised separated annotated lists. Morphisms $A \to B$ of* NomLinRelPol *are nominal polarised linear relations $A^\perp \star_{\mathsf{pol}} B$ as described above.*

By construction, NomLinRelPol is a compact closed category, since it is equivalent to Int(NomLinRel). We furthermore show that given two objects A, B, the object $A \oplus B$ is their coproduct, and that 0 is the unit for it, that is, an initial object. Namely, we need to exhibit two morphisms $\mathsf{inl} : A \to A \oplus B$ and $\mathsf{inr} : B \to A \oplus B$, such that for every object C, for every morphisms $\mathcal{R}_A : A \to C$ and $\mathcal{R}_B : B \to C$, there exists a unique nominal polarised relation $Q : A \oplus B \to C$ such that $\mathsf{inl}; \tau = \mathcal{R}_A$ and $\mathsf{inr}; Q = \mathcal{R}_B$. The corresponding diagram is showcased in Figure 3.4. The morphism $\mathsf{inl} : A \to A \oplus B$ is defined to be the nominal polarised relation $\mathsf{inl}_A = \{u^\perp.\mathsf{inl}(u) \mid u \in A\}$ and $\mathsf{inr} : B \to A \oplus B$ is built in a similar fashion. Finally, we set Q as follows:

$$Q = \{\mathsf{inl}(u).v \mid u.v \in \mathcal{R}_A\} \uplus \{\mathsf{inr}(u).v \mid u.v \in \mathcal{R}_B\}$$

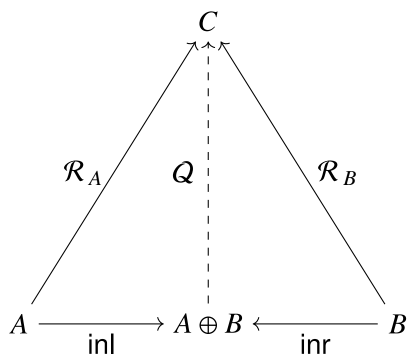


Figure 3.4: Coproduct diagram of $A \oplus B$

Then one can straighforwardly check that $\mathsf{inl}; Q = \mathcal{R}_A$, and similarly for inr. Furthermore, Q is unique verifying this property. Hence $A \oplus B$ is the coproduct of $A \oplus B$. 0 is initial since for any object A, there is a unique morphism $0 \to A$, namely the empty relation. As the co-product distributes over the tensor it is a bi-product, as established in [44]. Hence we can conclude this paragraph with the following proposition.

Proposition 3.23. NomLinRelPol *is a star-autonomous category with products, and therefore, a model of* MALL.

One could wonder what is the "canonical" logic associated with the nominal linear polarised relations. In other terms, what is the sequent calculus associated with compact closed categories with bi-products. Surely, one can present a collapsed version of MLL, where $\otimes = ⅋$ and $\& = \oplus$. However, as explained in [86], "*it turned out that there is a counter-example to the cut-elimination*". On the other hand, in the context of strongly compact closed category, Abramsky and Duncan were able to define a notion of proof-net [4] that precisely define those morphisms arising from the categorical structure, although no presentation was given in terms of sequent calculus.

As a final remark, these relations are the relations that we should use in the second part of the thesis devoted to nominal games to tensorial logic. Before moving to the next chapter, we present a severe downside of nominal polarised relations, and show in Section 3.5 a model with hypercoherences that overcomes it.

3.4.4 Nominal linear relations, a downside

Nominal relations suffer from that, even restricted to denotations of MLL formulas, they do not define linkings properly. For instance, consider the following relation $\mathcal{R} : X \otimes X \to X \otimes X$ defined by:

$$\mathcal{R} = \{(u.v.v.u) \mid u, v \in \mathbb{A}_X\} \cup \{(u.v.u.v) \mid u, v \in \mathbb{A}_X\}. \tag{3.25}$$

Then $\mathcal{R}$ implements two possible choices of axiom links for the sequent canonically associated with the previous formula. As long as the category remains compact-closed, there is no simple solution to this problem. For instance, let us implement a simple-minded criterion that imposes that nominal relations implements at most a unique linking per additive resolution. Given $\mathcal{R} : A \to B$, we write $\mathcal{R}'$ for the linear relation $\mathsf{Pos}(A) \star \mathsf{Neg}(B) \to \mathsf{Neg}(A) \star \mathsf{Pos}(B)$ that canonically corresponds to $\mathcal{R}$. Let us consider the following criterion:

$$x\mathcal{R}'y \ \wedge \ x\mathcal{R}'z \ \wedge \ y \simeq z \Rightarrow y = z$$

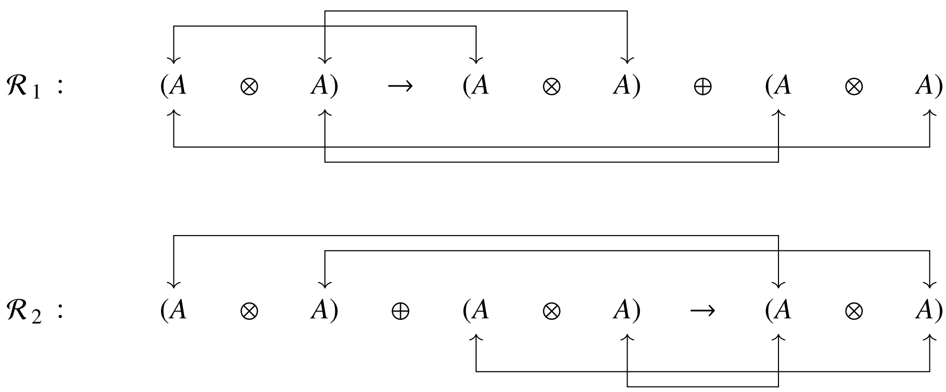
It basically enforces that on a additive resolution (that is a choice, for each biproduct, of one of its branches), the relation can only implement one set of axiom links.

This property is symmetric. That is, it automatically entails :

$$x\mathcal{R}'z \ \wedge \ y\mathcal{R}'z \ \wedge \ x \simeq y \Rightarrow x = y$$

Indeed, take π such that $x = \pi \cdot y$. Then $x\mathcal{R}'z \wedge x\mathcal{R}'\pi \cdot z$ as the relation $\mathcal{R}$ is closed under permutations. So it directly entails that $z = \pi \cdot z$. As the elements have strong support, it entails $\pi \# z$, and as $\nu(z) = \nu(x) = \nu(y)$, $\pi \cdot y = y$ and therefore $x = y$.

Unfortunately, this property does not compose. If we examine the relations associated with the two following proof structures:



then we notice that the composition $\mathcal{R}_1; \mathcal{R}_2$ of these two leads to the problematic relation $\mathcal{R}$ of 3.25, described at the beginning of this section.

3.4.5 Lax polarised nominal relations

In this section, we show that the lax polarised nominal relations form a compact-closed category with products as well, although the separation property is dropped. This technical result will be used later in the discourse of Section 3.5.

Just as the closure under strict substitutions of separated nominal relations leads to a definition of a functor to a new category, called lax nominal relations, the closure under strict substitutions of separated, polarised nominal relations defines a functor from the compact closed category to a new one of lax polarised nominal relations called LaxNomLinPol.

This category is similar to NomLinRelPol except that the lists are not "polarised and separated" anymore. That is, the grammar for the lists is :

$$L = (\bullet, p) \ \mid \ (a, p) \ \mid \ \mathsf{inl}(L) \ \mid \ \mathsf{inr}(L) \ \mid L_1.L_2$$

where $a \in \mathbb{A}$, and $p \in \{-1, 1\}$.

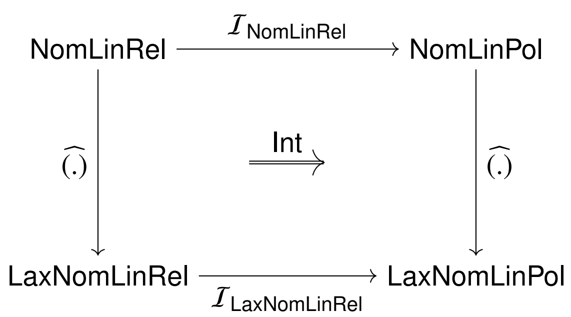
The tensor product $\otimes$ is now simply the cartesian product, except that, as usual, we concatenate lists instead of pairing them. The $\oplus$ and the negation are similar to the ones of NomLinRelPol. The morphisms are nominal linear polarised relations.

The functor $\widehat{(.)}$ is defined by sending a relation to its closure under strict substitutions, and on objects as follows :

$$\widehat{\mathbb{A}_X} = \mathbb{A}_X \qquad \widehat{\{\bullet\}} = \{\bullet\}$$
$$\widehat{A_1 \star_{\mathsf{pol}} A_2} = \hat{A}_1 \times \hat{A}_2 \qquad \widehat{A \oplus B} = \hat{A}_1 \oplus \hat{A}_2 \qquad \widehat{A^\perp} = \hat{A}^\perp$$

In order to check that this is a functor, we again have to prove that $\widehat{R;Q} = \widehat{R};\widehat{Q}$, given $\mathcal{R} : A \to B$ and $Q : B \to C$. This follows from seeing $\mathcal{R}$ as a nominal linear relation: $\mathcal{R} : \mathsf{Pos}(A) \star \mathsf{Neg}(B) \to \mathsf{Neg}(A) \star \mathsf{Pos}(B)$ (and the similar form for Q), and relying on the fact that $\widehat{(.)}$ indeed defines a monoidal functor when restricted to nominal linear separated relations.

Finally, one can see that LaxNomLinPol is the free compact closure of LaxNomLinRel. Just as for NomLinRel, it results from a slight variation of the Int construction of LaxNomLinPol. So we have the following diagram. Writing $\mathcal{I}_{\mathsf{NomLinRel}}$ for the inclusion functor $\mathcal{I}_{\mathsf{NomLinRel}}$: $\mathsf{NomLinRel} \to \mathsf{NomLinPol}$, and $\mathcal{I}_{\mathsf{LaxNomLinRel}}$ for its lax equivalent we have:



In particular, $\widehat{(.)}$ defines a functor of compact closed categories between NomLinPol and LaxNomLinPol. as it commutes with the negation.

3.5 Nominal hypercoherence spaces

3.5.1 The category

To refine the previous model in order to avoid bad relations, one can use the notion of **coherence**, developed by Girard in the so-called coherence spaces, to express the fact that the bad relations, such as the $\mathcal{R}$ in equation 3.25 above, are incoherent. As this notion has since been refined, in [27] into hypercoherence, we present the latter. Furthermore, coherence removes a second downside of nominal linear polarised relations, their degeneracy. Nominal hypercoherence spaces differentiate between $\otimes$ and $⅋$, and between $\oplus$ and $\&$.

All the definitions are taken from [27], just being modified to take into account the fact that our atomes are nominal, and polarised. More precisely, the elements of our nominal sets, renamed to “webs”, are, as before, separated annotated lists of polarised atomes. We write $\mathcal{P}_{\mathsf{fin}}(A)$ for the set of finite subsets of any set A, and $\mathcal{P}^*_{\mathsf{fin}}(A)$, for the set of non-empty finite subsets. Similarly, we write $w \subseteq^*_{\mathsf{fin}} A$ to mean $w \in \mathcal{P}^*_{\mathsf{fin}}(A)$.

Definition 3.24. *A **nominal hypercoherence space** $A = (|A|, \Gamma(A))$ consists of:*

- *A nominal enumerable set $|A|$, called **web**, of annotated, polarised and separated lists.*
- *A nominal subset $\Gamma(A) \subseteq \mathcal{P}^*_{\mathsf{fin}}(|A|)$, such that all singletons are in $\Gamma(A)$.*

This structure on objects allows us to discriminate between relations, thanks to these definitions.

Definition 3.25. *Let $(|A|, \Gamma(A))$ be a hypercoherence space.*

- *A non-empty finite subset $v \subseteq^*_{\mathsf{fin}} |A|$ is **coherent** when $v \in \Gamma(A)$.*
- *A set of elements $\mathcal{R} \subseteq |X|$ forms a **clique** when:*
  1. *$\forall v \subseteq^*_{\mathsf{fin}} \mathcal{R}$, v is coherent.*
  2. *$\mathcal{R}$ is closed under permutations.*
  3. *$\mathcal{R}$ is linear: $\forall x \in \mathcal{R}$, $\nu(\mathsf{Neg}(x)) = \nu(\mathsf{Pos}(x))$.*

That is, a clique is a refinement of a nominal linear polarised relation. Cliques are the underlying structure behind the morphisms of the category of hypercoherence nominal spaces, that we shall define below. Given $\Gamma(A) \subseteq \mathcal{P}^*_{\mathsf{fin}}(A)$, we write $\Gamma^*(A)$ for the subset of $\Gamma(A)$ of sets of cardinality greater than one, and say that such sets are **strictly coherent**. We write $\Gamma^{\perp}(A)$ for the set of non-empty subsets that are **incoherent**: $\Gamma^{\perp}(A) = \mathcal{P}^*_{\mathsf{fin}}(A) \setminus \Gamma^*(A)$, and $\Gamma^{\perp,*}(A)$ for those that are **strictly incoherent**, meaning incoherent and of cardinality greater than one. Finally, we set $(\Gamma(A))^{\perp} = \{w^{\perp} \mid w \in \Gamma(A)\}$, where $w^{\perp} = \{x^{\perp} \mid x \in w\}$.

The **dual** $A^{\perp}$ of a hypercoherence space $(|A|, \Gamma(A))$ is defined as follows:

- $|A^{\perp}| = |A|^{\perp}$
- $\Gamma(A^{\perp}) = \mathcal{P}^*_{\mathsf{fin}}(|A|^{\perp}) \setminus (\Gamma^*(A))^{\perp}$

We now give the interpretations of the connectives of linear logic. In the sequel, for simplicity, we will write $w \upharpoonright A$ for $w \upharpoonright |A|$. The tensor product of two hypercoherence spaces $A \otimes B$ is defined as follows, where $\star_{\mathsf{pol}}$ is the polarised separated product as defined in section 3.4.3.

- $|A \otimes B| = |A| \star_{\mathsf{pol}} |B|$
- $\Gamma(A \otimes B) = \{w \in \mathcal{P}^*_{\mathsf{fin}}(|A| \star_{\mathsf{pol}} |B|) \mid w \upharpoonright A \in \Gamma(A) \wedge w \upharpoonright B \in \Gamma(B)\}$

The unit of the tensor product is $I = (\{(\bullet, 1)\}, \{\{(\bullet, 1)\}\})$. We can define $A ⅋ B$ through the De-Morgan duality $(A ⅋ B) = (A^\perp \otimes B^\perp)^\perp$.

- $|A ⅋ B| = |A| \star_{\mathsf{pol}} |B|$
- $\Gamma^*(A ⅋ B) = \{w \in \mathcal{P}^*_{\mathsf{fin}}(|A| \star_{\mathsf{pol}} |B|) \mid (w \upharpoonright A \in \Gamma^*(A) \vee w \upharpoonright B \in \Gamma^*(B))\}$

⅋ has almost the same unit as the tensor product, except the atom is negated: $\perp = (\{(\bullet, -1)\}, \{\{(\bullet, -1)\}\})$. As a result, we can define the connective $\multimap$, with $A \multimap B = A^\perp ⅋ B$.

- $|A \multimap B| = |A|^\perp \star_{\mathsf{pol}} |B|$
- $\Gamma(A \multimap B) = \{w \in \mathcal{P}^*_{\mathsf{fin}}(|A \multimap B|) \mid (w \upharpoonright |A|^\perp \in \Gamma(A)^\perp \Rightarrow w \upharpoonright |B| \in \Gamma(B))$
  $\wedge (w \upharpoonright |A|^\perp \in (\Gamma^*(A))^\perp \Rightarrow w \upharpoonright |B| \in \Gamma^*(B)\}$

In the future, we will simply write $w \upharpoonright A$ for $w \upharpoonright A^\perp$, the fact that the polarities in A are reversed being clear from the context. Finally, we define the additives. We start with $A \,\&\, B$, writing $\uplus$ for the disjoint union.

- $|A \,\&\, B| = \mathsf{inl}(|A|) \uplus \mathsf{inr}(|B|)$
- $\Gamma(A \,\&\, B) = \{w \in \mathcal{P}^*_{\mathsf{fin}}(|A \,\&\, B|) \mid (w \upharpoonright \mathsf{inl}(A) = \emptyset \Rightarrow w \upharpoonright \mathsf{inr}(B) \in \mathsf{inr}(\Gamma(B))) \wedge (w \upharpoonright \mathsf{inr}(B) = \emptyset \Rightarrow w \upharpoonright \mathsf{inl}(A) \in \mathsf{inr}(\Gamma(A)))\}$

We finish with $A \oplus B$, that is the dual of $A \,\&\, B$, in the sense that $(A \oplus B) = (A^\perp \oplus B^\perp)^\perp$

- $|A \oplus B| = \mathsf{inl}(|A|) \uplus \mathsf{inr}(|B|)$
- $\Gamma(A \oplus B) = \{w \in \mathcal{P}^*_{\mathsf{fin}}(|A \oplus B|) \mid (w \upharpoonright A = \emptyset \wedge w \upharpoonright B \in \Gamma(B)) \vee (w \upharpoonright B = \emptyset \wedge w \upharpoonright A \in \Gamma(A))\}$

The empty set acts as units for the two additive connectives. We therefore have the following denotation function:

- $[\![1]\!] = I$, $[\![\perp]\!] = \perp$.
- $[\![\top]\!] = [\![0]\!] = (\emptyset, \emptyset)$.
- We define the interpretation of an atomic type $[\![X]\!] = (|X|, \Gamma(X))$ as being the following hypercoherence space:
  - $|X| = \{(a, 1) \mid a \in \mathbb{A}_X\}$
  - $\Gamma(X) = \{\{(a, 1)\} \mid a \in \mathbb{A}_X\}$

In the following, we simply write X for $[\![X]\!]$.

Definition 3.26. NomHypCoh *is the category of nominal linear polarised hypercoherent relations (often abbreviated nominal hypercoherent relations, or cliques) that has as objects the* *smallest set such that* $I, X, 0 \in \mathsf{Obj(NomHypCoh)}$ *and*

$$\begin{aligned} A, B \in \mathsf{Obj(NomHypCoh)} &\Rightarrow A \otimes B \in \mathsf{Obj(NomHypCoh)} \\ A, B \in \mathsf{Obj(NomHypCoh)} &\Rightarrow A \oplus B \in \mathsf{Obj(NomHypCoh)} \\ A \in \mathsf{Obj(NomHypCoh)} &\Rightarrow (A)^{\perp} \in \mathsf{Obj(NomHypCoh)} \end{aligned}$$

Morphisms $A \to B$ *of* NomHypCoh *are cliques of* $A \multimap B$.

The composition of cliques in NomHypCoh comes from the composition of nominal linear polarised relations. The rest of this section is devoted to check that the composition is well defined. More precisely, we have to ensure that the resulting linear nominal polarised relation is indeed a clique. To do that, we introduce the notation $\widehat{\Gamma(A)}$, to denote the appropriate notion of coherence when we close will close the cliques under strict substitution. That is, $\widehat{\Gamma(A)} \subseteq \mathcal{P}^*_{\mathsf{fin}}(|\widehat{A}|)$, and $(|\widehat{A}|, \widehat{\Gamma(A)})$ forms a hypercoherence space. Given a clique $\mathcal{R}$, after closure, $\widehat{\mathcal{R}}$ is not a subset of $A \multimap B$ anymore but a subset of $\widehat{A \multimap B}$, and we must consider a new notion of coherence for this space. Given a formula F, $\widehat{\Gamma(F)}$ is defined exactly as $\Gamma(F)$, except that we consider cartesian product instead of separated product. This is defined formally below.

Definition 3.27. *The category* LaxHypCoh *is the category of lax nominal linear polarised hypercoherent relations (often abbreviated lax nominal hypercoherent relations, or lax cliques), that has as set of objects the smallest set containing the elements* $(|\widehat{A}|, \widehat{\Gamma(A)})$ *subsequently defined, and closed under the operations defined below.*

- $\widehat{I} = I_{\mathsf{NomHypCoh}}$, $\widehat{\perp} = \perp_{\mathsf{NomHypCoh}}$, $\widehat{0} = 0_{\mathsf{NomHypCoh}}$.
- $\widehat{X} = X_{\mathsf{NomHypCoh}}$, $\widehat{X^{\perp}} = X^{\perp}_{\mathsf{NomHypCoh}}$.
- $\widehat{A \otimes B} = (|\widehat{A}| \times |\widehat{B}|, \widehat{\Gamma(A)} \times \widehat{\Gamma(B)})$
- $\widehat{A \parr B} = (|\widehat{A}| \times |\widehat{B}|, \Gamma^*(\widehat{A \parr B}) = \{w \in \mathcal{P}^*_{\mathsf{fin}}(|\widehat{A}| \times |\widehat{B}|) \mid w \upharpoonright |\widehat{A}| \in \widehat{\Gamma^*(A)} \vee w \upharpoonright |\widehat{B}| \in \widehat{\Gamma^*(B)}\}$.
- $(\widehat{A \,\&\, B}) = (\mathsf{inl}(|\widehat{A}|) \uplus \mathsf{inr}(|\widehat{B}|), \{w \in \mathcal{P}^*_{\mathsf{fin}}(|\widehat{A \,\&\, B}|) \mid w \subseteq \mathsf{inl}(|\widehat{A}|) \Rightarrow w \in \mathsf{inl}(\widehat{\Gamma(A)})$
  $\wedge\, w \subseteq \mathsf{inr}(|\widehat{B}|) \Rightarrow w \in \mathsf{inr}(\widehat{\Gamma(|B|)})\}$
- $\widehat{A \oplus B} = (\mathsf{inl}(|\widehat{A}| \uplus \mathsf{inr}(|\widehat{B}|, \mathsf{inl}(\widehat{\Gamma(A)}) \uplus \mathsf{inr}(\widehat{\Gamma(B)})$.
- $\widehat{A^{\perp}} = (|\hat{A}|^{\perp}, \mathcal{P}^*_{\mathsf{fin}}(|\hat{A}|^{\perp}) \setminus \{w^{\perp} \mid w \in \widehat{\Gamma^*(A)}\})$

The morphisms $A \to B$ *are cliques of* $A^{\perp} \parr B$, *where seeing* A *has a formula of linear logic,* $A^{\perp}$ *is the lax hypercoherence space that corresponds to its negation.*

Note that the definitions are compatible with the De-Morgan formula, that is $\widehat{\Gamma(A^{\perp})} = \widehat{\Gamma(A)}^{\perp}$. Indeed, these hold for the base cases, and the definitions of $A \parr B$ and $A \oplus B$ are settled according to this formula. The composition of cliques in LaxNomLinPol follows from the one of lax nominal linear polarised relations. That is, it is simply their relational composition (while forgetting polarities in the middle), and it hence follows that LaxHypCoh is simply a subcategory of the category of lax nominal linear polarised relation. Notably, the identity morphism is simply the identity relation.

In order to prove that the composition of separated relations is compatible with the definition of cliques, we proceed by steps and start with this lemma.

Lemma 3.28. • *Let* $w \subseteq^*_{\mathsf{fin}} |A|_{\mathsf{NomHypCoh}}$, *such that* $w \in \widehat{\Gamma^*(A)}$. *Then* $w \in \Gamma(A)$.
• , $\Gamma(A) \subseteq \widehat{\Gamma(A)}$.

Proof. The proof is done by induction on the structure of the formula A. For the first point, the base cases $X, I, 0, \top, \bot$ are immediate, as they are equal, and the inductive cases are automatic.

The first point is needed to prove the second. Just as in the first point, the proof is done by induction on the structure of the formulas, the base case and the inductive cases for $\otimes, ⅋, \oplus$ being immediate. Remaining is the case for &, setting $A = A_1 \,\&\, A_2$. If $w \upharpoonright A_1 \in \mathsf{inl}(\widehat{\Gamma(A_1)})$, then as $w \subseteq^*_{\mathsf{fin}} (A_1)$, it entails, as proven in the first part of the proof, that $w \upharpoonright A_1 \in \mathsf{inl}(\Gamma(A_1))$. Therefore $w \upharpoonright A_2 \in \mathsf{inl}(\Gamma(A_2))$ and by induction $w \upharpoonright A_2 \in \widehat{\Gamma(A_2)}$. The second case is dealt with on an equal footing, and therefore $\Gamma(A_1 \,\&\, A_2) \subseteq \widehat{\Gamma(A_1 \,\&\, A_2)}$. To conclude, $\Gamma(A) \subseteq \widehat{\Gamma(A)}$. □

Lemma 3.29. *Let* $\mathcal{R}$ *be a clique of* A. *Then* $\widehat{\mathcal{R}}$ *is a clique of* $\hat{A}$.

In other terms, this expresses that $\widehat{(.)}$ acts seemingly as a functor from hypercoherent linear nominal polarised relations to hypercoherent lax nominal polarised relations, assuming they indeed form a category. The main property to prove is that for every finite subset $w \subseteq_{\mathsf{fin}} \widehat{\mathcal{R}}$, then w is lax coherent: $w \in \widehat{\Gamma(A)}$. For the proof we will rely on the following property, whose proof is immediate.

Proposition 3.30. *Let* $w, w' \subseteq^*_{\mathsf{fin}} (|A|)$, *such that, for any occurrence of atomic formula* X *within* A *we have:*

- $w \upharpoonright X \neq \emptyset \Leftrightarrow w' \upharpoonright X \neq \emptyset$
- $w \upharpoonright X \in \Gamma(X) \Leftrightarrow w' \upharpoonright X \in \Gamma(X)$.
- $w \upharpoonright X \in \Gamma^*(X) \Leftrightarrow w' \upharpoonright X \in \Gamma^*(X)$.
- $w \upharpoonright X \in \Gamma^\perp(X) \Leftrightarrow w' \upharpoonright X \in \Gamma^\perp(X)$.
- $w \upharpoonright X \in \Gamma^{\perp,*}(X) \Leftrightarrow w' \upharpoonright X \in \Gamma^{\perp,*}(X)$.

Then $w \in \Gamma(A)$ *(respectively* $\Gamma^*(A), \Gamma^\perp(A), \Gamma^{*,\perp}(A)$*)* $\Leftrightarrow$ $w' \in \Gamma(A)$ *(respectively* $\Gamma^*(A), \Gamma^\perp(A), \Gamma^{*,\perp}(A)$*)*.

The proof of the proposition is a simple induction. The proposition is true in any category with hypercoherence, such as NomHypCoh or LaxHypCoh.

Proof of lemma 3.29. Let $\mathcal{R}$ be a clique of $[\![A]\!]$. Let $w \subseteq_{\mathsf{fin}} \widehat{\mathcal{R}}$ and $w = \{e_1 \cdot x_1, ..., e_n \cdot x_n\}$ where $\{x_1, ..., x_n\} \subseteq \mathcal{R}$ and $e_1, .., e_n \in \Xi$. We do the proof by induction on the sum of the lengths of the substitutions $|e_1| + ... + |e_n|$. If the sum is 0, then $w = \{x_1, ..., x_n\} \in \mathcal{R}$ and therefore $w \in \Gamma(A) \subseteq \widehat{\Gamma(A)}$.

So suppose the sum is equal to $n+1$, and we proved the property up to n. Writing x'_i for $e_i \cdot x_i$, we know that $w = \{x'_1, ..., x'_n\} \in \widehat{\Gamma(A)} \cap \mathcal{P}(\widehat{\mathcal{R}})$ and we want to prove that $\{[a/b] \cdot x'_1, ..., x'_n\} \in \widehat{\Gamma(A)}$. The first case to tackle is the case where $b \notin \nu(x'_1)$. Then $[a/b] \cdot x'_1 = (a, b) \cdot x'_1$, and $(a, b) \cdot (e_1 \cdot x_1) = ((a, b) \cdot e_1) \cdot ((a, b) \cdot x_1)$. As $\mathcal{R}$ is closed under permutations, $(a, b) \cdot x_1 \in \mathcal{R}$, and the set $\{(a,b)\cdot x_1,...,x_n\}$ is a subset of $\mathcal{R}$. Furthermore, w is now equal to $\{((a,b)\cdot e_1)\cdot((a,b)\cdot x_1), e_2\cdot x_2,...,e_n\cdot x_n\}$. The length of the substitutions applied on this set being n, we conclude by induction hypothesis that w is lax-hypercoherent: $\{[a/b]\cdot x'_1,....,x'_n\}\in\widehat{\Gamma(A)}$.

The harder case is when $b\in\nu(x'_1)$. Then let us consider a permutation (b,c), such that $c\#\{x'_1,...,x'_n,a\}$. We know that $\{(b,c)\cdot x'_1,...,x'_n\}\in\widehat{\Gamma(A)}$, as proven above. Finally, let us apply the substitution $[a\to c]$ to x'_1. Applying it changes the lax coherence as much as a permutation (a,d) when d is fresh. This follows from noticing that $\{[a/c]\cdot(b,c)\cdot x'_1,...,x'_n\}\upharpoonright X$ has same coherence as $\{(a,d).(b,c)\cdot x'_1,...,x'_n\}\upharpoonright X$ for every occurrence X of atomic formula in A (since both a,d are fresh for $x'_2,..,x'_n$). As a result, if $\{(a,d).(b,c)\cdot x'_1,...,x'_n\}$ belongs in $\widehat{\Gamma(A)}$, then so does $\{[a/d].(b,c)\cdot x'_1,...,x_n\}$. As the first member of the previous sentence indeed is lax hypercoherent (for the same reasons as explained above), then $\{[a/c]\cdot(b,c)\cdot x'_1,...,x_n\}\in\widehat{\Gamma(A)}$. Finally, we apply the permutation (b,c) back , noticing that $(b,c).[a\to c].(b,c)\cdot x'_1=[a\to b]\cdot x'_1$, and therefore as $\{[a/c].(b,c)\cdot x'_1,...,x'_n\}\in\widehat{\Gamma(A)}\cap\mathcal{P}(\widehat{\mathcal{R}})$, this entails $\{(b,c).[a/c].(b,c)\cdot x'_1,...,x'_n\}\in\widehat{\Gamma(A)}\cap\widehat{\mathcal{R}}$, or, equivalently, $w=\{[a/b]\cdot x'_1,...,x'_n\}\in\widehat{\Gamma(A)}\cap\mathcal{P}(\widehat{\mathcal{R}})$

□

Lemma 3.31. *Let* $w\subseteq_{\mathsf{fin}}\widehat{A\multimap B}$ *such that* $w\in\widehat{\Gamma(A\multimap B)}$. *If* $w\subseteq_{\mathsf{fin}} A\multimap B$ *then* $w\in\Gamma(A\multimap B)$.

The proof is straightforward.

Proposition 3.32. *Let* $\mathcal{R}:A\multimap B$ *and* $\mathcal{Q}:B\multimap C$ *be two linear cliques. Then* $\mathcal{R};\mathcal{Q}$ *is a linear clique of* $A\multimap C$.

Proof. The linearity and nominal closure follows from the composition of nominal linear polarised relations. Therefore, in rest of this proof, we forget about the local polarities of the lists. Let $w\subseteq^*_{\mathsf{fin}}\mathcal{R};\mathcal{Q}$, and let $v\subseteq^*_{\mathsf{fin}}\widehat{A}\times\widehat{B}\times\widehat{C}$ a witness of interaction; that is, for all x in w there exists a unique $y\in v$ such that $y\in\widehat{\mathcal{R}};\widehat{\mathcal{Q}}$ and $y\upharpoonright A\multimap C=x$. Then suppose $w\upharpoonright A\in\Gamma(A)$, then $v\upharpoonright A\in\widehat{\Gamma(A)}$. As $v\upharpoonright\widehat{A}\multimap\widehat{B}\subseteq^*_{\mathsf{fin}}\widehat{\mathcal{R}}$, then $v\upharpoonright\widehat{A}\multimap\widehat{B}\in\Gamma(\widehat{A\multimap B})$, entailing $v\upharpoonright\widehat{A}\in\widehat{\Gamma(A)}\Rightarrow v\upharpoonright\widehat{\Gamma(B)}$. Hence $v\upharpoonright\widehat{B}\in\widehat{\Gamma(B)}$. Doing the same for $v\upharpoonright\widehat{B}\multimap\widehat{C}$, we conclude that $v\upharpoonright\widehat{C}\in\widehat{\Gamma(C)}$. Now as $w\upharpoonright C=v\upharpoonright\widehat{C}$ then $w\upharpoonright C\in\Gamma(C)$. To sum up, $\forall w\subseteq^*_{\mathsf{fin}}\mathcal{R};\mathcal{Q}.w\upharpoonright A\in\Gamma(A)\Rightarrow w\upharpoonright C\in\Gamma(C)$. A similar reasoning holds for proving $\forall w\subseteq^*_{\mathsf{fin}}\mathcal{R};\mathcal{Q}.w\upharpoonright A\in\Gamma^*(A)\Rightarrow w\upharpoonright C\in\Gamma^*(C)$.That is, $\forall w\subseteq^*_{\mathsf{fin}}\mathcal{R};\mathcal{Q}.w\in\Gamma(A\multimap C)$. □

Therefore NomHypCoh forms a category, and the operation $\widehat{(.)}$ defines a star-autonomous functor from NomHypCoh to LaxHypCoh. This is a direct consequence of $\widehat{(.)}$ being a functor of compact closed categories between their underlying categories NomLinPol and LaxNomLinPol.

3.5.2 Properties of nominal hypercoherence spaces

The hypercoherence condition is strong enough to ensure that, on an additive resolution, the relation will define exactly one set of linkings. We make that clear in the following proposition. In the sequel, Δ denotes a sequent. Let us note that each element of a clique defines an additive resolution, and a linking. For instance, a list of $A \oplus B$ (respectively $A \,\&\, B$) is either a list of A, or a list of B. Furthermore, as each element x comes from a nominal linear polarised relation, and that the list is nominal separated, one can associate to it a function relating its negative and positive literals, as done in Section 3.1.2.1. This one precisely checks the definition of a linking. Note that two elements on the same additive resolution defines the same linking if they are equivalent. This follows from the list being polarised separated.

Proposition 3.33. *Let $\mathcal{R} : [\![\Delta]\!]$ a clique, and let us pick an element x of $\mathcal{R}$. This one defines an additive resolution on Δ, together with a set of linkings. Then every element y of $\mathcal{R}$ on the same additive resolution is equivalent to x.*

For the proof we rely on what we proved in the above lemma: if $\mathcal{R}$ is a clique, then $\hat{\mathcal{R}}$ is a lax-clique.

Proof. We prove the property by contradiction, assuming that there is a y that implements a different linking on the same additive resolution.

We write λ_x for the linking that x encodes, and $\Delta \upharpoonright \lambda_x$ for the associated additive resolution (and similarly for y). So let $X^\perp$ be an occurrence of an atomic variable of $\Delta \upharpoonright \lambda_x$, such that $\lambda_x(X^\perp) \neq \lambda_y(X^\perp)$.

Now, let a the name that appears for $X^\perp$ in x, and b the name that appears for $X^\perp$ in y. We apply a permutation (b, a) to y so that they share the same name a for this location. Furthermore, let c_i be a name such that c_i has same sort as a_i. We apply substitutions $[a_i/c_i]$ to x for all names $a_i \in \nu(x) \setminus a$, and similarly for y. Hence in all locations l of $\Delta \upharpoonright \lambda_x$ different than $X^\perp$ and $\lambda_x(X^\perp)$, $x \upharpoonright l = c$. Similarly, in all locations l of $\Delta \upharpoonright \lambda_x$ different than $X^\perp$ and $\lambda_y(X^\perp)$, we got $y \upharpoonright l = c$. Furthermore, $x \upharpoonright X^\perp = y \upharpoonright X^\perp = a$. So basically, we obtain two elements (that we keep on calling with their original names) x, y that are equal on all locations except $\lambda_x(X^\perp), \lambda_y(X^\perp)$ where they differ and are strictly incoherent.

We show that $\{x, y\} \notin \Gamma(\widehat{[\![\Delta]\!]})$ by induction on the structure of Δ, seen as a unique formula (based on the equivalence between the interpretation of $\vdash F_1, .., F_n$ and $\vdash F_1 \parr ... \parr F_n$) . We prove the following intermediate property: if two elements $x, y \in [\![F]\!]$ are strictly incoherent on one or several locations and equal everywhere else, then they are strictly incoherent.

We call X the location where they are strictly incoherent. The base case consists in F being this single location. Then they are incoherent in F by definition. So there are now two induction cases to tackle : $F = F_1 \otimes F_2$ or $F = F_1 \parr F_2$. In the first case if X is in F_1, then $\{x, y\} \upharpoonright F_1$ is strictly incoherent and therefore, by definition of $\Gamma(F_1 \otimes F_2)$, $\{x, y\} \upharpoonright F$ is. The other cases are dealt with on an equal footing.

Finally, we observe that the $\{x, y\}$ from the main proof satisfies the required hypotheses, hence $\{x, y\}$ is strictly incoherent. As $\mathcal{R}$ being a clique entails that $\hat{\mathcal{R}}$ is a lax-clique, this implies that $\mathcal{R}$ is not a clique. This is a contradiction, and the two elements $y, x \in \mathcal{R}$ were equivalent. □

Thus, the hypercoherence condition enables us to avoid bad relations, as presented in section 3.4.4. We now turn to study if they satisfy the condition $(P1')$ of MALL^- proof structures 2.2.3.2, namely that for each &-resolution, the relation defines a unique ⊕-resolution on it. We say that an element $x \in [\![\Delta]\!]$ is on a additive resolution of Ψ of Δ if it is in the image of the natural embedding $[\![\Delta \upharpoonright \Psi]\!] \to [\![\Delta]\!]$.

Proposition 3.34. *Given Δ a sequent, $\mathcal{R}$ a clique of $[\![\Delta]\!]$, Ψ a &-additive resolution of Δ, and $x, y \in \mathcal{R}$ such that x, y on Ψ. Then x, y are on the same additive-resolution.*

Proof. The proof is done in a similar fashion as above. We take two lists x, y that are on the same &-resolution, but on a different ⊕-resolution. We then use substitutions to equalise all names in the list. Then there must some some sub-formulas $F_1 \oplus F_2$ of F such that the x explores F_1 whereas y is on F_2. Hence on these sub-formulas, the two lists are strictly lax-incoherent. Thus, using the same reasoning as above, they are lax-incoherent. This is a contradiction, and therefore on each &-resolution, the relation can define only one additive-resolution. □

However, nothing prevents the clique from being empty on a &-resolution. So in order for a clique $\mathcal{R} \subseteq [\![\Delta]\!]$ to form a valid proof-structure, we have to add the condition that for any &-resolution Ψ of Δ, there exists $x \in \mathcal{R}$, such that x is on that Ψ. This condition will be explored in more details in the last chapter 7.3.2, where we shall notably prove that it composes.

Even with this additive property, the clique might still fail to form a proper proof structure as the property $(P2)$ is not automatically verified. We recall that $(P2)$ imposes that on each additive resolution, the linking defines a MLL-proof net. However, the hypercoherence condition fails at the MLL-level, allowing the mix-rule. For instance, the sequent $\vdash A, A^\perp, B, B^\perp$ with its unique possible linking, has a valid encoding as a clique, despite being not provable.

As proven in [87] [83], hypercoherence is not strong enough for a completeness result for MALL. However, the denotation of atomic formulas can be chosen such as a full completeness result holds for MLL^-+ mix. Furthermore, if the proof structure is (P2), then the hypercoherence model can be strong enough to enforce full completeness, although with a different modelling of atomic types than the one presented above. This will be further explored in the last chapter of this thesis.

Part II

Nominal Asynchronous Games

Chapter 4

Nominal Structures for Asynchronous Games

As nominal refinement of traditional static nominal models proved to be not fully complete for linear logic, we take another direction, using the aforementioned link between tensorial and linear logic as means to achieve full completeness. The goal of this section is to present nominal structures that extend the current ones already established for games for tensorial logic, by translating them within a nominal universe. Ultimately, this will allow us to project morphisms onto nominal relations, and hence, denotations of proofs of linear logic.

Dialogue games were introduced in [69], as the objects supporting the game semantics of tensorial logic, without proposional variables. We devise their nominal sibling, providing us with the appropriate arenas, that let the strategies play with names. Therefore, we can enforce them to capture the required linearity between negative and atomic type variables, and hence extend the already established semantics of tensorial logic to include atomic variables.

We take advantage of this shift to deepen the relation between syntax, that is, terms, and semantics. It was established in [22], that the linear head reductions of lambda-terms in normal form produce look-alike strategies, that correspond to the denotations of the terms as strategies in game semantics. This correspondence was later refined, establishing an analogy between the "tree of views" of an innocent strategy, and the Böhm tree of the lambda-term. As we rely on nominal constructions, we give a nominal structure to the tensorial lambda-calculus, and characterise precisely alpha-equivalence. This permits the definition of nominal Böhm trees. We relate those Böhm trees with the sub-graphs of our nominal arenas, that will correspond to strategies, that we define in the next chapter 5. Therefore, although it was not the primary goal of this work, our framework allows us to draw an almost perfect correspondence between our strategies, and the set of nominal paths that arise as traces from the α-equivalence classes of lambda terms. This correspondence fails primarily due to the greater symmetry our structures enjoy compared to the lambda calculus, where the intermediate resources generated by player are not taken into account.

To sum up, the material presented here extends previous work on asynchronous game semantics [69, 64, 65, 66] by enriching arenas with names, providing appropriate structures for a clean semantics of atomic variables, while strengthening its relation with syntax. We start by characterising graphs that form nominal trees 4.2. Equipping the tensorial lambda calculus with names, we redefine Böhm trees as nominal graphs 4.3. Next, we introduce nominal dialogue games, alongside exposing the denotation function from formulas of tensorial logic onto them 4.4. Dialogue games are the backbone behind arenas, however we still need to unravel some nominal structure between the two. We set to define nominal event structures together with their relation with nominal di-domains. We present the event structure associated with a dialogue game, and characterise its set of positions 4.5. At this stage, we make a pause, and expose how one can project maximal positions onto lists 4.5.6; projections that will later allow us to project strategies onto nominal relations. Finally, we expose how one can see the set of positions as a polarised nominal asynchronous graph 4.6. At last, the Böhm trees lead as well to asynchronous graphs, that form sub-graphs of their respective arenas 4.7.

In the sequel, we add a new infinite enumerable set $\mathbb{A}_{\mathsf{cells}}$ of names, that will accommodate the untyped cells. These will be a key element to describe our games, and notably the additive units. The set $\mathbb{A}$ now becomes $\mathbb{A} = (\biguplus_{X \in \mathsf{TVar}} \mathbb{A}_X) \uplus \mathbb{A}_{\mathsf{cells}}$, and we write $\mathbb{A}_T$ for $\biguplus_{X \in \mathsf{Tvar}} \mathbb{A}_X$, the set of typed names. Furthermore, we set $\mathsf{Perm}(\mathbb{A}) = \mathsf{Perm}(\mathbb{A}_T) \oplus \mathsf{Perm}(\mathbb{A}_{\mathsf{cells}})$. We will refer to the elements of $\mathbb{A}_{\mathsf{cells}}$ as untyped names, or cell-names. Finally, we write $\nu_T(x)$ for $\nu(x) \cap \mathbb{A}_T$ and $\nu_{\mathsf{cells}}(x)$ for $\nu(x) \cap \mathbb{A}_{\mathsf{cells}}$. Similarly, we will sometimes use notations $\simeq_T$, (respectively $\simeq_{\mathsf{cells}}$) meaning that there are permutations of $\mathbb{A}_T$ (respectively $\mathbb{A}_{\mathsf{cells}}$) equalising the two elements.

4.1 Fraenkel-Mostowski sets

Nominal sets suffer that they do not allow us to consider elements with non-empty support as first-class objects. That is, given an element a of a nominal set, if a has non-empty support then the set $\{a\}$ does not form a nominal set. To make up for this, we introduce Fraenkel-Mostowski set theory, that provides a model of set theory encompassing nominal sets. Notably, as it is closed under $\in$ precedence, each element of a Fraenkel-Mostowski set forms a set of the Fraenkel-Mostowski model, abbreviated FM in the future.

The FM model of set theory is a model of set theory with atoms, that form the building blocks of the model. The atoms are primitive elements: no set can belong to an atom. More precisely, the FM model of set theory $\mathcal{V}_{FM}$ is built according to a cumulative hierarchy following similar steps to those leading to the Von Neumann model of set theory. However, the starting point $\mathcal{V}_0$ for FM consists of the set of names $\mathbb{A}$ instead of the empty-set $\emptyset$ for the Von Neumann model. At each iteration n of the construction, the action of nominal permutations on $\mathcal{V}_n$ is well defined by $\in$-recursion. Notably, all related notions, such as support, are well-defined for elements of $\mathcal{V}_n$. The inductive step consists in:

$$\mathcal{V}_{n+1} = \{S \subseteq \mathcal{V}_n \mid S \text{ has finite support}\}$$

The resulting model $\mathcal{V}_{FM}$ leads to sets that have finite support, and such that each element of the set has finite support. Working within this model allows us to consider functions $\mathcal{V}_{FM} \to \mathcal{V}_{FM}$, and reason about them. For instance, we will make use of the following proposition.

Proposition 4.1 ([32])**.** *Let F an equivariant function. Then $F(u) = v \Rightarrow \nu(F(v)) \subseteq \nu(F(u))$*

Looking at general constructions on sets (such as $\cup, \cap, ...$) as functions on $\mathcal{V}_{FM}$, this allows us to deduce inequalities like $\nu(A \cup B) \subseteq \nu(A) \cup \nu(B)$ for instance.

As we will have some use of trees with non-empty support, we provide here some additional terminology. Given $A \subseteq_{\mathsf{fin}} \mathbb{A}$, we say that an object T of FM is **A-nominal** if its support belongs in A: $\nu(T) \subseteq A$. For instance, an A-nominal function between two A-nominal sets $S \to T$ is a function such that $\nu(f) \subseteq A$. Given an A-nominal object T, properties in T hold up to A-equivalence, that is, up to equivalence for permutations π such that $\pi\#A$. For instance, an A-nominal function f satisfies: $\forall \pi\#A. f(\pi \cdot x) = \pi \cdot f(x)$. A nominal set is an $\emptyset$-nominal set. Finally, given an element x, or a subset S of an A-nominal set T, we write $[x]_A$, $[S]_A$ for their **A-orbits**, $[x]_A = \{\pi \cdot x \mid \pi\#A\} \subseteq T$ and $[S]_A = \{\pi \cdot x \mid x \in S, \pi\#A\} \subseteq T$. Accordingly, we say that two elements x, y are **A-equivalent**, written $x \simeq_A y$, if $[x]_A = [y]_A$.

Definition 4.2. *A-nominal sets and A-nominal functions form a category, called the **category of A-nominal sets**, written NSet_A.*

We gather in the following proposition some relevant properties of A-nominal sets.

Proposition 4.3.

- *Given $A \subseteq B \subseteq_{\mathsf{fin}} \mathbb{A}$, there is a faithful functor $F : \mathsf{NSet}_A \to \mathsf{NSet}_B$.*
- *Given any set S of $\mathcal{V}_{FM}$ with finite support, S is a $\nu(S)$-nominal set.*
- *For all $S \subseteq T$, then $[S]_{\nu(T)} \subseteq T$.*

The functor F is simply the functor that sends an object and a morphism to itself. The proof of the proposition is straightforward. This new framework allows us to produce the following definition.

Definition 4.4. *An **A-partially ordered set** $(S, \leq)$ is an A-nominal set together with a partial order $\leq$ such that $\nu(\leq) \subseteq A$. That is, $\forall v_1, v_2 \in S, \forall \pi\#A$,*

$$v_1 \leq v_2 \Leftrightarrow \pi \cdot v_1 \leq \pi \cdot v_2.$$

For more on this, we refer to [88].

4.2 Nominal trees

Dialogue games form a special class of trees, or, more precisely, directed rooted trees. Therefore, to be able to cope with nominal denotations of atomic variables within trees, one needs to define nominal trees. This is done by mimicking the set theoretic definitions that characterise them as graphs within nominal set theory. This section is divided into three parts. First, we present the definitions. Then, we explore the properties of the trees we obtained. Finally, we present a way to quickly produce and denote nominal trees.

4.2.1 Definitions

For our setting, "finite" trees will be appropriate. We say that a A-nominal set T is **orbit finite** if there is a finite (non-nominal) subset $S \subseteq T$, such that S is finite, and $T = [S]_A$.

Proposition 4.5. *Let T be an A-nominal set which is orbit finite. Let B such that $A \subseteq B \subseteq_{\mathsf{fin}} \mathbb{A}$. Then T is an orbit finite B-nominal set.*

This corresponds to the theorem 3.3 of [17], where the proof can be found. All the sets used along this section will be considered to be orbit finite. We rely our approach on the following notion of tree, similar to the one used within proof nets.

Definition 4.6. *Within the category of sets,a **directed rooted tree** (abbreviated directed tree, or simply tree in the future) is a directed graph in which there exists a node u, called the root, such that for any vertex v, there is exactly one path from u to v.*

To start, we define A-nominal graphs. Let us remind that within the category of sets, an oriented graph is a diagram $E \to V \times V$. As the first step of our shift, we consider the exact same diagram within the category of A-nominal sets. We furthermore assume the function to be injective. This is a quite strong property, expressing, in different terms, that there is at most one edge $\mathsf{e} : v_1 \to v_2$. If we were to relax this condition, the graphs obtained would actually be multigraphs.

Definition 4.7. *An **A-nominal graph** (V, E, f) consists of two orbit finite A-nominal sets V and E, called **vertices** and **edges**, and an A-nominal injective function $f : E \to V \times V$ (written $v_1 \xrightarrow{\mathsf{e}} v_2$ for $f(\mathsf{e}) = (v_1, v_2)$), that is, such that for all permutations $\pi \# A$, $v_1 \xrightarrow{\mathsf{e}} v_2 \Leftrightarrow \pi \cdot v_1 \xrightarrow{\pi\cdot\mathsf{e}} \pi \cdot v_2$. The graph is **directed** if we assume a direction of the edge from v_1 to v_2, written $v_1 \xrightarrow{\mathsf{e}} v_2$. Otherwise, it is **undirected**.*

We will sometimes refer to the elements of V as **nodes**. With this definition, an A-nominal graph is naturally oriented, and there is a natural forgetful operation that transforms it into an undirected graph. We will refer to it as the underlying undirected graph. When forgetting about the orientation, we will write $v_1 \overset{e}{\longleftrightarrow} v_2$ to denote an edge between v_1 and v_2.

Given an A-nominal graph, we recall the definition of **walk**, as being a finite sequence of triples $((v_1, \mathsf{e}_1, v'_1), (v_2, \mathsf{e}_2, v'_2)......, (v_n, \mathsf{e}_n, v'_n))$ such that $v'_i = v_{i+1}$ and $v_i \overset{\mathsf{e}_i}{\longleftrightarrow} v'_i$. A **path** is a walk in which all vertices v_i and all edges are distinct. In a directed A-nominal graph, a walk (respectively path) is **directed** if we only allow triples (v, e, v') where $v \xrightarrow{\mathsf{e}} v'$ in it. Given a directed walk s, we write $u \overset{s}{\twoheadrightarrow} v$, or $s : u \twoheadrightarrow v$, if $v_1 = u$ and $v'_n = v$, or $u \overset{s}{\twoheadleftrightarrow} v$ (respectively $s : u \twoheadleftrightarrow v$) in the case where the walk is undirected. The first simple observation to make is that the sets of walks, and paths, are A-nominal in the following sense: if $u \overset{s}{\twoheadleftrightarrow} v$ then for any nominal permutation π such that $\pi \# A$ we have $\pi \cdot u \overset{\pi \cdot s}{\twoheadleftrightarrow} \pi \cdot v$. Given two walks $s : u \twoheadleftrightarrow v$ and $s' : v \twoheadleftrightarrow w$, one can compose them to form a new walk $s.s' : u \twoheadleftrightarrow w$ by concatenating them. However, the concatenation of paths does not necessarily lead to a path. A **subsequence** of a sequence s is a sequence t such that there exist two sequences s, s' satisfying $s = s'.t.s''$. Similarly, given a sequence s, a **pre-sequence**, or **prefix** of s is a sequence t such that there exists u, $s = t.u$. This leads straightforwardly to notions of **pre-path**, **pre-walk** and **subpath**, **subwalk**. A **cycle** is a non-empty path whose first and last vertices are equal. A graph is **acyclic** if no paths in it are cycles. It is **connected** if for every pair of vertices u, v there is a path whose ending points are precisely u, v.

Definition 4.8. *A directed A-nominal graph is an* ***A-nominal tree*** *if it is a directed rooted tree.*

4.2.2 Properties of nominal trees

We investigate and present some properties of A-nominal graphs and trees. We notably relate A-nominal trees and A-partially ordered sets.

Proposition 4.9. *Let $T = (V, E, f)$ be an A-nominal graph (respectively tree). Let B such that $A \subseteq B \subseteq_{\mathsf{fin}} \mathbb{A}$. Then, relying on the functor $F : \mathsf{NSet}_A \to \mathsf{NSet}_B$, (that is, the trivial inclusion functor) $F(T) = (F(V), F(E), F(f))$ is a B-nominal graph (respectively tree).*

We present some properties relating the names of V and those of E.

Lemma 4.10. *Given an edge $\mathsf{e} : u \to v$ of an A-nominal graph , then $(\nu(\mathsf{e}) \setminus A) = (\nu(u) \cup \nu(v)) \setminus A$.*

Proof. We start by proving the left to right inclusion. Let us suppose there exists $a \in (\nu(\mathsf{e}) \setminus (\nu(u) \cup \nu(v))) \setminus A$, and let us consider a fresh b, $b \# \mathsf{e}, u, v, a, A$. We then have $(a, b) \cdot \mathsf{e} = \mathsf{e}$, but $(a, b) \cdot u = u$ and $(a, b) \cdot v \neq v$. As the graph's map $f : E \to V \times V$ is injective, $f((a, b) \cdot \mathsf{e}) \neq f(\mathsf{e})$. On the other hand, by nominality $f((a, b) \cdot \mathsf{e}) = ((a, b) \cdot u, (a, b) \cdot v) = (u, v) = f(\mathsf{e})$. Hence, we reach a contradiction and $(\nu(\mathsf{e}) \setminus A) \subseteq (\nu(u) \cup \nu(v)) \setminus A$. The same reasoning works to prove the reverse inclusion. □

Proposition 4.11. *A-nominal trees are* ***conservative****. For any pair of nodes u, v such that there exists an edge $\mathsf{e} : u \to v$, then $\nu(u) \setminus A \subseteq \nu(v) \setminus A$. In particular, in a nominal tree $\nu(u) \subseteq \nu(v)$.*

Proof. Suppose that there is an $a \in (\nu(u) \setminus \nu(v)) \setminus A$, and pick a b fresh : $b \# u, v, A$. Then, as the graph is A-nominal, there is an edge $(a, b) \cdot \mathsf{e} : (a, b) \cdot u \to (a, b) \cdot v = v$. Therefore v has two predecessors $(a, b) \cdot u$ and u. Now, taking a path from the root r to u, and one from r' to $(a, b) \cdot u$, we obtain two paths from r to v, contradicting the definition of the graph being a directed tree. □

We present another definition of directed tree within set theory. A subset S of a partially ordered set $(T, \leq)$ is well-ordered if $\leq \cap (S \times S)$ is a total order on S. A tree could be defined as a partially ordered set $(V, \leq)$ such that, for each element $v \in V$, its down-closure $v\downarrow = \{w \mid w \leq v\}$ is well-ordered, together with the existence of a unique least element. This definition forgets about any labelling E may bring, and hence is not equivalent to the first definition, though strongly related. In this case, E is the subset of $V \times V$ embodying the successor relation $\vdash$ coming from the partial order $\leq$. This definition does translate smoothly in our case. We recall that the **successor relation** is defined as the relation such that $v \vdash w \Leftrightarrow (v \leq w \wedge (\forall x. v \leq x \leq w \Rightarrow (v = x \vee w = x)))$.

Proposition 4.12.

- *Let T be an A-nominal tree. Then T gives rise to an A-partially ordered set $(S, \leq)$ such that the down-closure of each element is well-ordered.*
- *Let $(S, \leq)$ be an A-partially ordered set such that the down-closure of each element is well-ordered. Then this one provides a description of an A-nominal tree.*

Proof. We rely on the correspondence established within set theory. let T be an A-nominal tree. Forgetting about its nominal structure, it leads to a partially ordered set such that the down-closure of each element is well-ordered, that is, the restriction of the partial order to this subset is a total order. By definition, it set V is A-nominal, and as the partial order relation is coming from (E, f), which are A-nominal objects, it is A-nominal. Therefore, it gives rise to an A-nominal partially ordered set as expected. The reverse direction is proved on an equal footing. □

The successor relation $\vdash$ on V is characterised by $u \vdash v$ if there exists an $\mathsf{e} \in E$ such that $u \xrightarrow{\mathsf{e}} v$. In that case, we say that u **justifies** v. Furthermore, we say that an element is **initial** if it is justified by the root. Finally, we notice that the proposition 4.11 entails that any vertex in a nominal tree must remember the history of names leading to it. That is, if there is a name in the support of an element $w \leq v$, then it must be in the support of v as well. Formally, for any vertex $v \in V$, we have $\nu(v\downarrow) \setminus A = \nu(v) \setminus A$.

4.2.3 Structured nominal trees

In this section, we give our trees additional structure that help us handle them easily. More precisely, as any node remembers the names of the nodes appearing in its downward closure, we present nodes as lists, that reflect on their downward closures.

Definition 4.13. *A **structured A-nominal tree** (V, E, f) is an A-nominal tree such that:*

- *the vertices are finite lists of elements.*
- *the root is the empty list.*
- *if there is an edge $\mathsf{e} : v_1 \to v_2$, then $v_2 = v_1.k$, where k is a list of length 1, and $\mathsf{e} = (v_1, v_2)$.*

Note that a structured tree is perfectly defined by its set V. Hence in the sequel, we will forget about E. Furthermore, in a structured tree, the lengths of the lists correspond actually to the distances between the nodes and the root. In that case, to each vertex v, the well-ordered set $v\downarrow$ of vertices that appear before it is simply the set of pre-sequences of v. For simplicity, we write $\ulcorner v \urcorner$ for the last element of the list. Therefore, $\ulcorner v \urcorner = v$ only if v is the root or is initial. Similarly, we write $\ulcorner \mathsf{e} \urcorner$ for k, called its view. Let us note that an edge in a structured tree is perfectly specified by its starting node and its view. Given an A-nominal tree, there is a canonical structured A-nominal tree associated, by replacing each node v with $v\downarrow$ structured as a list, and by replacing E with the appropriate set of $V \times V$

We present below a convenient way to construct structured nominal trees. We start with a way to produce directed trees within set theory.

Proposition 4.14. *Let T be a set of finite lists. We define its **closure** $\mathsf{Clos}(T)$ by:*

$$\mathsf{Clos}(T) = \{l \mid \exists l_2 \in T, \exists l_1,\ l.l_1 = l_2\}$$

and equipping it with its natural partial order $l_1 \leq l_2$ if l_1 is a prefix of l_2. Then $(\mathsf{Clos}(T), \vdash)$ is a directed tree, where $\vdash$ is the successor relation associated with $\leq$.

This follows straightforwardly from the characterisation of directed tree as poset having a single minimal element, here the empty-list, and well-ordered downward closure for each of their elements. This proposition provides a simple way to check if a set of lists T produces a directed tree, one simply needs to check that $T = \mathsf{Clos}(T)$.

Proposition 4.15. *Given a set of lists T such that $\mathsf{Clos}(T) = T$, $v(T) = A$, then T is an A-nominal tree.*

Proof. We set $V = T$, E the required subset of $V \times V$, and f being the injection function $E \to V \times V$. Writing A for $v(T)$, we need to prove that E is A-nominal. This follows by noticing that given $(v_1, v_2) \in V \times V$, such as $v_2 = v_1.k$, and $\pi \# A$ then $\pi \cdot (v_1, v_2) \in V \times V$ by definition of V being A-nominal. Therefore, E is A-nominal, and the natural injection function is A-nominal also. Therefore, (V, E, f) is an A-nominal graph, forming an directed tree. □

The next theorem presents some constructions that will be useful to build structured nominal trees. To start, we introduce the following definition, that allows us to see sets as lists. Given n A_i nominal sets S_i, we define $S_1.S_2.\ldots.S_n$ to be the set of lists $l = l_1.\ldots.l_n$ such that $l_i \in S_i$. Finally, given a structured tree T, we say that an element l of T is maximal if it is maximal with relation to the partial order. We write $\mathsf{Max}(T)$ for the set of such lists.

Definition 4.16. *Given T_1, T_2 two structured A_1, A_2-nominal trees respectively, we define $T_1.T_2$ as the following set of lists:*

$$T_1.T_2 = T_1 \cup \{l_1.l_2 \mid l_1 \in \mathsf{Max}(T_1), l_2 \in T_2\}$$

*Given n A_i nominal trees T_i, we define their **rooted sum** $\bigoplus_{i\in[1,n]} T_i$, to be the set of lists $\bigcup_{i\in[1,n]} \mathsf{in}_i(T_i)$ where $\mathsf{in}_i(T_i) = \{\epsilon\} \uplus \{(\mathsf{in}_i e).l \mid e.l \in T\}$, and e is a list of length* 1.

Theorem 4.17. *In the following, we will consider T_i to be structured, orbit-finite, A_i-nominal trees.*

- $T_1.T_2 \dots .T_n$ *is a structured* $(\bigcup_{i\in[1,n]} A_i)$*-nominal tree.*
- $\bigoplus_{i\in[1,n]} T_i$ *is a structured* $\bigcup_{i\in[1,n]} A_i$*-nominal tree.*
- $\bigcup_i T_i = \{l \mid \exists i \in [0,n], l \in T_i\}$ *is a structured* $\bigcup_{i\in[0,n]} A_i$*-nominal tree.*
- *Given n* A_i *nominal sets* S_i*,* $\mathsf{Clos}(S_1.S_2.\dots S_n)$ *is a* $\bigcup_{i\in[1,n]} A_i$*-nominal tree. For instance, given an element a, then seeing a as a list of length* 1*,* $\mathsf{Clos}\{a\}$ *is a* $\nu(a)$*-nominal tree.*

Proof. The proofs of all these statements rely on proposition 4.15, following a similar reasoning. Therefore, we only present the proof for the first. We start by noticing that $\mathsf{Clos}(T_1.T_2.\dots.T_n) = T_1.T_2.\dots.T_n$, and, by proposition 4.1, one gets that $\nu(T_1.T_2.\dots.T_n) \subseteq \bigcup_{i\in[1,n]} \nu(T_i)$. That allows us to conclude the proof. □

Given a vertex v, we write v^i for the i^{th} element of v, subject that $i \leq \mathsf{length}(v)$.

Definition 4.18. *We define the relation of* ***compatibility for equality****, written* C *.*

- *Given two elements* x, y *of a nominal set without additional structure, we say that* x, y *are compatible for equality, written* $x\ C\ y$ *if* $x \simeq y \Rightarrow x = y$
- *Given two vertices of a structured nominal tree* v, w*, we say that* v, w *are compatible for equality, written* $v\ C\ w$*, if* $\forall i \leq \mathsf{min}(\mathsf{length}(v), \mathsf{length}(w))$*,* $(v^i\ C\ v^i)$*.*

There are a few interesting properties about the relation C that will prove useful in the future. We present them below.

- if $v\ C\ v'$ and $v \simeq v'$ then $v = v'$.
- if $v\ C\ v'$ and $w \leq v, w' \leq v'$ then $w\ C\ w'$.
- if $\neg(u\ C\ v)$ and $(v \leq w)$ then $\neg(u\ C\ w)$.

The last property merely states that C^c (the complement of C in $V \times V$) somehow behaves like a conflict relation in an event structure. This will be relevant for the next section 4.5.

Proposition 4.19. *If* $v\ C\ w$ *then one can speak about* $v \cap w = v^1.v^2 \dots v^i$ *such that* $i = \mathsf{max}\{j \leq \mathsf{min}(\mathsf{length}(v), \mathsf{length}(w)) \mid v^j = w^j\}$*.*

The proof is straightforward. In the case where $\neg(v\ C\ w)$, we will say, despite the fact that the two elements might have a greatest lower bound, that the intersection is ill-defined. We do so to avoid the intersection to be too dependent on the names chosen.

4.3 Böhm trees

Böhm trees are among the most famous nominal trees in the literature, although they have, to the best of the author's knowledge, never been presented from a nominal point of view. Accordingly, the definition is presented for terms, not for their classes of α-equivalence.

Given a game semantics for a pure functional programming language (that is, without effects), there is a strong relationship between the object denoting a type, and the Böhm trees of the terms realising this type. This has been the subject of the investigation of [64, 22, 74, 21], and we will give more details once the definition of arena for tensorial logic types is settled.

4.3.1 Nominal tensorial calculus

We enrich the terms of the lambda tensorial calculus with a nominal flavour, to make them inline with our framework. That is, we change the structure of terms in a suitable manner to relate them to the nominal structures previously defined. In the sequel $x, y, z \in \mathbb{A} \oplus \{\bullet\}$.

$$\begin{array}{rr} \text{Types} & \mathrm{TY} \ni T, U := X \mid I \mid \neg T \mid T \otimes U \\ \text{Terms} & \mathrm{TE} \ni t, u := x \mid \bullet \mid tu \mid \neg_x.t \mid t \otimes u \mid \mathsf{let}\ z\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ t \\ \text{Typing context} & \Gamma := \emptyset \mid x : T, \Gamma \\ & \text{where } X \in \mathsf{TVar} \end{array}$$

Γ is a typing context, that is, a set of variables together with a type. We write $x : T, \Gamma$ for $\{x : T\} \cup \Gamma$, with x such that $x \# \Gamma$. That is, the set of variables is separated. The typing rules of our terms are as follows, where we write $\bot$ for the type $\neg I$.

$$\frac{}{\vdash \bullet : I} \qquad \frac{\Gamma \vdash t : T}{\Gamma, \bullet : I \vdash t : T} \qquad \frac{}{x : X \vdash x : X}\ x \in \mathbb{A}_X$$

$$\frac{\Gamma, x : U \vdash t : \bot}{\Gamma \vdash \neg_x.t : \neg U} \qquad \frac{\Gamma \vdash t : T}{\Gamma, f : \neg T \vdash ft : \bot}\ f \in \mathbb{A}_{\mathsf{cells}}, f \# \Gamma.$$

$$\frac{\Gamma \vdash t : T \qquad \Delta \vdash u : U}{\Gamma, \Delta \vdash t \otimes u : T \otimes U}\ \Gamma, t \# \Delta, u \qquad \frac{\Gamma, x : T, y : U \vdash t : V}{\Gamma, z : T \otimes U \vdash \mathsf{let}\ z\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ t : V}\ \mathbb{A}_{\mathsf{cells}} \ni z \# \Gamma.$$

We only present the fragment without cut, since cut-free proofs yields terms in normal form, in the sense that no β-reduction steps can be applied to them. Note that we imposed $t \# u$ in the right tensor rule to avoid term like $(\ \mathsf{let}\ u\ \mathsf{be}\ (x \otimes y)\ \mathsf{in}\ x \otimes y) \otimes (\ \mathsf{let}\ v\ \mathsf{be}\ (x \otimes y)\ \mathsf{in}\ x \otimes y)$ which would be, otherwise, perfectly valid.

We will simplify the terms and the typing system a bit, transforming it into an almost equivalent one. This transformation is akin to the transformation of the original sequent calculus of multiplicative tensorial logic to the focalised one. This will allow us to having terms with more structure, and comes at a the price of some mild assumptions and restrictions on terms:

- Forgetting about terms whose open variables are of type $T \otimes U$. That is, the variables of our open terms are either atomic, or of negation type. One can easily transform a term that got an open variable of type $U \otimes V$ into an somehow equivalent one having two open variables of types U, V.

- Considering that the $\otimes$ operation on types and terms is strictly associative: $T \otimes (U \otimes V) = (T \otimes U) \otimes V$, and similarly for terms.

The terms and types now have the following forms:

$$\begin{array}{rl}
\text{Types} & \mathrm{TY} \ni T, U := X \mid I \mid \neg T \mid \bigotimes_i T_i \\
\text{Terms} & \mathrm{TE} \ni t, u := x \mid \bullet \mid \bigotimes_i t_i \mid tu \mid \neg(x_1, ... x_n).t; \\
\text{Typing context} & \Gamma := \emptyset \mid x : T, \Gamma \\
& \text{where } X \in \mathsf{TVar}
\end{array}$$

where we assume that, when writing $\bigotimes t_i$, none of the t_i were already tensor terms.

The typing rules for terms are as follows:

$$\frac{}{\vdash \bullet : I} \qquad \frac{\Gamma \vdash t : T}{\Gamma, \bullet : I \vdash t : T} \qquad \frac{}{x : X \vdash x : X}\ x \in \mathbb{A}_X$$

$$\frac{\Gamma, x_1 : U_1, ..., x_n : U_n \vdash t : \bot}{\Gamma \vdash \neg(x_1, ..., x_n).t : \neg(\bigotimes_{i \in [1,n]} U_i)} \qquad \frac{\Gamma \vdash t : T}{\Gamma, f : \neg T \vdash ft : \bot}\ f \in \mathbb{A}_{\mathsf{cells}}, f \# \Gamma.$$

$$\frac{\Gamma_1 \vdash t_1 : T_1 \quad ... \quad \Gamma_n \vdash t_n : T_n}{\Gamma_1, ..., \Gamma_n \vdash \bigotimes_{i \in [1,n]} t_i : \bigotimes_{i \in [1,n]} T_i}\ \forall i \neq j \in [1,n]. \Gamma_i, t_i \# \Gamma_j, t_j$$

We call terms coming from this typing system **focalised terms**.

We now define an equivalence relation on terms of the tensorial lambda-calcululs.

Definition 4.20. *We define the σ-equivalence between terms of the tensorial lambda-calculus of the same type as the smallest reflexive transitive relation such that:*

$$\begin{array}{rl}
\mathsf{let}\ z\ \mathsf{be}\ (x \otimes y)\ \mathsf{in}\ \mathsf{let}\ w = (u \otimes v)\ \mathsf{in}\ t \sim_\sigma & \mathsf{let}\ w = (u \otimes v)\ \mathsf{in}\ \mathsf{let}\ z\ \mathsf{be}\ (x \otimes y)\ \mathsf{in}\ t \\
(\ \mathsf{let}\ z\ \mathsf{be}\ (x \otimes y)\ \mathsf{in}\ t) \otimes v \sim_\sigma & \mathsf{let}\ z\ \mathsf{be}\ (x \otimes y)\ \mathsf{in}\ (t \otimes v) \\
t \otimes (\ \mathsf{let}\ z\ \mathsf{be}\ (x \otimes y)\ \mathsf{in}\ v) \sim_\sigma & \mathsf{let}\ z\ \mathsf{be}\ (x \otimes y)\ \mathsf{in}\ (t \otimes v) \\
f(\ \mathsf{let}\ z\ \mathsf{be}\ (x \otimes y)\ \mathsf{in}\ t) \sim_\sigma & \mathsf{let}\ z\ \mathsf{be}\ (x \otimes y)\ \mathsf{in}\ f(t) \\
\mathsf{let}\ z\ \mathsf{be}\ (x \otimes y)\ \mathsf{in}\ \neg_w.t \sim_\sigma & \neg_w.\ \mathsf{let}\ z\ \mathsf{be}\ (x \otimes y)\ \mathsf{in}\ t \\
t \sim_\sigma t' \Rightarrow & C[t] \sim_\sigma C[t']
\end{array}$$

where C is any context coming from a formula of tensorial logic. Furthermore, we define the *following rewriting system:*

$$\text{let } z \text{ be } (x \otimes y) \text{ in } \text{ let } x \text{ be } (u \otimes v) \text{ in } t \to \text{ let } z \text{ be } (u \otimes v \otimes y) \text{ in } t$$

$$\text{let } z \text{ be } (x \otimes y) \text{ in } \text{ let } y \text{ be } (u \otimes v) \text{ in } t \to \text{ let } z \text{ be } (x \otimes y \otimes v) \text{ in } t$$

$$\neg(v_1, ..., x, ..., v_n). \text{ let } x \text{ be } (u_1 \otimes ... \otimes u_n) \text{ in } t \to \neg(v_1, ..., u_1 \otimes ... \otimes u_n, ..., v_n).t$$

$$t \otimes (u \otimes v) \to t \otimes u \otimes v$$

$$(t \otimes u) \otimes v \to t \otimes u \otimes v$$

$$t \to t' \Rightarrow C[t] \to C[t']$$

We denote by $\rightsquigarrow$ *the reflexive transitive closure of the relation* $\to$.

Proposition 4.21. *Every well-typed term t whose open variables are not of tensor types is* σ*-equivalent to a well-typed term* t' *such that* $t' \rightsquigarrow t''$ *and* t'' *is a focalised term of the same type as t. Furthermore* t'' *is unique satisfying this property.*

The proof is similar to the proof that every proof of MLL is equivalent to a proof of the focalised fragment of MLL, and hence will not be reproduced here. Indeed, the σ equivalence rules are precisely the rules coming from the permutations of rules in proofs of tensorial logic. Finally, the rewriting system giving rise to $\rightsquigarrow$ encapsulates precisely the transformation to a strictly associative terminology.

We define bound and open variables for terms of focalised tensorial logic.

$\mathsf{bv}(x) = \emptyset$	$\mathsf{fv}(x) = x$
$\mathsf{bv}(\bullet) = \emptyset$	$\mathsf{fv}(\bullet) = \emptyset$
$\mathsf{bv}(tu) = \mathsf{bv}(t) \cup \mathsf{bv}(u)$	$\mathsf{fv}(tu) = \mathsf{fv}(t) \cup \mathsf{fv}(u)$
$\mathsf{bv}((\neg(x_1, ..., x_n).t) = \{x_1, ..., x_n\} \cup \mathsf{bv}(t)$	$\mathsf{fv}(\neg(x_1, .., x_n).t) = \mathsf{fv}(t) \setminus \{x_1, .., x_n\}$
$\mathsf{bv}(t \otimes u) = \mathsf{bv}(t) \cup \mathsf{bv}(u)$	$\mathsf{fv}(t \otimes u) = \mathsf{fv}(t) \cup \mathsf{fv}(u)$

As our terms are based on names, one can define the action of a permutation on them.

- $\pi \cdot x = \pi \cdot x$ (that is, the action on the term x is the action on name x).
- $\pi \cdot \bullet = \bullet$.
- $\pi \cdot (tu) = (\pi \cdot t)(\pi \cdot u)$
- $\pi \cdot (\neg(x_1, ..., x_n).u) = \neg(\pi \cdot x_1, ..., \pi \cdot x_n).(\pi \cdot u)$
- $\pi \cdot (t \otimes u) = (\pi \cdot t) \otimes (\pi \cdot u)$

Writing (x, y) for the smallest permutation that switches x, y, the α-equivalence on terms is defined as the smallest relation $=_\alpha$ on terms such that:

- $x =_\alpha x$ if $x \in \mathbb{A} \uplus \{\bullet\}$.
- $tu =_\alpha t'u'$ if $t =_\alpha t'$ and $u =_\alpha u'$.
- $t \otimes u =_\alpha t' \otimes u'$ if $t =_\alpha t'$ and $u =_\alpha u'$.
- $\neg(x_1, ..., x_n).t =_\alpha \neg(x'_1, ..., x'_n).t'$ if:
  $\text{И}(y_1, ..., y_n)\#t, t',\ (x_1, y_1)(x_2, y_2)...(x_n, y_n) \cdot t =_\alpha (x'_1, y_1)(x'_2, y_2)....(x'_n, y_n) \cdot t'$.

We write $[t]_\alpha$ for the equivalence class of t under α-equivalence. Formally:

$$[t]_\alpha = \{t' \mid t =_\alpha t\}$$

One can notice that $t =_\alpha t' \Rightarrow \mathsf{fv}(t) = \mathsf{fv}(t')$. Furthermore, the support of $[t]_\alpha$ is precisely $\mathsf{fv}(t)$, hence it has finite support.

4.3.2 Böhm trees

A Böhm tree is a presentation of the α-equivalence class of a focalised λ-term in normal form, under the shape of a tree. We present in the figure below the general shape of a Böhm tree, and it should be clear from context how one can assign to an (α-equivalence class of a) term its associated Böhm tree. Though Böhm trees are naturally typed, they are more general than terms: to some Böhm trees might correspond no well-typed terms. Furthermore, in a Böhm tree two additional constants are added. For each type T we introduce a constant of type Ω_T, and a special constant $\mho_\perp$ of type $\perp$. Ω_T reflects that we stop the exploration of the tree at this stage. It has to be seen like a decision made by the environment, declaring that it does not wish to explore the branch of the tree any further. On the other hand $\mho_\perp$ reflects a failure from the tree to provide a sub-tree of the appropriate type. From the game semantics point of view it expresses the fact that term/strategy cannot answer a query from the opponent. These intuitions will be made clearer later, in section 4.7.

A Böhm tree M on a simple type T of the form $\neg(T_1 \otimes ... \otimes T_m)$ is of the following shape:

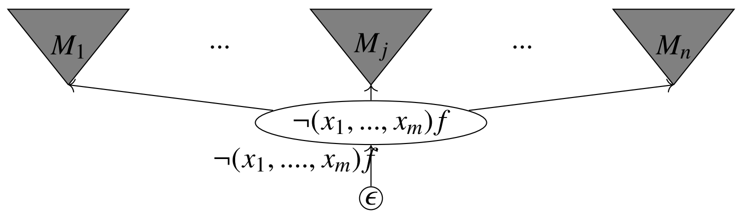


Figure 4.1: Structure of a Böhm tree of type $\neg(T_1 \otimes ... \otimes T_m)$

It corresponds to an α-equivalence class of terms of the shape $\neg(x_1, ..., x_m).f(t_1 \otimes ... \otimes t_n))$, where t_i are the terms corresponding to the sub-trees M_i, themselves seen as Böhm trees. However, the Böhm trees M_i now have additional information: the bound names $(x_1, ..., x_n)$ introduced by the first edge, together with their types. Therefore, we introduce a finite typing context Γ, that represents the names that have been disclosed to the tree. They correspond to the free variables of a term.

We say that a type T is **$\otimes$-irreducible** if it cannot be decomposed into $T = T_1 \otimes T_2$. A $\otimes$-irreducible type is either an atomic type or a negated type: $T = \neg U$. Therefore, the types of the focalised tensorial calculus are of three sorts: negated types, atomic types, or types of the shape $T_1 \otimes ... \otimes T_n$ (with n > 1). Finally, we impose that in the typing context Γ, the names of

$\mathbb{A}_{\mathsf{cells}}$ have types negated types only, whereas the names of $\mathbb{A}_T$ have types their corresponding atomic types.

Definition 4.22. *For Γ a typing context, writing S for $\nu(\Gamma)$, we define a **Γ-Böhm tree** M on the type T to be a is a structured S-nominal tree, whose structure is depending on the structure of the type.*

If T is a negated type, $T = \neg(T_1 \otimes ... \otimes T_m)$, where each T_i is $\otimes$-irreducible, then a Γ-Böhm tree M is in one of the following three forms:

1. *A tree $\mathsf{Clos}([\neg(x_1, ..., x_m)f]_S.M_{x_1,...,x_n})$ such that:*
   - *If T_i is not an atomic type, then $x_i \in \mathbb{A}_{\mathsf{cells}}$.*
   - *if T_i is an atomic type X then $x_i \in \mathbb{A}_X$.*
   - *if T_i is the atomic type I, then $x_i = \bullet$.*
   - *$\forall i, j. i \neq j \Rightarrow x_i \# x_j$ and $(x_1, ..., x_m) \# S$.*
   - *Defining $\Gamma' = \Gamma \cup (x_1 : T_1, ..., x_m : T_m)$, $f \in \nu(\Gamma') \cap \mathbb{A}_{\mathsf{cells}}$ and therefore is of type $U = \neg(V)$.*
   - *$M_{x_1,...,x_m}$ is a Γ'- Böhm tree of type V.*
   - *Given $(y_1, ..., y_m) \in [(x_1, ..., x_m)]_S$, and π permutation of minimal support such that $\pi \cdot (x_1, ..., x_m) = (y_1, ..., y_m)$ then $M_{y_1,...,y_m} = \pi \cdot M_{x_1,...,x_n}$.*
2. *$\mathsf{Clos}([\neg(x_1, ..., x_m)\mho_\bot]_S)$ where the x_i have same constraints as above.*
3. *$\mathsf{Clos}(\{\Omega_T\})$*

If T is an atomic type then a Γ-Böhm tree of type T is in one of the following forms:

- *$\mathsf{Clos}(\{\Omega_T\})$.*
- *$\mathsf{Clos}(\{\bullet\})$ if $T = I$.*
- *$\mathsf{Clos}([x]_S)$ for a given $x \in \mathbb{A}_X$ if $T = X \in \mathsf{TVar}$.*

And finally we define a Γ-Böhm tree of type $T_1 \otimes ... \otimes T_n$ to be the a rooted sum of Γ-Böhm trees $M_1, ...M_n$ of types $T_1, ..., T_n$ respectively.

Given an α-equivalence class of terms $[t]_\alpha$, one can assign its associated Γ- Böhm tree, where Γ is its context set of free variables. Therefore, the Böhm tree associated with a term is a structured $\mathsf{fv}(t)$-nominal tree, and therefore, it is a structured nominal tree if and only if the term is closed. For instance, the structured Böhm tree associated with the term x of type X consists of one root and one initial element x. On the other hand, the Böhm tree associated with the term $\neg(f, x).fx$, of type $\neg(\neg X \otimes X)$ is a nominal tree of empty support that consists of the closure of a set of lists of length 2 $\mathsf{Clos}(\{\neg(f, x)f.x \mid f \in \mathbb{A}_{\mathsf{cells}}, x \in \mathbb{A}_X\})$.

We give some more examples below. The Böhm tree associated to the term $\neg(x, w, f, g, h).h(x \otimes \neg u.(g(w \otimes \neg v.(f(u \otimes v))))))$, of type $\neg(X \otimes Y \otimes (\neg(Z \otimes W) \otimes (\neg(Y \otimes \neg W)) \otimes (\neg(X \otimes \neg Z)))$ is presented in the figure 4.2 below. Another interesting Böhm tree is the one of the term presented in figure 4.3, whose associated Böhm tree is displayed in the next figure 4.4. While displaying these structured trees, we denote each edge e by its view $\ulcorner \mathsf{e} \urcorner$.

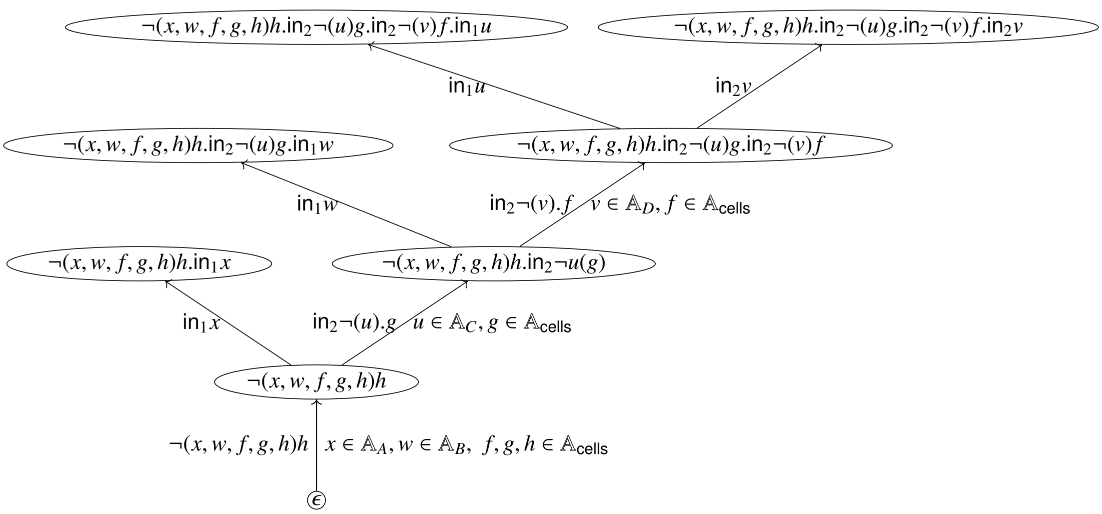


Figure 4.2: Böhm tree associated with $[\neg(x, w, f, g, h).h(x \otimes \neg u.(g(w \otimes \neg v.(f(u \otimes v))))))]_\alpha$

$$\dfrac{\dfrac{\dfrac{\dfrac{u : X \vdash u : X}{u : X, f : \neg X \vdash f(u) : \bot}}{v : X \otimes \neg X \vdash \ \mathsf{let}\ v\ \mathsf{be}\ (u \otimes f)\ \mathsf{in}\ f(u) : \bot}}{\vdash \neg(u, f).f(u) : \neg(X \otimes \neg X)} \qquad \dfrac{\dfrac{\dfrac{w : Y \vdash w : Y}{w : Y, g : \neg Y \vdash g(w) : \bot}}{x : Y \otimes \neg Y \vdash \ \mathsf{let}\ x\ \mathsf{be}\ (w \otimes g)\ \mathsf{in}\ g(w) : \bot}}{\vdash \neg(w \otimes g).g(w) : \neg(Y \otimes \neg Y)}}{\vdash \neg(u, f).f(u) \otimes \neg(w \otimes g).g(w) : \neg(X \otimes \neg X) \otimes \neg(Y \otimes \neg Y)}$$

Figure 4.3: A term of the tensorial lambda calculus

4.4 Nominal dialogue games

We now have all the background material necessary to begin to present the objects that will be the interpretation of our formulas of tensorial logic. We approximately follow the presentation given in [69], while carefully adapting it with names. We start by presenting the tree-interpretation of formulas, though these trees will not be the final interpretation. In the following section 4.6, we will equip them with additional structure, transforming them into asynchronous graphs. Those will be precisely the arenas on which the strategies, defined in the next chapter 5, will play.

We remind that in a structured tree, each vertex is a list l, and we write $\ulcorner l \urcorner$ for the last element of the list. Furthermore, we remind that a pattern is a sequence highlighting the position of an element.

$$\mathsf{pattern} ::= \mathsf{inl}\,\mathsf{pattern} \mid \mathsf{inr}\,\mathsf{pattern} \mid \epsilon$$

Finally, given a structured nominal tree A, we write $\overline{A}$ for its set of lists that are non empty.

Definition 4.23. *A **dialogue game** is a structured $\emptyset$-nominal tree A whose vertices, except for the root, are either labelled **cells** or **values** (that is, with a function* $\mathsf{label} : \overline{A} \to \{\mathsf{cells}, \mathsf{values}\}$*), and such that* $x \vdash y \Rightarrow \mathsf{label}(x) \neq \mathsf{label}(y)$*. Each cell* c *is such that* $\ulcorner \mathsf{c} \urcorner$ *is of the shape* $\mathsf{pattern}(\alpha)$ *(often written α for short), where $\alpha \in \mathbb{A}$. Furthermore, this graph is equipped with a function*

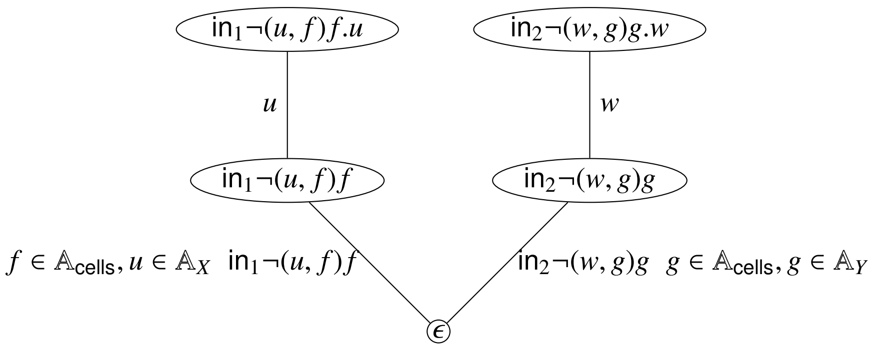


Figure 4.4: Böhm tree associated with $[\neg(u, f).f(u) \otimes \neg(w \otimes g).g(w)]_\alpha$

$\lambda : \overline{A} \to \{-1, 1\}$, *called polarity, such that for each cell* α *and value* v *:*

$$\alpha \vdash v \Rightarrow \lambda(\alpha) = \lambda(v) \qquad v \vdash \alpha \Rightarrow \lambda(\alpha) \neq \lambda(v)$$

We request that the initial vertices are cells that consists of a unique name members of $\mathbb{A}_{\mathsf{cells}}$*, that is, all initial cells are equivalent. A dialogue game is said to be* ***positive*** *if all the initial cells are of positive polarity, and* ***negative*** *if all the initial cells are of negative polarity.*

The values and cells of positive polarity are those belonging to proponent, often called player, whereas those of negative polarity are related to opponent. In this graph, each cell represents a question, that is, a request of data. On the other hand, the values represent the answer to the question, that is, they correspond to new data. For historical reasons, the polarity of the cell does not correspond to the protagonist asking the question, but the protagonist answering it. That is, proponent will asks questions, or bring cells, of polarity -1, but will answer questions of polarity 1. We write that a cell is typed if the support of its last element belongs in $\mathbb{A}_T$, and untyped if it belongs in $\mathbb{A}_{\mathsf{cells}}$. If the name of a cell belongs in $\mathbb{A}_X$, we say that the cell is of type X.

In our games, the only values we will encounter are products of patterned $\bullet$, as the only ground type we work with is I. In the case where one would like to implement a programming language based on the tensorial lambda calculus extended with Booleans or integers, for instance, then the values would range over these two sets. Furthermore, if one woud like to implement a nominal language, such as the ν-calculus, then the values could be nominal. Similarly, if one were to adapt them to cope with references, or other kind of resources, that are encoded nominally (just as in [76] [77] for instance), then, in that case, our values would not necessarily have empty support anymore.

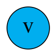
4.4.1 Interpretation of formulas

The interpretation of formulas as dialogue games has been introduced in [69], but we slightly modify it here. Indeed, in the original presentation, the dialogue games were forests, rooted at values, whereas here we forget the root and focus on the initial elements, that are cells. The initial cells always have polarity +1 by definition. Therefore the polarity function is fully defined for the whole tree from the two equations:

$$\alpha \vdash v \Rightarrow \lambda(\alpha) = \lambda(v) \qquad v \vdash \alpha \Rightarrow \lambda(\alpha) \neq \lambda(v).$$

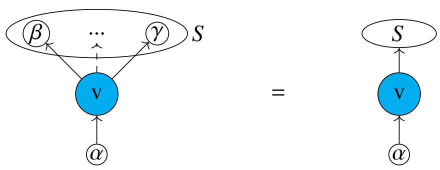
Thus, in the future, we only present the set of nodes, as the polarity of each node can simply be computed from its distance from the root. We request that all dialogue games that are denotation of formulas have positive polarity. Negative dialogue games will be later used to denote the arena on which the strategies play.

In the sequel, by abuse of notation and terminology, we will not make the distinction between the formula and its denotation as a dialogue game. For instance we write "dialogue games A, B" instead of "dialogue games that are denotations of formulas A, B". There is a nice graphical display for dialogue games tha was introduced in [69] that we recall below. In order to draw the graphs, we only display one node for each equivalence class of nodes . Furthermore, we do not display the root. The values are represented as filled circles :

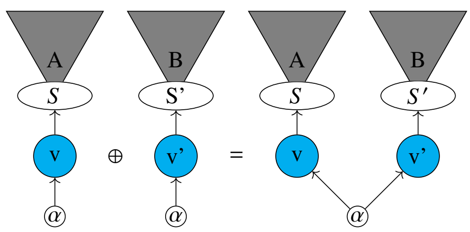


Whereas the cells are drawn as smaller, plain circles.

By definition, the initial elements of the dialogue games are always cells of positive polarity. For a value justifying multiple cells, we adopt the following drawing convention:



The interpretation of formulas is given below. First, we define the sum of two dialogue games A and B by merging their initial cell. This is a slight variant of the rooted sum.

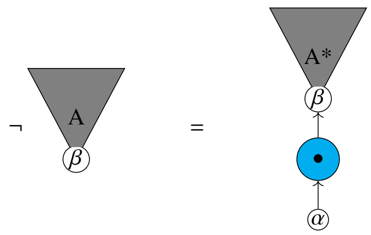


Formally, given two dialogue games A, B , we define $A \oplus B$ by:

$$\mathsf{Clos}(\{\alpha.\mathsf{inl}(v).l \mid \alpha.v.l \in A\}) \cup \mathsf{Clos}(\{\alpha.\mathsf{inr}(v).l \mid \alpha.v.l \in B\})$$

A dialogue game is **simple** if each initial cell justifies a unique value. That is, if it cannot be written as a sum of two dialogue games.

The second step consists in defining the negation. First, given a dialogue game A, we define A^* as being just like A but with a reverse polarity function, $\lambda_{A^*} = -\lambda_A$. A^* is not a dialogue game, as its initial elements have negative polarity. We solve this by lifting it, that is, adding a couple (α, v) such that v justifies the initial cells of A^*, and α is an untyped cell.

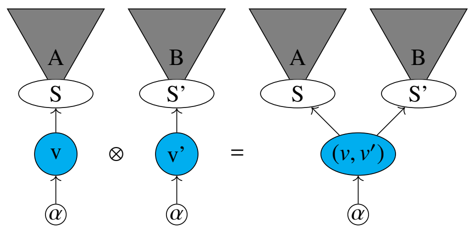


Formally, we set:

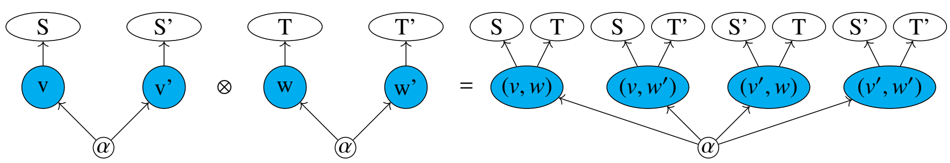
$$\neg(A) = \mathsf{Clos}(\mathbb{A}_{\mathsf{cells}}.\{\bullet\}).A$$

where we recall that the concatenation of structured trees is defined in 4.16.

The tensor product of two simple games A and B is defined by merging their initial cells and their initial values. The associated picture is displayed below;



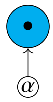
Relying on the the distributivity of $\oplus$ over $\otimes$, we generalise the definition for any dialogue game. For instance, we display the product in the case of two dialogues games having two initial values.



So formally, for two games A, B, the list definition of the tree $A \otimes B$ is:

$$A \otimes B = \mathsf{Clos}(\{\alpha.(v_1, v_2).\mathsf{inl}(\beta).l. \mid \alpha.v_1.\beta.l \in A \wedge \alpha.v_2 \in B\})$$
$$\cup\, \mathsf{Clos}(\{\alpha.(v_1, v_2).\mathsf{inr}(\beta).l \mid \alpha.v_2.\beta.l \in B \wedge \alpha.v_1 \in A\})$$

We now give the interpretation of the units. The game 1 consists of a unique value, justified by a set of cells of $\mathbb{A}_{\mathsf{cells}}$.



Properly, we write :

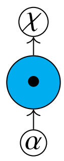
$$[\![I]\!] = \mathsf{Clos}(\mathbb{A}_{\mathsf{cells}}.\{\bullet\})$$

The game for 0 is a 1-orbit set of equivalent untyped cells.

That is :

$$[\![0]\!] = \mathsf{Clos}(\mathbb{A}_{\mathsf{cells}})$$

The remaining units $\top$ and $\bot$ are defined through the equations $\top = \neg 0$ and $\bot = \neg 1$. Finally, we give the interpretation of an atomic type X. We want it to be a typed resource brought by proponent. Therefore, we define the denotation of X to be the nominal graph displayed below, where $\chi \in \mathbb{A}_X$.



In terms of lists, this gives :

$$[\![X]\!] = \mathsf{Clos}(\mathbb{A}_{\mathsf{cells}}.\{\bullet\}.\mathbb{A}_X)$$

Finally, we extend our denotation to any formula of tensorial logic through $[\![A \otimes B]\!] = [\![A]\!] \otimes [\![B]\!]$, $[\![\neg A]\!] = \neg[\![A]\!]$ and $[\![A \oplus B]\!] = [\![A]\!] \oplus [\![B]\!]$. Let us note here that the fact that the typed names are brought in cells, and not values, is a choice that we made. We could certainly have designed similar games where typed resources are values and obtained similar results. However, as each atomic variable can be interpreted as a hole, that can be filled with any other formulas, it made sense to put it into a cell. Moreover, this incorporation allows us to deal with them through the sequentiality structure, defined in the course of the next chapter 5.3.2.

4.4.1.1 On negative dialogue games

In the next chapter, we will consider, following [69], that strategies $A \to B$ are sets of plays in $A \triangleright B = (A \otimes \neg B)^*$, where the $(.)^*$ denotes the function that inverses polarities. Given that

A, B will be positive, $A \triangleright B$ shall be a negative game. To refer to such structure, we name it pre-dialogue game. We draw below how it looks like in the case where A, B are simple.

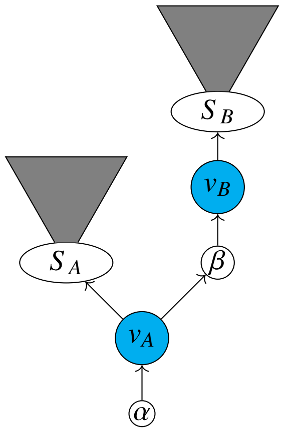
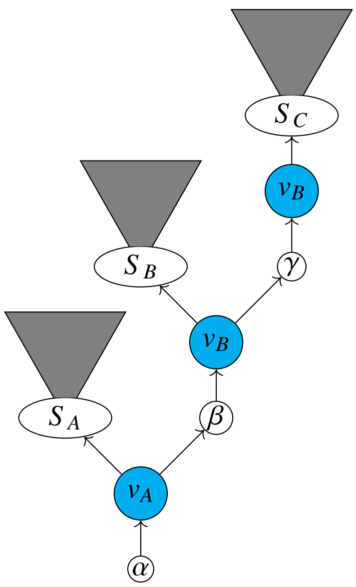


Figure 4.5: Pre-dialogue game $A \triangleright B$

In order to deal with composition, we have to speak about global sequences in a structure $A \triangleright B \triangleright C$, such that it has locally the structure of $A \triangleright B$ on the left-hand side and $B \triangleright C$ on the right-hand side. More precisely, we define the structure $A \triangleright B \triangleright C$ to be the dialogue game $A \otimes \neg(B \otimes \neg C)$ except that we do not endow it with a polarity function.

The shape of the dialogue game $A \triangleright B \triangleright C$ is displayed in the picture below 4.6, whenever A, B, C are simple.

Figure 4.6: Structure of $A \triangleright B \triangleright C$



4.4.1.2 Distributivity, normal form and dialogue games

Recalling that a dialogue game is simple if each initial cell justifies only one value, every dialogue game A can be decomposed as a finite sum of simple games: $A = \bigoplus_i A_i$. Furthermore, any simple game can be written $A = \bigotimes_i A_i$ where each A_i is simple and $\otimes$-irreducible, that is, cannot be written $A_i = B_1 \otimes B_2$. We characterise the $\otimes$-irreducibility of simple games by the fact that the unique initial value justifies a set of cells such that all cells are equivalent. In other terms, the set of cells justified by the initial value forms a 1-orbit set. Therefore, each game can be decomposed as $\bigoplus_i(\bigotimes_{j|i} A_{i,j})$, where we write $j \mid i$ to indicate that the set j ranges upon depends on i.

Furthermore, if we were to forget some of the bureaucracy given by the patterns, there would be a bijection between the positive dialogue games such that every typed cell is terminal (that is, forms a maximal vertex of the structured tree), and the formulas of tensorial logic, up to associativity-distributivity equivalence, as defined in section 2.3.2.1. More precisely, to each positive dialogue game such that typed cells are terminals, looking only at its structure (that is, forgetting about the additional data given with the ordering of patterns), a formula of tensorial logic in normal form can be canonically associated, and reversely. Additionally, given a normal form formula of tensorial logic, and its associated dialogue game, to each untyped cell of the dialogue game corresponds a sub-formula of the initial formula, and reciprocally. Hence, the dialogue game of a certain type can be seen as a variant of the syntactical tree of the formula of this type.

4.5 Nominal event structures and linear di-domains

In this section, we present the definition of nominal event structures, and expose their correspondence with nominal di-domains. This will enable us to associate to each dialogue game an event structure, through the intermediate description of moves. Thanks to that correspondence, we will be able to conclude that the set of positions associated with a dialogue game forms a nominal di-domain. We start by laying the definition of nominal event structure.

Definition 4.24. *A* ***nominal event structure*** $E = (|E|, \leq, \smile)$ *consists of:*

- *A nominal set* $|E|$.
- *A nominal partial order relation* $\leq$.
- *A symmetric irreflexive nominal conflict relation* $\smile$ *such that* $m \smile n$ *and* $n \leq p \Rightarrow m \smile p$.

A nominal event structure is ***linear*** *if* $\forall e, e' \in |E|.(e \simeq e' \wedge e \neq e') \Rightarrow e \smile e'$.

We restrict to orbit finite event structures, that are those whose nominal sets $|E|$ are orbit finite. We write $\uparrow$ for the complement of the conflict relation: $e \uparrow e' \Leftrightarrow \neg(e \smile e')$. If $e \uparrow e'$, we say that e, e' are **compatible**. In particular, if $e \leq e'$ then $e \uparrow e'$. Finally, we say that two events are **independent** if they are compatible and not related by the partial order: $e \uparrow e' \wedge \neg(e \leq$

$e') \wedge \neg(e' \leq e)$. To simplify, we write $e \in E$ for $e \in |E|$. For the orbit finite linear event structures, the axiom of finite causes holds automatically:

$$\forall e \in E.\{e' \mid e' \leq e\} \text{ is finite.}$$

The linearity hypothesis states that two equivalent events are in conflict, that is, they cannot happen within the same configuration. This assumption has major consequences on the design of the event structures.

Proposition 4.25. *Let E be a linear event structure, and $e, e' \in E$ such that $e \leq e'$. Then $\nu(e) \subseteq \nu(e')$.*

Proof. Suppose $\exists a \in \mathbb{A}$ such that $a \in \nu(e) \setminus \nu(e')$, and consider also b such that $b \# e, e'$. As the relation $\leq$ is nominal, $(a,b) \cdot e \leq (a,b) \cdot e'$. On the other hand $(a,b) \cdot e' = e'$ and $(a,b) \cdot e \neq e$. From linearity, we know that $e \smile (a,b) \cdot e$. Therefore, as $e \smile (a,b) \cdot e \leq e'$, by the axiom of event structures, $e \smile e'$. Finally, as $e \leq e'$, this leads to $e \uparrow e'$, contradicting $e \smile e'$. To sum up, $\nu(e) \subseteq \nu(e')$. □

Therefore, the nominal trees behave in a compatible manner with regards to linear nominal event structures. Indeed, each nominal tree can be seen as a finite orbit nominal partial ordered set. Furthermore, since they are conservative, they can be extended with a notion of conflict that transforms them to linear event structures. That is, the relation $\smile$ defined by $v_1 \smile v_2$ if $\neg(v_1 \; C \; v_2)$ behaves like a conflict relation in a nominal event structure.

We introduce the notion of positions. We recall that in a partially ordered set P, given $D \subseteq P$, we write $D\downarrow$ for the **downward closure** of D, defined as being the following subset:

$$D\downarrow = \{p \in P \mid \exists d \in D, p \leq d\}.$$

Definition 4.26. *A **position** p of a nominal event structure is a finitely supported set of events such that:*

- *p is conflict-free: $\forall e, e' \in p. e \uparrow e'$.*
- *p is downward closed: $p = p\downarrow$*

As a result, no two events in a position are in conflict. We write $\mathsf{Pos}(A)$ for the set of positions of A. We define the actions of nominal permutations on $\mathsf{Pos}(A)$ as follows: $\pi \cdot p = \{\pi \cdot e \mid e \in p\}$. In an orbit finite linear event structure, the positions have a useful representation.

Proposition 4.27.

- *Let S be a finitely supported subset of a nominal partially ordered set E. Then $\nu(S\downarrow) \subseteq \nu(S)$.*
- *Let p be a position in an orbit finite linear event structure E. Then $\exists e_1, ..., e_n \in E$ such that $p = \{e_1, ..., e_n\}\downarrow$.*

Proof. Let $\pi \# S$. Then for any $e \in (S\downarrow)$, $\pi \cdot e \in (\pi \cdot S)\downarrow = S\downarrow$. Reversely, for any $e \in \pi \cdot (S\downarrow)$, $e \in S\downarrow$. So $\pi \cdot (S\downarrow) = (S\downarrow)$, and therefore $\nu(S\downarrow) \subseteq \nu(S)$.

We tackle the second bullet-point. We consider the set of elements that are maximal in p; $\mathsf{Max}(p) = \{e \in p \mid \nexists e' \in p, e \leq e'\}$. As E is orbit finite, every chain of growing elements of p is bounded by one of these elements, and $p = \mathsf{Max}(p)\downarrow$. Furthermore, as E is orbit finite and linear, $\mathsf{Max}(p)$ is finite. Indeed, if it were infinite, there would be some equivalent elements, and these would be in conflict, contradicting p being a position. Therefore, $\exists e_1, ..., e_n \in E$ such that $p = \{e_1, ..., e_n\} \downarrow$. □

On the other hand, the reverse inclusion $\nu(S) \subseteq \nu(S\downarrow)$ does not generally hold. Let E be defined as follows:

$$E = (\{\bot\} \uplus \mathsf{inl}(\mathbb{A}) \uplus \mathsf{inr}(\mathbb{A}), \bot < \mathsf{inl}(\mathbb{A}) < \mathsf{inr}(\mathbb{A}), \smile = \emptyset\}$$

Then, we have $\{\mathsf{inl}(a), \mathsf{inr}(b)\}\downarrow = \mathsf{inl}(\mathbb{A}) \uplus \{\bot, \mathsf{inr}(b)\}$. Therefore, $\nu(\{\mathsf{inl}(a), \mathsf{inr}(b)\}) = \{a, b\}$, but $\nu(\{\mathsf{inl}(a), \mathsf{inr}(b)\}\downarrow) = \{b\}$.

We introduce some notions that will be relevant for the sequel. In the set of positions, we say that two positions p_1, p_2 are **compatible**, written $p_1 \uparrow p_2$, if the set $\{p_1, p_2\}$ is bounded. We will write $\bot_E$ (or simply $\bot$) for the empty position, and $\mathsf{Pos}^*(E)$ for the set of non-empty positions, that is $\mathsf{Pos}^*(E) = \mathsf{Pos}(E) \setminus \{\bot_E\}$.

In ordinary set theory the set of positions of an event structure forms a di-domain. We remind below some basic notions of domain theory, translated into nominal sets.

Definition 4.28. *In this definition, all structures are assumed to be orbit finite.*

- *A* ***nominal domain*** $(\mathbb{D}, \sqsubseteq_{\mathbb{D}}, \bot_{\mathbb{D}})$ *is a nominal poset* $(\mathbb{D}, \sqsubseteq_{\mathbb{D}})$ *together with a least element* $\bot_{\mathbb{D}}$*. Abusing notation, we refer to* $\mathbb{D}$ *for the whole structure.*
- *A nominal domain* $\mathbb{D}$ *is* ***linear*** *if for all* x, y *in* $\mathbb{D}$ *such that* $x \neq y$ *and* $x \simeq y$*, then* x, y *have no common upper bound.*
- *A nominal domain* $\mathbb{D}$ *is* ***bounded complete*** *if every finitely supported bounded subset* X *of* $\mathbb{D}$ *has a least upper bound* $\bigsqcup X$*.*
- *A* ***prime*** *of a bounded complete domain* $\mathbb{D}$ *is an element* $\mathsf{p} \in \mathbb{D}$ *such that for any finitely supported subset* X *of* $\mathbb{D}$, $\mathsf{p} \sqsubseteq \bigsqcup X \Rightarrow \exists x \in X.\mathsf{p} \sqsubseteq x$*. We denote* $\mathrm{Pr}(\mathbb{D})$ *the set of primes of a domain.*
- *A nominal domain* $\mathbb{D}$ *is* ***prime algebraic*** *if it is bounded complete and* $\forall x \in \mathbb{D}.x = \bigsqcup\{\mathsf{p} \sqsubseteq x \mid \mathsf{p} \in \mathrm{Pr}(\mathbb{D})\}$*.*

We write **di-domain** for those nominal domains that are prime algebraic.

Proposition 4.29. *In a bounded complete nominal domain, every finitely supported* $X \subseteq \mathbb{D}$ *has a greatest lower bound, denoted* $\bigsqcap X$*.*

Proof. We simply need to ensure that the set $P = \{y \mid \forall x \in X.y \leq x\}$ has finite support. More precisely, we can prove in a similar manner as the proof of proposition 4.27 that $\nu(P) \subseteq \nu(X)$. Therefore, $\bigsqcup P$ is well-defined and is obviously the greatest lower bound of X. □

Just as for its non-nominal counterpart, there is an equivalence between nominal di-domains and event structures, that relies on a correspondence between the primes of the di-domains and the events in one direction, and the positions of the event structures and the elements of the domains for the other direction. Before doing the proof, we introduce this lemma.

Lemma 4.30. *In a di-domain $\mathbb{D}$, the following hold:*

- $\forall \mathsf{p} \in \Pr(\mathbb{D}), \forall \mathsf{p}' \in [\mathsf{p}].\mathsf{p}' \in \Pr(\mathbb{D})$.
- *If $\mathbb{D}$ is linear then $\forall x \in \mathbb{D}$, $\forall \mathsf{p}, \mathsf{p}' \in \Pr(\mathbb{D}).\mathsf{p}, \mathsf{p}' \sqsubseteq x \wedge \mathsf{p} \simeq \mathsf{p}' \Rightarrow \mathsf{p} = \mathsf{p}'$.*

Proof. Let p be a prime, and $\mathsf{p}' = \pi \cdot \mathsf{p}$, where π is any name permutation. Suppose $\mathsf{p}' \sqsubseteq \sqcup X$, where $X \subseteq \mathbb{D}$. Then $\pi^{-1} \cdot \mathsf{p}' \sqsubseteq \pi^{-1} \cdot (\bigsqcup X) = \bigsqcup(\pi^{-1} \cdot X)$. Now, as $\mathsf{p} = \pi^{-1} \cdot \mathsf{p}'$ is prime, there is a x such that $\mathsf{p} \sqsubseteq \pi^{-1} \cdot x$ and $\mathsf{p}' \sqsubseteq x$. So p' is prime. The second part of the proposition is obvious by linearity. □

Proposition 4.31.

- *Let $E = (|E|, \leq, \smile)$ be a nominal event structure. Then its set of positions ordered by inclusion form a di-domain. Furthermore, if the event structure is linear then so is the di-domain.*
- *Let $\mathbb{D} = (\mathbb{D}, \sqsubseteq_{\mathbb{D}}, \perp_{\mathbb{D}})$ be a nominal di-domain. Then its set of primes ordered by $\sqsubseteq$ forms a nominal event structure, by setting $\mathsf{p} \smile \mathsf{p}'$ if $\{\mathsf{p}, \mathsf{p}'\}$ has no upper bound. Furthermore, if the domain is linear then so is the event structure.*
- *The two transformations are inverse to each other.*

Proof. We start with an event structure, and endeavour to show that its set of positions forms a di-domain. We already know that $\mathsf{Pos}(A)$ is a nominal set. Inclusion is obviously a nominal partial order relation, and there is a least element, namely the empty position. Hence, the positions form a domain. As the event structure is orbit-finite, so is this domain. Let us suppose that there is a set X of positions such that X is bounded, and X has finite support. As X consists of a set of positions, and each of them is downward closed, we already know that the set $\bigsqcup X = \{e \in E \mid \exists x \in X, e \in x\}$ is such that $\bigsqcup X = (\bigsqcup X)\downarrow$, and therefore $\nu(\bigsqcup X) \subseteq \nu(X)$ is finite. Furthermore, as it is bounded, it entails, $\exists p.\forall x \in X.x \sqsubseteq p$. As a result, all events e of $\bigsqcup X$ are in p, and therefore are compatible. Therefore $\bigsqcup X$ is a position. As a result, the domain is bounded-complete. Finally, one can check that to each event e corresponds a prime $e\downarrow = \{e' \mid e' \leq e\}$. Then as $p = \bigcup\{e\downarrow \mid e \in p\}$, the domain is prime algebraic.

Now, we prove that if the event structure is linear, so is the di-domain. We consider two positions p, p' such that $p \simeq p'$ and $p \neq p'$. Then this entails that there are two elements $e, e' \in p, p'$ such that $e \simeq e' \wedge e \neq e'$. As a result, $e \smile e'$. Therefore, there is no position that contains both e, e', and therefore no position p'' such that $p, p' \leq p''$. Therefore, the domain is linear.

We now focus on the reverse direction. By the previous lemma, we know that the set of primes organises itself as a nominal set. As the set of primes forms a subset of the domain, the nominal partial order relation $\sqsubseteq$ restricts straightforwardly to a nominal partial order relation on the set of primes. Two primes are incompatible if their union is not bounded, and therefore the axiom required for $\smile$ is clearly satisfied. In the case where the di-domain is linear, then every two equivalent but different primes are incompatible, and therefore so is the event structure. □

4.5.1 Structured event structures and operations

In this section, we refine the class of event structures we will be working with. This additional structure will allow us to define more easily some operations on them.

Definition 4.32. *A **tree event structure** is an event structure such that for each e, the position $e\downarrow$ is well-ordered.*

In other words, a tree event-structure is an event structure such that $(E, \leq)$ is a forest; a forest is just like a tree, except that there might be several minimal elements.

Lemma 4.33. *Every tree event structure is linear.*

Proof. Let e an event, and f, f' such that $f \simeq f'$ and $f, f' \leq e$. As $e\downarrow$ is well-ordered, if $f \neq f'$ this implies $f < f'$ or $f' < f$. Let π be such that $\pi \cdot f = f'$, then $\exists n \in \mathbb{N}. \pi^n = \pi$. As $\pi \cdot f < f$, $<$ being a nominal partial order relation and $<$ being transitive, this entails $\pi^n \cdot f < f$, that is, $f < f$. This is a contradiction, implying that $f = f'$. Hence the event structure is linear. □

Recycling terminology from game semantics, we say that an element of an event structure is **initial** if it is minimal with regards to the partial order. We write I_E for the set of initial events of E, and $\overline{I_E}$ for the set of non-initial ones. Formally, $\overline{I_E} = |E| \setminus I_E$. In analogy with trees, we defined structured tree event structures.

Definition 4.34. *A tree event structure E is **structured** if:*

- *Every element e of $|E|$ is a list.*
- *Given an element $e \in E$, then, if e is not initial, it has a unique predecessor e', and $e = e'.\ulcorner e \urcorner$.*
- *The initial elements are lists of length* 1.

That is, each element of the event structure is a list that reflects on its downward closure. In particular, in that case, E augmented with the empty list, acting as a bottom element, forms a structured tree. Note that the partial order $\leq$ of a structured tree comes from the prefix ordering. Therefore, when defining a structured event structure E, we only need to specify $|E|$ and $\smile$.

We present two operations on structured tree event structures. To do so, we introduce this notation: given three nominal sets A, B, C, together with two (implicit) nominal functions $\varpi_1 : A \to B$ and $\varpi_2 : B \to C$, we write $A \times_C B$ for the **fibred product** of A, B over C:

$$A \times_C B = \{(x, y) \in A \times B \mid \varpi_1(x) = \varpi_2(y)\}.$$

We will sometimes write $x \times_C y$ for the elements of $A \times_C B$. Given two structured event structures $E_1 = (|E_1|, \leq_1, \smile_1), E_2 = (|E_2|, \leq_2, \smile_2)$, together with a nominal set C, and two projections $\varpi_1 : I_{E_1} \to C$, $\varpi_2 : I_{E_2} \to C$, we define the following operations:

- $E_1 \oplus E_2 = (|E|, \leq, \smile)$, where :
  1. $|E| = \{\mathsf{inl}(e_1).e_2. \ldots .e_n \mid e_1.e_2. \ldots .e_n \in |E_1|\} \uplus \{\mathsf{inr}(e_1).e_2. \ldots .e_n \mid e_1.e_2. \ldots .e_n \in |E_2|\}$. We overload the notation $\mathsf{inl}, \mathsf{inr}$ for describing the two natural injections $|E_i| \to |E|$.
  2. $\smile = \mathsf{inl}(\smile_1) \uplus \mathsf{inr}(\smile_2) \uplus \mathsf{inl}(|E_1|) \times \mathsf{inr}(|E_2|) \uplus \mathsf{inr}(|E_2|) \times \mathsf{inl}(|E_1|)$.
- $E_1 \overline{\otimes}_C E_2 = (|E|, \leq, \smile)$:
  1. $I_E = I_{E_1} \times_C I_{E_2}$.
  2. $\overline{I_E} = \{(i_1, i_2).\mathsf{inl}(e_2). \ldots .e_n \mid i_1.e_2. \ldots .e_n \in |E_1|, (i_1, i_2) \in I_E)\} \uplus \{(i_1, i_2).\mathsf{inr}(e_2). \ldots .e_n \mid i_1.e_2. \ldots .e_n \in |E_2|, (i_1, i_2) \in I_E\}$
  3. $\leq$ is defined through the structure: $e_1 \leq e_2$ if e_1 is a prefix of e_2.
  4. For $e = (i_1, i_2).\mathsf{inl}(e_2). \ldots .e_n$, we write $e \upharpoonright E_1$ for the event $i_1.e_2. \ldots .e_n$, and $e \upharpoonright E_2 = i_2$. Similarly, we can define projections in the case $e = (i_1, i_2).\mathsf{inr}(e_2). \ldots .e_n$. Then, $e_1 \smile e_2$ if $(e_1 \upharpoonright E_1 \smile_1 e_2 \upharpoonright E_1) \vee (e_1 \upharpoonright E_2 \smile_2 e_2 \upharpoonright E_2)$

Proposition 4.35. *Given E_1, E_2 two structured tree event structures, then $E_1 \oplus E_2$ and $E_1 \otimes_C E_2$ are tree event structures.*

The proof is immediate. We endeavour to determine how these operations lift at the level of positions. For the first one, we have $\mathsf{Pos}^*(E_1 \oplus E_2) \simeq \mathsf{inl}(\mathsf{Pos}^*(E_1)) \uplus \mathsf{inr}(\mathsf{Pos}^*(E_2))$. We write $\overline{\uplus}$ for the adequate binary operation, defined below:

$$\mathsf{inl}(\mathsf{Pos}(E_1)) \,\overline{\uplus}\, \mathsf{inr}(\mathsf{Pos}(E_2)) = \{\bot\} \uplus \mathsf{inl}(\mathsf{Pos}^*(E_1)) \uplus \mathsf{inr}(\mathsf{Pos}^*(E_2))$$

And therefore, we conclude that:

$$\mathsf{Pos}(E_1 \oplus E_2) \simeq \mathsf{Pos}(E_1) \,\overline{\uplus}\, \mathsf{Pos}(E_2).$$

The set $\mathsf{Pos}(E_1 \overline{\otimes}_C E_2)$ is a bit harder to describe in the general case. However, for every event structure we will consider in the future, each position has a unique initial move. That is, for any $i_1, i_2 \in I_E$, $i_1 \smile i_2$. In that case, one can straightforwardly extend the functions ϖ_1, ϖ_2 from initial events to positions.

- $\varpi_1(p)$ is undefined if p is empty.
- $\varpi_1(p) = \varpi_1(i_1)$ if $p \neq \bot$ and i_1 is the unique initial event of p.

and similarly for ϖ_2. Therefore, we overload the operator $\overline{\otimes}_C$ to sets of positions:

$$\mathsf{Pos}(E_1) \,\overline{\otimes}_C \mathsf{Pos}(E_2) = (\bot_1, \bot_2) \uplus \mathsf{Pos}^*(E_1) \times_C \mathsf{Pos}^*(E_2)$$

And straightforwardly, in these cases, $\mathsf{Pos}(E_1 \overline{\otimes}_C E_2) \simeq \mathsf{Pos}(E_1) \,\overline{\otimes}_C \mathsf{Pos}(E_2)$.

In the case where we do consider a cartesian product instead of a fibred product, we will simply write $\overline{\otimes}$, without specifying an additional set.

4.5.2 Event structure of a dialogue game

Given a nominal set X, we call **representative** of X a minimal (not-nominal) subset $S \subset X$, such that $[S] = X$. Minimality entails that $\forall \beta_1, \beta_2 \in S.(\beta_1 \neq \beta_2) \Rightarrow (\beta_1 \not\approx \beta_2)$. An orbit finite nominal tree is automatically locally finite, in the sense that for every node v, the set $\mathsf{Succ}(v) = \{v' \mid v \vdash v'\}$ is orbit finite, where we remind that $\vdash$ denotes the successor relation.

We associate to each dialogue game of section 4.4.1 a structured tree event structure. Each event roughly corresponds to either a player or opponent move. The conflict relation is designed such that a position can only explore one side of the additive connective $\oplus$, while allowing the exploration of both sides of the multiplicative one $\otimes$. The event structure is built around the concept of moves: the event structure is a structured one, and each event is a list of moves. We first start by defining a look-alike event structure based on moves.

Definition 4.36. *Given a dialogue game A,* $\mathsf{M}_A = (\mathsf{M}_A, \leq_A, \smile_A)$ *is a nominal partially ordered set with a relation* $\smile_A$ *on* M_A*, called conflict.*

- *The set* M_A *consists of all triples* (α, v, S)*, where* α, v *are nodes of A and* $\alpha \vdash v$, $\mathsf{label}(\alpha) = \mathsf{cells}$. *S is a representative of* $\mathsf{Succ}(v)$*, and therefore is a set of cells. We will call* ***moves*** *the elements of* M_A*. Moves will be denoted by* m *and variants. Given a move* $\mathsf{m} = (\alpha, v, S)$ *we write* $\ulcorner \mathsf{m} \urcorner$ *for* $(\ulcorner \alpha \urcorner, \ulcorner v \urcorner, \ulcorner S \urcorner)$ *where* $\ulcorner S \urcorner = \{\ulcorner \mathsf{c} \urcorner \mid \mathsf{c} \in S\}$.
- *We define the* ***justifying relation*** *between moves by* $\mathsf{m} = (\alpha, v, S) \vdash_A (\alpha', v', S') = \mathsf{m}'$ *if* $\alpha' \in S$. *In this case we say that* m *justifies* m'*, written* $\mathsf{m} \vdash \mathsf{m}'$. *A move is initial if it is not justified by any move. We write* $\leq$ *for the reflexive transitive closure of* $\vdash$*, leading to a partially ordered set* $(\mathsf{M}_A, \leq)$.
- *The relation of* ***compatibility for equality****, written* C *is stronger than in the case of trees 4.18. We define it first for moves:*

$$(\alpha, v, S)\ C\ (\alpha', v', S') \text{ if } (v\ C\ v') \wedge (\forall \mathsf{c} \in S. \forall \mathsf{c}' \in S'. \mathsf{c}\ C\ \mathsf{c}')$$

- *The* ***conflict relation*** *between two moves is defined by* $\mathsf{m} \smile_A \mathsf{m}'$ *if* $((\mathsf{m}\ C\ \mathsf{m}') \Rightarrow (\mathsf{label}(v \cap v') = \mathsf{cells}))$*, where* $\mathsf{m} = (\alpha, v, S)$ *and* $\mathsf{m}' = (\alpha', v', S')$.

A move (α, v, S) answers the query α with the value v, and brings forth S as new questions. We will expose the intuitions behind this definition below. However, to start, it is important to notice that $(\mathsf{M}_A, \leq_A, \smile_A)$ is almost an event-structure, but not quite. To understand why, let us consider the dialogue game $\top \otimes \neg \top$, displayed below.

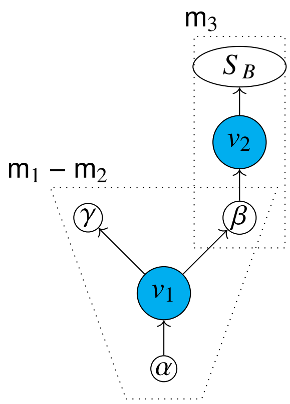


Figure 4.7: Dialogue game $\top \otimes \neg\top$

Let us pick two initial moves $\mathsf{m}_1 = (\alpha, v_1, \{\mathsf{in}_1(\mathsf{c}_1), \mathsf{in}_2(\mathsf{c}_2)\})$ and $\mathsf{m}_2 = (\alpha, v_1, \{\mathsf{in}_1(\mathsf{c}'_1), \mathsf{in}_2(\mathsf{c}_2)\}$ such that $\mathsf{c}_1 \neq \mathsf{c}'_1$. Finally, we consider a third move $\mathsf{m}_3 = \{(\mathsf{in}_2(\mathsf{c}_2), v_2, \{\mathsf{c}_3\})$. Then we have $\mathsf{m}_1 \smile \mathsf{m}_2$ as they are not compatible for equality since $\mathsf{in}_1(\mathsf{c}'_1) \neq \mathsf{in}_1(\mathsf{c}_1)$. On the other hand $\mathsf{m}_1 \vdash \mathsf{m}_3$ and $\mathsf{m}_2 \vdash \mathsf{m}_3$. So m_1 is compatible with m_3 but $\mathsf{m}_1 \smile \mathsf{m}_2$ and $\mathsf{m}_2 \leq \mathsf{m}_3$. So the conflict relation does not satisfy the necessary axiom to form an event structure.

So we turn this into an event structure by listing the moves that happen. This way, we obtain a tree event structure.

Definition 4.37. *Given a dialogue game A, and its associated set of moves* $(\mathsf{M}_A, \leq_A, \smile_A)$*, we define the event structure* $\mathsf{Event}(A) = (|E_A|, \leq_A, \smile_A)$ *by overloading the notations* $\leq_A$*,* $\smile_A$ *as follows:*

- *A nominal event* $e \in E_A$ *is a list* $\mathsf{m}_1.\mathsf{m}_2.\dots.\mathsf{m}_n$ *where* $\mathsf{m}_i \in \mathsf{M}_A$*,* $\mathsf{m}_1 \vdash \mathsf{m}_2 \vdash \dots \vdash \mathsf{m}_n$*, and* m_1 *is an initial move. We write* $\ulcorner e \urcorner$ *for* m_n*.*
- *Two events are compatible for equality if their moves are:*

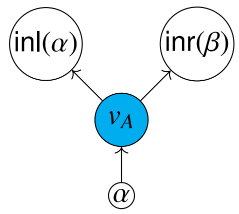
$$e_1 = \mathsf{m}_1.\dots.\mathsf{m}_n \; C \; e_2 = \mathsf{n}_1.\dots.\mathsf{n}_k \text{ if } \forall i \leq n, k. \mathsf{m}_i \; C \; \mathsf{n}_i.$$

- *Two events are in conflict* $e \smile e'$ *if* $(e \; C \; e') \Rightarrow \ulcorner e \urcorner \smile \ulcorner e' \urcorner$*.*
- *As expected,* $e \leq e'$ *if e is a prefix of* e'*.*

Note that, as expected, $e \simeq e' \wedge e \; C \; e' \Rightarrow e = e'$. Let us remind that as the dialogue game is a structured tree, every node encodes the nodes that happened before. Therefore, if $S, S' \neq \emptyset$. $S \; C \; S' \Rightarrow (v \; C \; v' \wedge \alpha \; C \; \alpha')$. However, it might be the case that $S, S' = \emptyset$, and that is why one needs to add $v \; C \; v'$ to define properly this relation for moves. Note that two initial moves that have not the same initial cells are in conflict, as they are not compatible for equality. Therefore, each position is "rooted" in one single cell.

The underlying idea behind the definition of $\smile_A$ is that only one value can fill a cell. For instance, when defining the interpretation $\oplus$ of two formulas, the two have same initial set of cells. Therefore, a position can only explore either the left or the right hand side of the $\oplus$, but not both, otherwise, in such a position, the initial cell would justify two values. On the other hand, it can explore both sides of a $\otimes$.

The relation of compatibility for equality C is designed in order to exclude moves that are *almost* equivalent. For instance, let us consider the following structure:



Then the two moves $\mathsf{m}_1 = (\alpha, v, \{\mathsf{inl}(\beta), \mathsf{inr}(\beta)\})$ and $\mathsf{m}_2 = (\alpha, v, \{\mathsf{inl}(\beta), \mathsf{inr}(\gamma)\})$ are not equivalent. However, they morally correspond to the same "move", from an abstract point of view. The relation C that we picked enforces that these two moves are not compatible for equality. That is, $\neg(\mathsf{m}_1\ C\ \mathsf{m}_2)$ and consequently $\mathsf{m}_1 \smile \mathsf{m}_2$.

Proposition 4.38. *The triple* $\mathsf{Event}(A)$ *is a nominal structured tree event structure.*

Proof. The only property we need to check is the one related to the the conflict relation. Let e, e' such that $e \smile e'$ and $e' \leq e''$. Then in the case where $e\ C\ e''$, it entails that, writing $\ulcorner e \urcorner = (\alpha, v, S)$ (and similarly for e', e''), $v \cap v'' = v \cap v'$ and therefore $\mathsf{label}(v \cap v'') = \mathsf{label}(v \cap v') = \mathsf{cells}$, thus $e'' \smile e$. In the case where $\neg(e\ C\ e')$ then $\neg(e\ C\ e'')$ and therefore $e \smile e''$ as well. Furthermore, if $e \simeq e'$ and $\neg(e = e')$, then $\neg(e\ C\ e')$, which entails $e \smile e'$ as expected. Therefore, the event structure is linear. □

Therefore, the positions of $\mathsf{Event}(A)$ form a prime algebraic domain. We denote it $(\mathsf{Pos}(A), \sqsubseteq_A, \bot_A)$. Furthermore, let us note that the polarity function extends to the event structure. Given $\ulcorner e \urcorner = \mathsf{m} = (\alpha, v, S)$, wet set:

$$\lambda(e) = \lambda(\mathsf{m}) = \lambda(v) = \lambda(\alpha)$$

leading to a **polarised event structure**.

Lemma 4.39. *In a dialogue game, two different initial events are incompatible.*

Proof. If two initial events are compatible for equality, for them to be different means they will be rooted at the same initial cell and pick each a different value. But then the intersection of these values will be an initial cell. □

One can notice that these games are actually fairly close to the nominal games from [31], where a move was also defined as a triple whose first and final element were names. However, this time, because of the linearity, we restrict the amount of threads that can depart from a move. The positions of the dialogue game A can be seen as (not-equivariant) subtrees of A, subject to some additional properties, such as, the leaves of the sub-tree can be values only if they do not justify cells in the original tree.

For technical purposes, we extend the relation C to positions: $p \, C \, q$ if $\forall e \in p, \forall e' \in q. e \; C \; e'$. In particular, $p \simeq q \wedge p \; C \; q \Rightarrow p = q$. Let us note that two non empty compatible positions in a dialogue game are rooted at the same cell. That is, their root cells have the same name. Finally, as expected, $p \uparrow q \Rightarrow p \; C \; q$.

4.5.3 On moves and events

Despite the fact that the set of moves, together with its associated structure, does not form an event structure, it nevertheless seems simpler to work with moves rather than events in the form of lists. For instance, let us consider two positions p, q such that there exists an event e, $p = q \uplus e$. Then writing $e = \mathsf{m}_1..\mathsf{m}_n.\mathsf{m}_{n+1}$, the event $e' = \mathsf{m}_1...\mathsf{m}_n$ is in q, otherwise the position q would not be downward closed. So, the additional data that p has compared to q is precisely $\ulcorner e \urcorner = \mathsf{m}_{n+1}$. In that case, we write $q = p \uplus \{\mathsf{m}_{n+1}\}$, to highlight the move. Similarly, a position p is perfectly described by the set of moves that happen in it, that is, by the set $\{\ulcorner e \urcorner \mid e \in p\}$. In that case, we write $\mathsf{m} \in p$ as abuse of notation for, formally, $\exists e \in p. \mathsf{m} = \ulcorner e \urcorner$. Furthermore, given two positions p, p' then $p \sqcup p' = \{\mathsf{m} \mid \mathsf{m} \in \mathsf{p} \vee \mathsf{m} \in \mathsf{p'}\}$ and $p \sqcap p' = \{\mathsf{m} \mid \mathsf{m} \in \mathsf{p} \wedge \mathsf{m} \in \mathsf{p'}\}$. Hence, when dealing with positions, it is enough to deal with moves. This is sum up in the following proposition.

Proposition 4.40. *A position is perfectly defined by the set* $\{\ulcorner e \urcorner \mid e \in p\}$.

In the sequel, we will establish some simple properties relating relations between moves and relation between events.

Lemma 4.41. *Given two moves* $\mathsf{m} = (\alpha, v, S)$ *and* $(\alpha', v', S') = \mathsf{m}'$ *the three properties are equivalent:*

- $\mathsf{m} \leq \mathsf{m}'$.
- $\mathsf{m} \; C \; \mathsf{m}'$ *and* $v \leq v'$.
- $\exists e, e'$ *such that* $\ulcorner e \urcorner = \mathsf{m}$, $\ulcorner e' \urcorner = \mathsf{m}'$ *and* $e \leq e'$.

Recalling that $\leq$ is the closure of the justifying relation $\vdash$ under transitivity and reflexivity, we simply have, for the proof, to consider a chain leading up to m, and see that we can extend it to m'.

We remind that two moves m, m' are compatible, written $\mathsf{m} \uparrow \mathsf{m}'$ if they are not in conflict $\neg(\mathsf{m} \smile \mathsf{m}')$. This translates into $\mathsf{m} \uparrow \mathsf{m}' \Leftrightarrow \mathsf{m} \; C \; \mathsf{m}' \wedge \mathsf{label}(v \cap v') = \mathsf{value}$ where v, v' are the values of m, m' respectively.

Lemma 4.42.

- $\mathsf{m} \uparrow \mathsf{m}' \;\Leftrightarrow\; \exists e, e'$ *such that* $\ulcorner e \urcorner = \mathsf{m}$, $\ulcorner e' \urcorner = \mathsf{m}'$ *and* $e \uparrow e'$.

- $\mathsf{m} \smile \mathsf{m}' \Leftrightarrow \forall e, e'.(\ulcorner e \urcorner = \mathsf{m} \wedge \ulcorner e' \urcorner = \mathsf{m}') \Rightarrow e \smile e'$.
- $\mathsf{m}\, C\, \mathsf{m}' \Leftrightarrow \exists e, e'.\ulcorner e \urcorner = \mathsf{m}.\ \ulcorner e' \urcorner = \mathsf{m}' \wedge e\, C\, e'$.

Proof. We consider two events e, e' such that $\ulcorner e \urcorner = \mathsf{m}$ and $\ulcorner e' \urcorner = \mathsf{m}'$, and such that all cells c that correspond to nodes that are not below m or m', are such that $\mathsf{c}\#\mathsf{m}, \mathsf{m}'$. Then, let us consider two cells c, c' belonging to some moves in e, e' such that $\mathsf{c} \simeq \mathsf{c}' \wedge \mathsf{c} \neq \mathsf{c}'$. Then, this cell does not correspond to a node below m, m', otherwise they would be compatible. Hence we can change the name of, for instance, c with a permutation π, such that it does not affect its last move $\ulcorner \pi \cdot e \urcorner = \mathsf{m}$. □

Definition 4.43.
- *In an event structure, two events e, e' are **independent** if they are compatible $e \uparrow e'$ and not related by the partial order $\neg((e \leq e') \vee (e' \leq e))$.*
- *By analogy, we say two moves m, m' are independent, if $\mathsf{m} \uparrow \mathsf{m}' \wedge (\neg((\mathsf{m} \leq \mathsf{m}') \vee (\mathsf{m}' \leq \mathsf{m}))$.*

Lemma 4.44. m, m' *are independent if and only if there exists e, e' such that $\mathsf{m} = \ulcorner e \urcorner, \mathsf{m}' = \ulcorner e' \urcorner$, and the events e, e' are independent.*

Proof. If there exists two events e, e' such that $\ulcorner e \urcorner = \mathsf{m}$, $\ulcorner e \urcorner' = \mathsf{m}'$, and $\mathsf{e} \uparrow \mathsf{e}'$, then it implies that $\mathsf{m} \uparrow \mathsf{m}'$, and, in particular $\mathsf{m}\ C\ \mathsf{m}'$. As e is not a sublist of e', and neither the other way around, it implies that $\neg(v \leq v')) \wedge \neg(v' \leq v)$, where v, v' are the values appearing in m, m'. Therefore, the moves are not comparable with $\leq$. The reverse direction is similar. □

Given a move $\mathsf{m} = (\alpha, v, S)$, we write $\ulcorner \mathsf{m} \urcorner$ for $(\ulcorner \alpha \urcorner, \ulcorner v \urcorner, \ulcorner S \urcorner)$, where $\ulcorner S \urcorner = \{\ulcorner \mathsf{c} \urcorner \mid \mathsf{c} \in S\}$. The events $e = \mathsf{m}_1 ... \mathsf{m}_n$ can be seen as $e \simeq \ulcorner \mathsf{m}_1 \urcorner ... \ulcorner \mathsf{m}_n \urcorner$, since the additional information that $e_{k+1} = \mathsf{m}_1 ... \mathsf{m}_{k+1}$ has compared to e_k is fully encoded in $\ulcorner \mathsf{m}_{k+1} \urcorner$.

As a conclusion of this Section, when it comes to positions and operations on them, one can work with moves instead of events. Furthermore, for a move m, we can often refer to it as $\ulcorner m \urcorner = (\alpha, \ulcorner v \urcorner, \{\beta_1, ..., \beta_n\})$, not mentioning the nodes happening before in the tree.

4.5.4 Lifting the operations to events and positions

The goal of this section is to lift the operations on dialogue games to event structures and positions. That is, we would like to characterise the set of positions corresponding to denotations of formulas of tensorial logic. We deal with the three connectives $\oplus, \otimes, \neg$, but also tackle the $\triangleright$ operation. Finally, we introduce the notion of transverse positions on $A \triangleright B$, and prove that transverse positions project well on A and B.

First, let us note that when we describe an event structure coming from a dialogue game, we only need to specify the set $|E|$. Indeed, the partial order relation is the prefix relation, and the conflict relation is structurally defined in the definition of dialogue game. Given a dialogue game A, we write $\mathsf{Event}(A)$ for its associated event structure. First, we give a description of the events and positions of our building blocks, the dialogue games $0, 1$ and X.

- $\mathsf{Event}(0) = \emptyset$ and therefore $\mathsf{Pos}(0) = \bot$.
- $\mathsf{Event}(1) = \{\mathsf{m} \mid \ulcorner \mathsf{m} \urcorner = (\alpha, \bullet, \emptyset)\ \alpha \in \mathbb{A}_{\mathsf{cells}}\}$ and therefore $\mathsf{Pos}(1) = \bot \uplus \mathsf{Event}(1)$
- $\mathsf{Event}(X) = \{\mathsf{m} \mid \ulcorner \mathsf{m} \urcorner = (\alpha, \bullet, \{\chi\}),\ \alpha \in \mathbb{A}_{\mathsf{cells}}, \chi \in \mathbb{A}_X\}$ and therefore $\mathsf{Pos}(X) = \{\bot\} \uplus \mathsf{Event}(X)$.

We recall that $\mathsf{Pos}^*(A)$ denotes the set of non-empty positions of A. Similarly, given a set of positions $X \subseteq \mathsf{Pos}(A)$, we define $X^* \subseteq \mathsf{Pos}^*(A)$ for $X \setminus \{\bot\}$.

We start with the operation $\oplus$. We establish that $\mathsf{Event}(A \oplus B) \simeq \mathsf{Event}(A) \oplus \mathsf{Event}(B)$, and give a description of the isomorphism.

- An event $e \simeq \ulcorner \mathsf{m}_1 \urcorner. \ldots .\ulcorner \mathsf{m}_n \urcorner$ of $\mathsf{Event}(A \oplus B)$ where $\ulcorner \mathsf{m}_1 \urcorner = (\alpha, \mathsf{inl}(v), S)$ is sent to $e' \simeq \mathsf{inl}(\ulcorner \mathsf{m}'_1 \urcorner).\ulcorner \mathsf{m}_2 \urcorner. \ldots .\ulcorner \mathsf{m}'_n \urcorner$, where $\ulcorner \mathsf{m}'_1 \urcorner = (\alpha, v, S)$.
- An event $e \simeq \ulcorner \mathsf{m}_1 \urcorner. \ldots .\ulcorner \mathsf{m}_n \urcorner$ of $\mathsf{Event}(A \oplus B)$ where $\ulcorner \mathsf{m}_1 \urcorner = (\alpha, \mathsf{inr}(v), S)$ is sent to $e' \simeq \mathsf{inr}(\ulcorner \mathsf{m}'_1 \urcorner).\ulcorner \mathsf{m}_2 \urcorner. \ldots .\ulcorner \mathsf{m}_n \urcorner$ where $\ulcorner \mathsf{m}'_1 \urcorner = (\alpha, v, S)$
- Furthermore, given an event $e = \mathsf{m}_1 \ldots \mathsf{m}_n$, and $e' = \mathsf{n}_1 \ldots \mathsf{n}_p$, such that $\ulcorner \mathsf{m}_1 \urcorner = (\alpha, \mathsf{inl}(\bullet), S)$ and $\ulcorner \mathsf{n}_1 \urcorner = (\alpha, \mathsf{inr}(\bullet), S')$, then if $e\ C\ e'$, calling v, v' the values of $\mathsf{m}_1, \mathsf{n}_1$ respectively, $v \cap v' = \alpha$ and hence $\mathsf{label}(v \cap v') = \mathsf{cell}$. Therefore, $e \smile e'$. Thus, we can prove that $\smile_{A \oplus B} \simeq \mathsf{inl}(\smile_A) \uplus \mathsf{inr}(\smile_B) \uplus \mathsf{inl}(\mathsf{Event}(A)) \times \mathsf{inr}(\mathsf{Event}(B)) \uplus \mathsf{inr}(\mathsf{Event}(B)) \times \mathsf{inl}(\mathsf{Event}(A))$.

This is obviously a bijection. As a result:

$$\mathsf{Pos}(A \oplus B) \simeq \mathsf{inl}(\mathsf{Pos}(A)) \overline{\uplus}\ \mathsf{inr}(\mathsf{Pos}(B))$$

We now tackle the $\otimes$ case. We prove that $\mathsf{Event}(A \otimes B) \simeq \mathsf{Event}(A) \overline{\otimes}_{\mathbb{A}_{\mathsf{cells}}} \mathsf{Event}(B)$, where the projections are:

$$\varpi_1 : \mathsf{Event}(A) \to \mathbb{A}_{\mathsf{cells}} : \quad ((\alpha, \mathsf{m}, S).\mathsf{m}_2. \ldots .\mathsf{m}_n) \mapsto \alpha$$
$$\varpi'_1 : \mathsf{Event}(B) \to \mathbb{A}_{\mathsf{cells}} : \quad ((\alpha, \mathsf{m}, S).\mathsf{m}_2. \ldots .\mathsf{m}_n) \mapsto \alpha$$

To simplify, we label both projections with the same name ϖ_1, since they act similarly.

- Let us consider an initial event e of $\mathsf{Event}(A \otimes B)$, then to e corresponds an initial move, that is, an element of the form $(\alpha, (v_1, v_2), \mathsf{inl}(S_1) \uplus \mathsf{inr}(S_2))$. Thus $e \simeq e_1 \times_{\mathbb{A}_{\mathsf{cells}}} e_2$, where $e_1 = (\alpha, v_1, S_1) \in \mathsf{Event}(A)$ and $e_2 = (\alpha, v_2, S_2)$. So $I_{A \otimes B} \simeq I_A \times_{\mathbb{A}_{\mathsf{cells}}} I_B$.
- Given an event $e \simeq \ulcorner \mathsf{m}_1 \urcorner.\ulcorner \mathsf{m}_2 \urcorner. \ldots .\ulcorner \mathsf{m}_n \urcorner$ of $\mathsf{Event}(A \otimes B)$, such that the cell α that justifies m_2 is of the shape $\mathsf{inl}(\ldots)$, then $e \simeq (\ulcorner i_1 \urcorner \times_{\mathbb{A}_{\mathsf{cells}}} \ulcorner i_2 \urcorner).\mathsf{inl}(\ulcorner \mathsf{m}_2 \urcorner). \ldots .\ulcorner \mathsf{m}_n \urcorner$. Similarly, in the case where the cell that justifies m_2 is of the shape $\mathsf{inr}(\ldots)$, then $e \simeq (\ulcorner i_1 \urcorner \times_{\mathbb{A}_{\mathsf{cells}}} \ulcorner i_2 \urcorner).\mathsf{inr}(\ulcorner \mathsf{m}_2 \urcorner). \ldots .\ulcorner \mathsf{m}_n \urcorner$.
- This isomorphism preserves the $\leq, \smile$ structure. This comes from the fact that in a dialogue game, two initial moves that are different are in conflict. Therefore, the conflict can only come from the projections on A and B.

Consequently, we get:

$$\mathsf{Pos}(A \otimes B) \simeq \mathsf{Pos}(A) \,\overline{\otimes}_{\mathbb{A}_{\mathsf{cells}}} \mathsf{Pos}(B).$$

Therefore, given $p \in \mathsf{Pos}(A \otimes B)$, we can define $p \upharpoonright A \in \mathsf{Pos}(A)$ and $p \upharpoonright B \in \mathsf{Pos}(B)$.

Finally, we describe the event structure of $\mathsf{Event}(\neg A)$, in function of $\mathsf{Event}(\top)$ and $\mathsf{Event}(A)$. We remind that $\mathsf{Event}(\top) = \{(\alpha, v_1, \beta) \mid \alpha, \beta \in \mathbb{A}_{\mathsf{cells}}\}$. Given $\ulcorner \mathsf{m} \urcorner = (\alpha, v_1, \beta)$ an initial move, we set $\varpi_2(\mathsf{m}) = \beta$, and given $e = (\alpha, v_2, \mathsf{c}).\mathsf{m}_2. \dots .\mathsf{m}_n \in \mathsf{Event}(A)$, we set $\varpi_1(e) = \alpha$ as before. We define the following operation:

$$\mathsf{Event}(\top) \times^{\mathbb{A}_{\mathsf{cells}}} \mathsf{Event}(A) = \{(i_\top, e_A), e_A = i_A.\mathsf{m}_2. \dots .\mathsf{m}_n \wedge \varpi_2(i_\top) = \varpi_1(i_A)\}$$

where $i_\top \in \mathsf{Event}(\top)$ and $e_A \in \mathsf{Event}(A)$. Then $\mathsf{Event}(\neg A) = \mathsf{Event}(\top) \uplus (\mathsf{Event}(\top) \times^{\mathbb{A}_{\mathsf{cells}}} \mathsf{Event}(A))$. Therefore, writing Pos^* for the non-empty positions, we have:

$$\mathsf{Pos}(\neg A) = \mathsf{Pos}(\top) \uplus (\mathsf{Pos}^*(\top) \times^{\mathbb{A}_{\mathsf{cells}}} \mathsf{Pos}^*(A))$$

where the functions ϖ_1, ϖ_2 have been extended to non-empty positions straightforwardly as before.

In this last paragraph, we interest ourselves in the structure of $A \triangleright B = A \otimes \neg B$. We start with the events.

$$\begin{aligned} \mathsf{Event}(A \triangleright B) &= \mathsf{Event}(A) \,\overline{\otimes}_{\mathbb{A}_{\mathsf{cells}}} \mathsf{Event}(\neg B) \\ &= \mathsf{Event}(A) \,\overline{\otimes}_{\mathbb{A}_{\mathsf{cells}}} (\mathsf{Event}(\top) \uplus \mathsf{Event}(\top) \times^{\mathbb{A}_{\mathsf{cells}}} \mathsf{Event}(B)) \end{aligned}$$

This allows us to define the set of positions:

$$\mathsf{Pos}(A \triangleright B) = \mathsf{Pos}(A) \,\overline{\otimes}_{\mathbb{A}_{\mathsf{cells}}} (\mathsf{Pos}(\top) \uplus (\mathsf{Pos}^*(\top) \times^{\mathbb{A}_{\mathsf{cells}}} \mathsf{Pos}^*(B))).$$

Definition 4.45. *A position p of $A \triangleright B$ is* ***transverse*** *if it is not of the form (e_A, i_B) where $e_a \in \bar{I_A}$ and $i_B \in I_{\neg B} = \mathsf{Event}(\top)$. We denote $\mathsf{Trans}(A \triangleright B)$ the subset of transverse positions of $\mathsf{Pos}(A \triangleright B)$.*

We would like to characterise the transverse positions of $A \triangleright B = A \otimes \neg B$. We have the following equations, where we write $p = i.p'$ to indicate that the unique initial move in the position p is i:

$$\begin{aligned} \mathsf{Trans}(A \triangleright B) &\simeq \mathsf{Pos}(A) \,\overline{\otimes}_{\mathbb{A}_{\mathsf{cells}}} (\mathsf{Pos}^*(\top) \times^{\mathbb{A}_{\mathsf{cells}}} \mathsf{Pos}^*(B)) \\ &\simeq \{\bot\} \uplus \{((\alpha, v, S).p,\ \ (\alpha, \bullet, \beta).(\beta, v, S').p') \mid (\alpha, v, S).p \in \mathsf{Pos}(A),\ (\beta, v, S').p' \in \mathsf{Pos}(B)\} \\ &\simeq \{\bot\} \uplus \{((\alpha, v, S).p,\ \ (\beta, v, S').p') \mid (\alpha, v, S).p \in \mathsf{Pos}(A), (\beta, v, S').p' \in \mathsf{Pos}(B)\} \\ &= \mathsf{Pos}(A) \,\overline{\otimes}\, \mathsf{Pos}(B) \end{aligned}$$

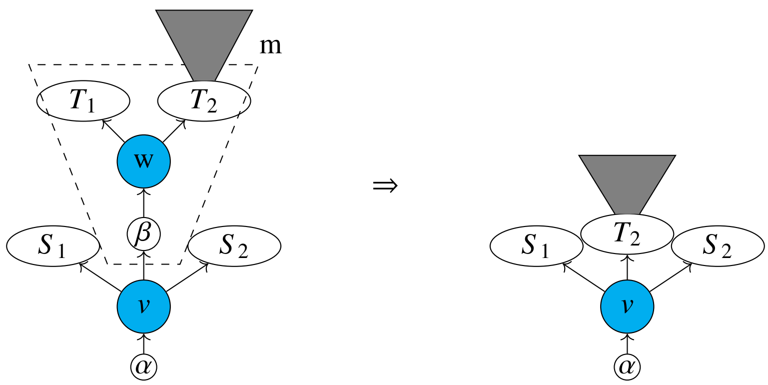
Therefore, given x a transverse position of $A \triangleright B$, one can talk of $x \upharpoonright A$ and $x \upharpoonright B$. Furthermore, given two positions of $x \in \mathsf{Pos}(A)$ and $y \in \mathsf{Pos}(B)$, such that $x \neq \bot_A \Leftrightarrow y \neq \bot_B$, one can form the position $x \triangleright y \in \mathsf{Trans}(A \triangleright B)$.

4.5.5 A quick note on the removal of moves

In order to later define composition, given a position in $x \in \mathsf{Pos}(A \triangleright B \triangleright C)$, we have to be able to speak of its $A \triangleright B$ part, just as its $B \triangleright C$ part, that is, define projections. Furthermore, if we want to look only at its external part, that is, its part in $A \triangleright C$ we have to forget its moves that belong in B.

Thanks to the isomorphisms described above, these projections are well-defined. On the other hand, in [30], where pointers are encoded through names just as we do here, projections are defined through erasure of moves. However, this does not translate well in our setting. We explain here why such a procedure fails. We denote by $\downharpoonleft$ the process of removing.

For instance, suppose that we try to remove the move m, as pictured below, where T_1 has no moves above, but T_2 has.



Suppose that the α in the above graph is a root, and consider the position $(\alpha, v, \{S_1, \beta, S_2\})$. It would be natural to imagine that we could consider its projection into the new arena where we have removed m. In order to do that, we need to replace the β in it with a new set of cells T_2. But what names shall we choose for T_2 ? There is no natural choice, hence the projection of a position does not lead to a position, but to a set of positions.

4.5.6 Projecting positions into lists

The positions and arenas are designed with tensorial logic in mind. However, at the end we would like to remove (by a quotient) the dynamic and project strategies onto an adequate model of linear logic. That is, given a formula F of linear logic, and its translation $(F)^F$ in tensorial logic (where we remind that $(.)^F$ denotes the focalised translation, defined in 2.4), we would like to project the denotation $(F)^F$ onto a denotation of F. As already explained, the denotation targeted is the category of separated polarised nominal relations defined in 3.4.3, hence the denotation of a formula would be a nominal set of polarised separated lists. We recall that $(.)^I$ is the reverse translation from formulas, and proofs, of tensorial logic into formulas, and proofs, of linear logic. It is defined such that $((A)^F)^I = A$ for any formula of A of linear logic.

Given a maximal position, seen as a subtree, we get its associated list by looking at its leaves. A position $p \in \mathsf{Pos}(A)$ is maximal if there are no positions $p' \in \mathsf{Pos}(A)$ such that $p' \geq p$. We write $\mathsf{PosMax}(A)$ for the set of maximal positions of A.

Given F a formula of tensorial logic, we define $\mathsf{proj}_F : \mathsf{PosMax}([\![F]\!]) \to \widehat{\mathsf{NomLinRelPol}((F)^I)}$ by induction on F. proj_F sends a maximal position to a list in the required set. However, this list is not necessarily separated nor linear. We will later need to check that the lists arising from the projecting positions of the strategies indeed project to relevant elements, that is, to nominal linear polarised relations. proj is a partial function. Indeed, untyped cells correspond to the elements $\top$ or 0, denoted by the empty-set in the category of polarised linear relations.

- If $F = X$ then $\mathsf{PosMax}([\![X]\!]) = \{(\alpha, \bullet, \chi) \mid \alpha \in \mathbb{A}_{\mathsf{cells}}, \chi \in \mathbb{A}_X\}$, then:

$$\mathsf{proj}(\alpha, \bullet, \chi) = (\chi, 1)$$

  This is in $\widehat{[\![(X)^I]\!]}_{\mathsf{NomLinRelPol}}$, since $(X)^I = X$.
- If $F = 1$ then $\mathsf{PosMax}([\![I]\!]) = \{(\alpha, \bullet, \emptyset) \mid \alpha \in \mathbb{A}_{\mathsf{cells}}\}$, and:

$$\mathsf{proj}(\alpha, \bullet, \emptyset) = (\bullet, 1).$$

  This is in $\widehat{[\![(I)^I]\!]}_{\mathsf{NomLinRelPol}}$, since $(I)^I = I$.
- If $F = 0$ then $\mathsf{PosMax}([\![0]\!]) = \{\bot\}$, and we set:

$$\mathsf{proj}(\bot) = \emptyset.$$

  that is, it proj is undefined for $\bot$. This is consistent since $\widehat{[\![(0)^I]\!]}_{\mathsf{NomLinRelPol}} = \emptyset$.
- if $F = F_1 \otimes F_2$ then $\mathsf{PosMax}([\![F_1 \otimes F_2]\!]) = \mathsf{PosMax}(F_1) \mathbin{\overline{\otimes}}_{\mathbb{A}_{\mathsf{cells}}} \mathsf{PosMax}(F_2)$. So given $x \in \mathsf{PosMax}([\![F_1 \otimes F_2]\!])$, then $x \simeq (x_1, x_2) \in \mathsf{PosMax}(F_1) \mathbin{\overline{\otimes}}_{\mathbb{A}_{\mathsf{cells}}} \mathsf{PosMax}(F_2)$. Then either $(x_1, x_2) = (\bot_{(F_1)}, \bot_{(F_2)})$, in which case it means either one of F_1, F_2 is 0 (suppose wlog F_1), and $(F_1 \otimes F_2)^I = 0 \otimes (F_2)^I$. Either none of them is $\bot$. In both case we have:

$$\mathsf{proj}((x_1, x_2)) = (\mathsf{proj}(x_1), \mathsf{proj}(x_2))$$

  This indeed belongs in $\widehat{[\![(F_1 \otimes F_2)^I]\!]}_{\mathsf{NomLinRelPol}}$ as $\widehat{[\![(F_1 \otimes F_2)^I]\!]}_{\mathsf{NomLinRelPol}} = \widehat{[\![(F_1)^I]\!]}_{\mathsf{NomLinRelPol}} \times \widehat{[\![(F_2)^I]\!]}_{\mathsf{NomLinRelPol}}$.

- if $F = F_1 \oplus F_2$ then $\mathsf{PosMax}([\![F_1 \oplus F_2]\!]) \simeq \mathsf{inl}(\mathsf{PosMax}(F_1)) \uplus \mathsf{inr}(\mathsf{PosMax}(F_2))$. Then:

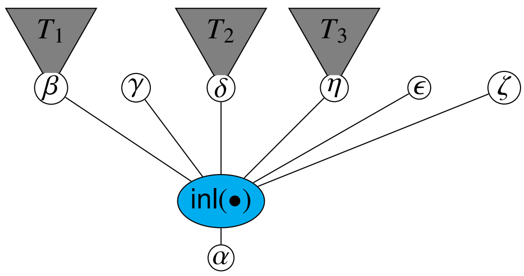
$$x \simeq \mathsf{inl}(x_{F_1}) \;\Rightarrow\; \mathsf{proj}(x) = \mathsf{inl}(\mathsf{proj}(x_{F_1}))$$
$$x \simeq \mathsf{inr}(x_{F_2}) \;\Rightarrow\; \mathsf{proj}(x) = \mathsf{inr}(\mathsf{proj}(x_{F_2}))$$

  In both cases these belong in $\widehat{[\![(F_1 \oplus F_2)^I]\!]_{\mathsf{NomLinRelPol}}}$ as $\widehat{[\![(F_1 \oplus F_2)^I]\!]_{\mathsf{NomLinRelPol}}} = \mathsf{inl}(\widehat{[\![(F_1)^I]\!]_{\mathsf{NomLinRelPol}}} \uplus \mathsf{inr}(\widehat{[\![(F_2)^I]\!]_{\mathsf{NomLinRelPol}}})$, since $(F_1 \oplus F_2)^I = (F_1)^I \oplus (F_2)^I$.
- If $F = \neg F_1$, then $\mathsf{PosMax}([\![F]\!]) \simeq \mathsf{Pos}^*(\neg) \times^{\mathbb{A}_{\mathsf{cells}}} \mathsf{PosMax}(F_1)$. Then given $x \simeq i_\top.e$, where $e \in \mathsf{PosMax}(F_1)$, we have:

$$\mathsf{proj}(x) = \mathsf{proj}(e)^\perp.$$

  Then as $(\neg F)^I = ((F)^I)^\perp$, we get $\mathsf{proj}(x) \in \widehat{[\![(F)^I]\!]_{\mathsf{NomLinRelPol}}}$.

The projection function could also has been defined by looking at a maximal position as a sub-tree of the initial dialogue game. We give a brief example. Let us consider a position of a dialogue game of type $T = A \oplus B$, such that the position is on the left resolution A, and such that A can be decomposed as $T_1 \otimes U \otimes I \otimes T_2 \otimes T_3 \otimes I \otimes V \otimes W$. We examine such a position in the figure below, where $\gamma \in \mathbb{A}_U, \epsilon \in \mathbb{A}_V, \zeta \in \mathbb{A}_W$, where $\mathbb{A}_U = \mathbb{A}_X$ if $U = X \in \mathsf{TVar}$, and $\mathbb{A}_U = \mathbb{A}_{\mathsf{cells}}$ if $U = \neg 0$ (and similarly for V, W)

proj(


)

$$= \mathsf{inl}(\mathsf{proj}(T_1).\mathsf{proj}(\gamma).\bullet\ \mathsf{proj}(T_2).\mathsf{proj}(T_3).\bullet\ .\mathsf{proj}(\epsilon).\mathsf{proj}(\zeta),$$

Furthermore, $\mathsf{proj}(\gamma) = \gamma$ if $\gamma \in \mathbb{A}_T$, and $\mathsf{proj}(\gamma)$ is undefined if $\gamma \in \mathbb{A}_{\mathsf{cells}}$ (and similarly for ϵ, ζ).

4.6 Nominal asynchronous games

4.6.1 On legal positions

We could straight away define a graph from the event structure having as nodes the positions of the event structures, and edges the events, or moves. However, this graph is not totally well-fitted for our setting. Indeed, in it, two untyped cells present in a position could share the same name, which is an unwanted configuration. This would correspond to a term such that the same name would appear under a λ-abstraction in two different places. To prevent that, we restrict the set of positions. We write $v\#_{\mathsf{cells}}w$ if $(\nu(v) \cap \nu(w)) \cap \mathbb{A}_{\mathsf{cells}} = \emptyset$. Given a set S of vertices of a structured tree, we write $\ulcorner S \urcorner$ for the set $\{\ulcorner v \urcorner \mid v \in S\}$.

Definition 4.46.
- *A move* $\mathsf{m} = (\alpha, v, S)$ *is* ***legal*** *if the untyped cells in it have different names:* $\forall \mathsf{c}, \mathsf{c}' \in \ulcorner S \urcorner; \mathsf{c}\#_{\mathsf{cells}}\mathsf{c}'$ *and furthermore* $\alpha\#\ulcorner S \urcorner$.
- *A position is* ***legal*** *if all the moves in it are, and furthermore, given* $\mathsf{m}, \mathsf{m}' \in p$*, such that* $\mathsf{m} = (\alpha, v, S), \mathsf{m}' = (\alpha', v', S')$*, then* $\mathsf{m} \neq \mathsf{m}' \Rightarrow \ulcorner S \urcorner \#_{\mathsf{cells}} \ulcorner S' \urcorner$.

In the sequel we will only consider legal positions. That is, if we consider a position p or a move m, we automatically assume that they are legal. Using the general domain of positions might prove useful for some proofs, but then we will specify it. We write $\mathsf{Legal}(A)$ for the set of legal positions of A. Working with legal positions also allows us to exclude some unwanted behaviour from our strategies. Indeed, a strategy could react differently if some moves by opponent would introduce two equal names. However, this does not correspond to any proof. Names are simply here to represent the different resources, or hypotheses, available at a certain point in the proof, and equality between names does not make sense in this context.

The restriction to legal positions might be troublesome when defining union of positions. That is, one might find two positions $p, q \in \mathsf{Legal}(A)$, $p \uparrow q$ but such that $p \sqcup q \notin \mathsf{Legal}(A)$. Indeed, the union might not be legal if the two positions use the same name for different cells. To ensure that two positions are name compatible for union, we introduce a "post-compatible" relation C_{post}, that selects those legal positions whose joints are legal.

- $\mathsf{m}\ C_{\mathsf{post}}\ \mathsf{m}'$ if $\mathsf{m}\ C\ \mathsf{m}' \ \wedge\ ((\mathsf{m} \neq \mathsf{m}') \Rightarrow \ulcorner S \urcorner \#_{\mathsf{cells}} \ulcorner S' \urcorner)$ where $\mathsf{m} = (\alpha, v, S)$ and $\mathsf{m}' = (\alpha', v', S')$
- $p\ C_{\mathsf{post}}\ q$ if $\forall \mathsf{m} \in p, \forall \mathsf{m}' \in q.\ \mathsf{m}\ C_{\mathsf{post}}\ m'$.

As expected, $p, q \in \mathsf{Legal}(A)$, $p\ C_{\mathsf{post}}\ q \Rightarrow (p\ C\ q\ \ \wedge (\ p \uparrow q \Rightarrow p \sqcup q \in \mathsf{Legal}(A)))$. It is worth noticing that if $p, q \in \mathsf{Legal}(A)$, and $p \uparrow q$, then there is a permutation π of $\mathbb{A}_{\mathsf{cells}}$ such that $p\ C_{\mathsf{post}}\ \pi \cdot q$, and $\{p, \pi \cdot q\}$ is bounded in $\mathsf{Legal}(A)$. This is proven below. Stated otherwise, if two legal positions are compatible in $\mathsf{Legal}(A)$, then it is possible to change their untyped names in such a way such that they remain compatible and their join becomes legal. We introduce the following terminology: we say that a move m **brings** a name a if, writing (α, v, S) then $a \in \nu(\ulcorner S \urcorner)$. Similarly, we say that an event e brings a name a if $\ulcorner e \urcorner$ brings a.

Lemma 4.47. *Let* p, q *be two legal positions of* A *such that* $p\ C\ q$*. Then there exists* $p' \simeq p$ *such that* $p'\ C_{\mathsf{post}}\ q$.

Proof. We split the names of $p \cap \mathbb{A}_{\mathsf{cells}}$ into two parts, calling them p_1, p_2. We set $p_1 = \nu(p \cap q) \cap \mathbb{A}_{\mathsf{cells}}$, and $p_2 = (\nu(p) \cap \mathbb{A}_{\mathsf{cells}}) \setminus p_1$. As p is legal, p_2 corresponds precisely to the set of cell-names brought by the moves in $p \setminus q$, seeing p, q as set of moves. Let π a permutation such that $\nu(\pi) \subseteq p_2 \cup z$ where $z\ \#\ q, p_1$, and such that $\forall a \in p_2.\ \pi \cdot a \in z$. Then $\pi \cdot p_2\ \#\ q$ and by definition of π, $\pi \cdot p_1 = p_1$ as $\nu(p_1) \subseteq \nu(q)$. Furthermore, legal positions are stable under permutation. Therefore, $\pi \cdot p$ is legal, and $\pi \cdot p\ C_{\mathsf{post}}\ q$. □

Remark 4.48. *All the moves within a legal position are automatically post-compatible to one-another.*

Typed names are dealt with differently than untyped names. Indeed, the typed names will be repeated in the play, to incorporate the fact that the strategy will establish axiom links between literals of opposite polarities. Therefore, it does not make sense to impose a condition similar as legality for typed names. To cope with the possibility that repetitions might occur, we will work up to closure under typed substitutions. Ξ_T denotes the set of strict substitutions of $\mathbb{A}_T$, called typed substitutions . We therefore define a new relation, written $\cong$, called congruence.

$$x \cong y \;\Leftrightarrow\; \exists \pi \in \mathsf{Perm}(\mathbb{A}_{\mathsf{cells}}), e \in \Xi_T.\, \pi \cdot (e \cdot x) = y.$$

We create two new relations C_{cells} and $C_{\mathsf{post,cell}}$, that are the restrictions of C, C_{post} to cells that are untyped.

- $\mathsf{m} = (\alpha, v, S)\ C_{\mathsf{cells}}\ \mathsf{m}' = (\alpha', v', S')$ if $\alpha \simeq \alpha \Rightarrow \alpha = \alpha'$ and $\forall \mathsf{c} \in S, \forall \mathsf{c}' \in S'.v(\mathsf{c}) \subseteq \mathbb{A}_{\mathsf{cells}}, v(\mathsf{c}') \subseteq \mathbb{A}_{\mathsf{cells}} \Rightarrow \mathsf{c}\ C\ \mathsf{c}'$.
- $p\ C_{\mathsf{cells}}\ q$ if $\forall \mathsf{m} \in p, \forall \mathsf{m}' \in q.\mathsf{m}\ C_{\mathsf{cells}}\ \mathsf{m}'$
- $e\ C_{\mathsf{cells}} e'$ if $e{\downarrow}\ C_{\mathsf{cells}}\ e'{\downarrow}$.
- $\mathsf{m}\ C_{\mathsf{post,cell}}\ \mathsf{m}'$ if $\mathsf{m}\ C_{\mathsf{cells}}\ \mathsf{m}'$ and writing $\mathsf{m} = (\alpha, v, S)$, $\mathsf{m}' = (\alpha', v', S')$, then $\mathsf{m} \neq \mathsf{m}' \Rightarrow \ulcorner S \urcorner \#_{\mathsf{cells}} \ulcorner S' \urcorner$.
- $p\ C_{\mathsf{post,cell}}\ q$ if $\forall \mathsf{m} \in p.\forall \mathsf{m}' \in q.\ \mathsf{m}\ C_{\mathsf{post,cell}}\ \mathsf{n}$,
- $e\ C_{\mathsf{post,cell}}\ e'$ if $e{\downarrow}\ C_{\mathsf{post,cell}}\ e'{\downarrow}$.

The extend straightforwardly to plays. Two plays $s : \star \twoheadrightarrow x, t : \star \twoheadrightarrow y$ satisfy $s\ C_{\mathsf{cells}}\ t$ if $x\ C_{\mathsf{cells}}\ y$. We prove the following properties:

Proposition 4.49.
- *If $p \uparrow q$ and $p\ C_{\mathsf{post,cell}}\ q$ then $p\ C_{\mathsf{post}}\ q$.*
- *If $\mathsf{m} \ncong \mathsf{n}$ and $\mathsf{m}\ C_{\mathsf{post,cell}}\ \mathsf{n}$ then $\mathsf{m}\ C_{\mathsf{post}}\ \mathsf{n}$.*

Proof. The first property is straightforward, since $p \uparrow q$ entails $p\ C\ q$. The second property follows from the fact that we consider m, n legal. Therefore, $\mathsf{m} \ncong \mathsf{n}$ implies that they are not, essentially, the same move with different names. Hence, we automatically got $\mathsf{m}\ C\ \mathsf{n}$. □

In the case where we forgot the condition $p \uparrow q$, then it becomes harder to create elements that become compatible.Finally, we prove that given two elements, we can use permutations to make them post compatible regarding untyped cells, and use substitutions to make them post compatible. We rely on the lemma 4.50.

Lemma 4.50. *Let f be a bijection between two finite subsets of $\mathbb{A}$. Then f can be completed into a permutation of $\mathbb{A}$ of finite support.*

Proof. Let us name X, Y the two subsets such that f sends X onto Y. As f is a bijection, the cardinality of X and Y is the same, and so is the cardinal of $Y \setminus (X \cap Y)$ and $X \setminus (X \cap Y)$. So consider g a function such that $g : Y \setminus (X \cap Y) \rightarrow X \setminus (X \cap Y)$ is a bijection. The union (in the sense union of graph) of f and g hence leads to a bijection $X \cup Y \rightarrow X \cup Y$. We can simply complete it into a full permutation π of $\mathbb{A}$ by letting π acting like the identity outside $X \cup Y$. Furthermore, as X and Y are finite, π has finite support. $\square$

At last, we present the last property that we will need regarding the C_{post} relation.

Proposition 4.51.

- *Let p, q be two legal positions. Then there exists p', q' such that $p' \cong p$, $q' \cong q$ and $p'\ C_{\mathsf{post}}\ q'$.*
- *Let p, q be two legal positions. Then there exists a permutation π of $\mathbb{A}_{\mathsf{cells}}$ such that $\pi \cdot p\ C_{\mathsf{post,cell}}\ q$.*

Given a move $\mathsf{m} = (\alpha, v, S)$ we write $\ulcorner S(m) \urcorner$ for $\ulcorner S \urcorner$.

Proof. We prove the two points at once. Let $p_1 = \{\mathsf{m} \mid \mathsf{m} \in p, \exists \mathsf{m}' \in y.\mathsf{m} \cong \mathsf{m}'\}$. As p is legal, every event in it brings different untyped names, distinct from the initial cell of p, and similarly for q. That is, given $\mathsf{m}_1, \mathsf{m}_2 \in p.\mathsf{m}_1 \neq \mathsf{m}_2 \Rightarrow \ulcorner S(\mathsf{m}_1) \urcorner \#_{\mathsf{cells}} \ulcorner S(\mathsf{m}_2) \urcorner$. Let us note that either the initial move of p is in p_1, or it is empty. In the case where it is not empty, $\nu(p_1) = \{\alpha\} \biguplus_{\mathsf{m} \in \mathsf{p}_1} \nu(\ulcorner S(\mathsf{m}_1) \urcorner)$, where α is the name of the initial cell of p. We can define a function $f : \nu(p_1) \cap \mathbb{A}_{\mathsf{cells}} \rightarrow \mathbb{A}_{\mathsf{cells}}$ such that for all $\mathsf{m} \in p_1$, given π exhibiting the equivalence $\pi \cdot \ulcorner S(\mathsf{m}) \urcorner \cap \mathbb{A}_{\mathsf{cells}} = \ulcorner S(\mathsf{m}') \urcorner \cap \mathbb{A}_{\mathsf{cells}}$ from the definition of p_1, $f \upharpoonright \nu(\ulcorner S(\mathsf{m}) \urcorner) \cap \mathbb{A}_{\mathsf{cells}} = \pi \upharpoonright \nu(\ulcorner S(\mathsf{m}) \urcorner) \cap \mathbb{A}_{\mathsf{cells}}$. We complete it into $f(\alpha) = \beta$, where β is the name of the initial cell of p_2. Furthermore, as y is legal, each m' in the definition p_1 brings different names as well, and and hence f establishes a bijection between $\nu(p_1) \cap \mathbb{A}_{\mathsf{cells}}$ and a subset of $\mathbb{A}_{\mathsf{cells}}$. Therefore, by applying the above lemma, we get a permutation π such that $\pi \cdot p\ C_{\mathsf{cells}}\ q$. Furthermore, doing the same reasoning as in the proof of 4.47, we can find a π' such that $\pi' \cdot (\pi \cdot p)\ C_{\mathsf{post,cell}}\ q$. As a consequence of the axioms of group actions $(\pi' \circ \pi) \cdot p\ C_{\mathsf{post,cell}}\ q$. Now, let us take two typed substitutions e_1, e_1, such that, for all $X \in \mathsf{TVar}$, e_1, e_2 send all names of p, q of type X to a unique name $c_X \in \mathbb{A}_X$. Then $e_1 \cdot (\pi' \circ \pi) \cdot p\ C_{\mathsf{post}}\ e_2 \cdot q$. $\square$

4.6.2 Nominal asynchronous games

Finally, we obtain a new graph from $\mathsf{Event}(A)$ by seeing its set of legal positions as a graph.

Definition 4.52. *Given a dialogue game A, and its associated event structure $\mathsf{Event}(A)$, the nominal graph $\mathsf{graph}(A)$ is defined as:*

- *having vertices the legal positions of A.*
- *having edges $x \xrightarrow{\mathsf{e}} y$ every time there is a move m such that $y = x \uplus \{\mathsf{m}\}$. In that case, we write $\ulcorner e \urcorner = \mathsf{m}$.*

Note that then there is a slight difference between moves and edges. That is, there might be two legal positions x, y such that $x \neq y$, and a single move m such that $x \uplus \mathsf{m}$, $y \uplus \mathsf{m}$ are legal positions. That is, a single move can correspond to several edges. However, given a position x and a move m, such that $x \uplus \mathsf{m}$ is legal, then m denotes a single edge. Therefore, we can write $x \xrightarrow{\mathsf{m}} y$. Therefore, we might speak about moves to refer to edges of the graph, and this should be clear from the context. Given a path s, we will write $\mathsf{m} \in s$ for $\mathsf{e} \in s, \ulcorner \mathsf{e} \urcorner = \mathsf{m}$.

In the $\mathsf{graph}(A)$ we can establish some basic definitions about paths.

Definition 4.53. *A path s is* ***legal*** *if it joins legal positions,*

$$x \overset{s}{\twoheadrightarrow} y \text{ and } x, y \in \mathsf{Legal}(A).$$

and ***alternating*** *if it alternates between $O - P$ move:*

$$v_n \xrightarrow{\mathsf{m}_1} v_{n+1} \xrightarrow{\mathsf{m}_2} v_{n+2} \Rightarrow \lambda(\mathsf{m}_1) = -\lambda(\mathsf{m}_2)$$

Proposition 4.54. *A path s is legal if and only if it starts at a legal position x, and satisfies:*

- $\forall \mathsf{m} \in s$, m *is legal and* $\ulcorner S(\mathsf{m}) \urcorner \#_{\mathsf{cells}} x$.
- $\forall \mathsf{m}.\mathsf{m}' \in S. \ulcorner S(\mathsf{m}) \urcorner \#_{\mathsf{cells}} \ulcorner S(\mathsf{m}') \urcorner$

The proof is straightforward.

Definition 4.55. *A* ***play*** *in a simple dialogue game is a path in its graph such that its starting node is the empty position, written $\star$.*

We denote by $\mathsf{Play}(A)$ the set of plays of the graph $\mathsf{graph}(A)$. We furthermore write $\mathsf{Legal}(\mathsf{Play}(A))$ for the set of legal plays, and $\mathsf{Alt}(\mathsf{Play}(A))$ the set of alternating plays.

Proposition 4.56. *A path $s = \mathsf{m}_1.\mathsf{m}_2.\dots.\mathsf{m}_n$ is a play if:*

- m_1 *is a initial move and the unique one of the sequence.*
- *for every $i \in [1, n]$, there exists a sub-sequence of s written $\mathsf{m}_\alpha.\mathsf{m}_\beta.\dots\mathsf{m}_\gamma.\mathsf{m}_i$ such that* $\mathsf{m}_\alpha \vdash \mathsf{m}_\beta \vdash \dots \vdash \mathsf{m}_\gamma \vdash \mathsf{m}_i$.
- *Each m_i is unique.*

Proof. The proof is done by induction on the length of s. If s is of size 1, then $s = \mathsf{m}_1$, m_1 corresponds to an initial edge, and thus indeed to a path from the root. So let us suppose that $s_{\leq n}$ (the restriction of s to its n^{th} first moves) reach a position p. Then by the third point, m_{n+1} does not appear in p. By the second point, there is a move of p that justifies m_{n+1}. Therefore, $p \uplus \mathsf{m}_{n+1}$ is a position, and s is a play. □

Finally, we say that a legal position is **balanced** if it can be reached by an alternating play.

We furthermore would like to endow our graph with a notion of homotopy between paths, to emphasise when their differences are bureaucracy. A prominent feature of innocence is that two paths that are co-initial and co-final, that is, paths having same initial and final positions, are homotopic. This might not be always the case. For instance, in the case of a programming language with control, the order in which the arguments are interrogated matters. Therefore the two paths that correspond to two programs interrogating the two arguments in different orders are not homotopic. We introduce nominal asynchronous graphs below.

Definition 4.57. *A nominal* ***asynchronous graph*** *is a pair* $(\mathcal{G}, \diamond)$ *consisting of a nominal graph* $\mathcal{G}$ *together with permutation tiles* $\diamond$ *of between co-initial and co-final paths of length two. We furthermore require that the homotopy relation is nominal:* $f \diamond g \iff \pi(f) \diamond \pi(g)$.

From these tiles, we can define a notion of homotopy between paths in the graph.

Definition 4.58. *We establish the* ***homotopy relation****, written* $\sim$*, between co-initial and co-final directed paths, as being the symmetric, reflexive and transitive closure of the intermediate relation* $\tilde{\diamond}$*, where* $\tilde{\diamond}$ *is the binary relation between co-initial and co-final paths of length greater than two defined as follows:*

$$u.s'.v \;\tilde{\diamond}\; u.t'.v \quad \text{if and only if} \quad s' \diamond t'.$$

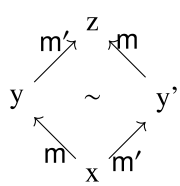
where s', t' *are paths of length two.*

So, given our graph $\mathsf{graph}(A)$, we add to it a relation $\diamond$ such that it becomes a nominal asynchronous graph. We define $\diamond$ to be such that every two moves that are independent have a tile. Or, more formally, we establish the diamond relation between edges e, e' such that $\ulcorner e \urcorner = \mathsf{m}, \ulcorner e' \urcorner = \mathsf{m}'$ and m, m' are independent.

Definition 4.59. *We define* $\diamond$ *as the smallest relation between every co-initial and co-final paths of length 2. That is,* $f \diamond g$ *if:*

$$f : x \xrightarrow{\mathsf{m}} y \xrightarrow{\mathsf{m}'} z \quad \diamond \quad g : x \xrightarrow{\mathsf{m}'} y' \xrightarrow{\mathsf{m}} z.$$

We denote this permutation property by a tile in the graph :



Definition 4.60. *We say that two moves are* ***strongly compatible****, written* $\mathsf{m} \Uparrow \mathsf{m}'$ *if there is a tile that permutes them. That is, there is one legal position* x*, such that* $x \uplus \mathsf{m} \in \mathsf{Legal}(A)$*,* $x \uplus \mathsf{m}' \in \mathsf{Legal}(A)$*,* $x \uplus \mathsf{m} \uplus \mathsf{m}' = x \uplus \mathsf{m}' \uplus \mathsf{m} \in \mathsf{Legal}(A)$.

We present a characterisation of strong compatibility in our framework. Basically, two legal moves are strongly compatible if they are independent, and their cell names are different.

Lemma 4.61. *Given two moves* m, m' *of* M_A*, if* $\neg(\mathsf{m} \le \mathsf{m}') \wedge \neg(\mathsf{m}' \le \mathsf{m}) \wedge \mathsf{m} \uparrow \mathsf{m}'$ *and* m, m' *are post-compatible* $\mathsf{m}\ C_{\mathsf{post}}\ \mathsf{m}'$ *then* $\mathsf{m} \Uparrow \mathsf{m}'$.

Proof. Let e, e' such that $\ulcorner e \urcorner = \mathsf{m}$, $\ulcorner e' \urcorner = \mathsf{m}'$, $e\ C_{\mathsf{post}}\ e' \wedge e \uparrow e'$. Then let $f = e \setminus \mathsf{m}$, and $f' = e' \setminus \mathsf{m}'$. Then $f\ C_{\mathsf{post}}\ f'$. So $(f\downarrow) \sqcup (f'\downarrow)$ is a legal position. As $\neg(\mathsf{m} \le \mathsf{m}')$, $\mathsf{m} \notin (f'\downarrow)$. Similarly, $\mathsf{m}' \notin (f\downarrow)$. Thus, by definition, $((f\downarrow) \sqcup (f'\downarrow)) \uplus \mathsf{m}$ is a position, and similarly for $((f\downarrow) \sqcup (f'\downarrow)) \uplus \mathsf{m}'$, just as $p = (f\downarrow) \sqcup (f'\downarrow) \uplus \mathsf{m} \uplus \mathsf{m}'$. Furthermore, $p = (e\downarrow) \sqcup (e'\downarrow)$, and, as $e \uparrow e'$, $e\ C_{\mathsf{post}}\ e'$, it entails $p \in \mathsf{Legal}(A)$. Hence, we have a tile permuting those two moves. □

We define $\mathsf{Async}(A)$ as the asynchronous graph whose graph is $\mathsf{Graph}(A)$ and whose tiles are the relations $\diamond$ defined above. Given the fact that our graph is coming from an event structure, we can establish that two paths are homotopic in it if and only if they are co-initial and co-final. For completeness sake, one can find the proof below. Consequently, a path will be uniquely determined, up to homotopy equivalence, by its initial and final positions, and hence by its set of moves. In particular, this implies that the homotopy relations is entirely defined by the relation $\Uparrow$ on moves. That is, given two co-initial and co-final paths of length four, $\mathsf{e}.\mathsf{e}' : x \twoheadrightarrow y$ and $\mathsf{f}.\mathsf{f}' : x \twoheadrightarrow y$, then setting $\mathsf{m} = \ulcorner e \urcorner = \ulcorner \mathsf{f}' \urcorner$ and $\mathsf{m}' = \ulcorner \mathsf{e}' \urcorner.\ulcorner \mathsf{f}' \urcorner$, there is a tile between the two paths if and only if $\mathsf{m} \Uparrow \mathsf{m}'$.

Proposition 4.62. *In* $\mathsf{Legal}(A)$*,* $s \sim t$ *if and only if they are co-initial and co-final.*

Proof. Let $s, t : p \twoheadrightarrow q$. Let $\mathcal{M} = p \setminus q$, where p, q are being seen as two sets of moves. Then s, t correspond to two total orderings of the events of $\mathcal{M}$. The proof is done by induction on the length of s, t (note that they have equal length). If the length is null, then the two paths are equal and hence homotopic. Let $s = \mathsf{m}_1....\mathsf{m}_k$ and $t = \mathsf{n}_1...\mathsf{n}_k$. Then there is a j such that $\mathsf{n}_j = \mathsf{m}_1$. If $j = 1$ then, writing p' for the position such that $p \xrightarrow{\mathsf{m}_1} p'$, we can apply the induction hypothesis on the paths $\mathsf{m}_2...\mathsf{m}_k$ and $\mathsf{n}_2...\mathsf{n}_k; p' \twoheadrightarrow q$. If $j \neq 1$, then we can apply a sequence of homotopy steps $\mathsf{n}_{j-1}.\mathsf{n}_j \diamond \mathsf{n}_{j-1}.\mathsf{n}_j$, since $\mathsf{n}_{j-1}, \mathsf{n}_j$ are not related by the partial order, are compatible, and furthermore are such that they lead to legal positions. We then hit a path $t' : p \twoheadrightarrow q$ that is homotopic to t and such that n_j appears as the first move : $t' = \mathsf{n}_j.\mathsf{n}_1...\mathsf{n}_{j-1}.n_{j+1}..\mathsf{n}_k$. At this stage, we do exactly as above, considering the position p', the paths $p' \twoheadrightarrow q$ and applying the induction hypothesis. □

Similarly, the following proposition ensues.

Proposition 4.63. *Let* $\mathsf{m}_1, \mathsf{m}_2$ *two moves such that* $\mathsf{m}_1, \mathsf{m}_2$ *appear in different orders in some paths. Then* $\mathsf{m}_1 \Uparrow \mathsf{m}_2$.

Definition 4.64. *Given a dialogue game A, we will speak of the* ***arena*** *A for the asynchronous graph* $\mathsf{Async}(A)$*.*

4.7 Asynchronous Böhm graph

In this last section, we will examine how the Böhm trees produce asynchronous graphs, and how these ones relate to the former arenas. This follows closely the steps of [64], simply examining it through a nominal perspective. This section is not needed for the next chapters. Its primary use is to introduce the strategies of the next section, how they relate to terms, and gives some indication on how to deal with atomic types. Therefore, we will stay at an informal level.

Just as we have defined some asynchronous graphs starting from the dialogue games, that were structured nominal trees, one can define asynchronous Böhm graphs from the Böhm trees (see [64] for more on that). This is what we set out to do in this section. We furthermore notice that these can be seen as subgraphs of the graphs coming from the dialogue games.

We start by defining a operation on asynchronous graphs. Given two asynchronous graphs $\mathcal{G}_1 = (V_1, E_1, \diamond_1)$, and $\mathcal{G}_2 = (V_2, E_2, \diamond_2)$ we define the asynchronous graph $\mathcal{G}_1 \otimes \mathcal{G}_2$ as follows:

- $V_{\mathcal{G}_1 \otimes \mathcal{G}_2} = \{v_1 \otimes v_2 \mid v_1 \in V_1, v_2 \in V_2\}$.
- $E_{\mathcal{G}_1 \otimes \mathcal{G}_2} \simeq V_1 \times E_2 \uplus E_1 \times V_2$. That is, it has edges $v_1 \otimes v_2 \xrightarrow{e} v_1' \otimes v_2$, whenever there is an edge $v_1 \xrightarrow{e} v_1'$ in E_1, and $v_1 \otimes v_2 \xrightarrow{e} v_1 \otimes v_2'$ whenever there is an edge $v_2 \xrightarrow{e} v_2'$ in E_2.
- There is a tile between co-initial and co-final paths of length two in three cases:
  1. Firstly, between paths $u_1 \otimes v_2 \xrightarrow{e_1} v_1 \otimes v_2 \xrightarrow{e_2} w_1 \otimes v_2$, and $u_1 \otimes v_2 \xrightarrow{f_1} v_1' \otimes v_2 \xrightarrow{f_2} w_1 \otimes v_2$, whenever there is a tile in $\mathcal{G}_1$ between the two paths: $u_1 \xrightarrow{e_1} v_1 \xrightarrow{e_2} w_1 \diamond_1 u_1 \xrightarrow{f_1} v_1' \xrightarrow{f_2} w_1$
  2. Secondly, between paths as above with the role between E_1 and E_2 being reversed.
  3. Finally, between paths $u_1 \otimes u_2 \xrightarrow{e_1} v_1 \otimes u_2 \xrightarrow{e_2} v_1 \otimes v_2$, and $u_1 \otimes u_2 \xrightarrow{e_2} u_1 \otimes v_2 \xrightarrow{e_1} v_1 \otimes v_2$.

In this paragraph, we set out to define a way to play the Böhm tree, through an alternating sequence of "moves", akin to the moves coming from the dialogue games. To each $\Gamma-$ Böhm tree M of negated type T, that is, $T = \neg(T_1 \otimes ... \otimes T_n)$, we associate the transitional following rooted asynchronous graph $\mathcal{G}'(M)$:

- Its root is labelled Ω_T.
- For each transition edge $\neg(x_1, ..., x_n)f$ starting from the root, there is a transition $(O, \Omega_T \to \neg(x_1, ..., x_n).\mho_\perp)$. These are called opponent-transitions.
- If the node $\neg(x_1, ..., x_n)f$ justifies $l+m$ new Böhm trees M_i, of types $(T_1, ..., T_l, A_1, ..., A_m)$, where T_i are simple non atomic types, and A_i are atomic types, then there are edges $\neg(x_1, ..., x_n).\mho_\perp \xrightarrow{P} \neg(x_1, ..., x_n).f(\Omega_{T_1} \otimes ... \otimes \Omega_{T_n} \otimes \alpha_1 \otimes ... \otimes \alpha_m)$, whenever they are edges $\neg(x_1, ..., x_n)f \to \neg(x_1, .., x_n)f.\mathsf{in}_j\alpha_i$, where α_i is a initial element of A_i, and $\neg(x_1, ..., x_n).f(\Omega_{T_1} \otimes ... \otimes \Omega_{T_n} \otimes \alpha_1 \otimes ... \otimes \alpha_m)$ is seen as the root of $\mathcal{G}(M_1) \otimes .... \otimes \mathcal{G}(M_n)$, where M_i is the subtree of M of type T_i.

And finally, given a Γ-Böhm tree M of any type $T = T_1 \otimes ... \otimes T_n \otimes A_1 ... \otimes A_m$, where each A_i is an atomic type, and where T_i is $\otimes$-irreducible, we associate to it the following asynchronous graph $\mathcal{G}(M)$, defined as follows:

- Its root is $\mho_\bot$
- There is a P-transition $\mho_\bot \to (\Omega_1 \otimes ... \otimes \Omega_n \otimes \alpha_1 \otimes .... \otimes \alpha_n)$ whenever α_1, α_n are initial nodes of M_i, the subtree of M of type A_i. This node is the root of the graph $\mathcal{G}'(M_1) \otimes ... \otimes \mathcal{G}'(M_n)$, where M_i is the sub-graph of M of simple type T_i.

For instance, the asynchronous graph associated with the η-long Böhm tree term $\neg(x, w, f, g, h).h(x \otimes \neg u.(g(w \otimes \neg v.(f(u \otimes v))))))$, of type $\neg(X \otimes Y \otimes (\neg(Z \otimes W) \otimes (\neg(Y \otimes \neg W)) \otimes (\neg(X \otimes \neg Z)))$ is presented in the figure 4.8, where we forget about the patterns to make it look simpler.

$$\neg(x, w, f, g, h).h(x \otimes \neg u.g(w, \neg v.f(u, v)))$$
$$(P, \mho_\bot \to f(u \otimes v)) \uparrow$$
$$\neg(x, w, f, g, h).h(x \otimes \neg u.g(w, \neg v.\mho_\bot))$$
$$(O, \Omega_T \to \neg v.\mho_\bot) \uparrow$$
$$\neg(x, w, f, g, h).h(x \otimes \neg u.g(w, \Omega_\bot))$$
$$(P, \mho_\bot \to g(w \otimes \Omega_\bot)) \uparrow$$
$$\neg(x, w, f, g, h).h(x \otimes \neg u.\mho_\bot)$$
$$(O, \Omega_T \to \neg u.\mho_\bot) \uparrow$$
$$\neg(x, w, f, g, h).h(x \otimes \Omega_\bot)$$
$$(P, \mho_\bot \to h(x \otimes \Omega_\bot)) \uparrow$$
$$\neg(x, w, f, g, h).\mho_\bot$$
$$(O, \Omega_T \to \neg(x, w, f, g, h).\mho_\bot) \uparrow$$
$$\Omega_T$$
$$P, \mho_\bot \to \Omega_T \uparrow$$
$$\mho_\bot$$

Figure 4.8: Asynchronous Graph for $[\neg(x, w, f, g, h).h(x \otimes \neg u.(g(w \otimes \neg v.(f(u \otimes v))))))]_\alpha$

More interesting is the Böhm graph associated with the term $\neg(u, f).f(u) \otimes \neg(w \otimes g).g(w)$ of type $\neg(X \otimes \neg X) \otimes \neg(Y \otimes \neg Y)$, which is presented in figure 4.7.

The Böhm asynchronous graphs have to start with a proponent move due to the potential presence of free variables. For instance, the Böhm asynchronous graph associated with the $\{x : X\}$-Böhm tree coming from the term x, simply consists of the graph $\mho_\bot \to x$.

4.7.1 On arenas and Böhm trees

We present in figure below 4.10 the dialogue game associated with the type: $\neg(X \otimes Y \otimes (\neg(Z \otimes W) \otimes (\neg(Y \otimes \neg W)) \otimes (\neg(X \otimes \neg Z)))$. Forgetting the values, and putting emphasis on the moves, it can be presented in a simpler fashion, displayed in figure 4.11.

Now the asynchronous Böhm graph associated with the α-equivalence class of the term

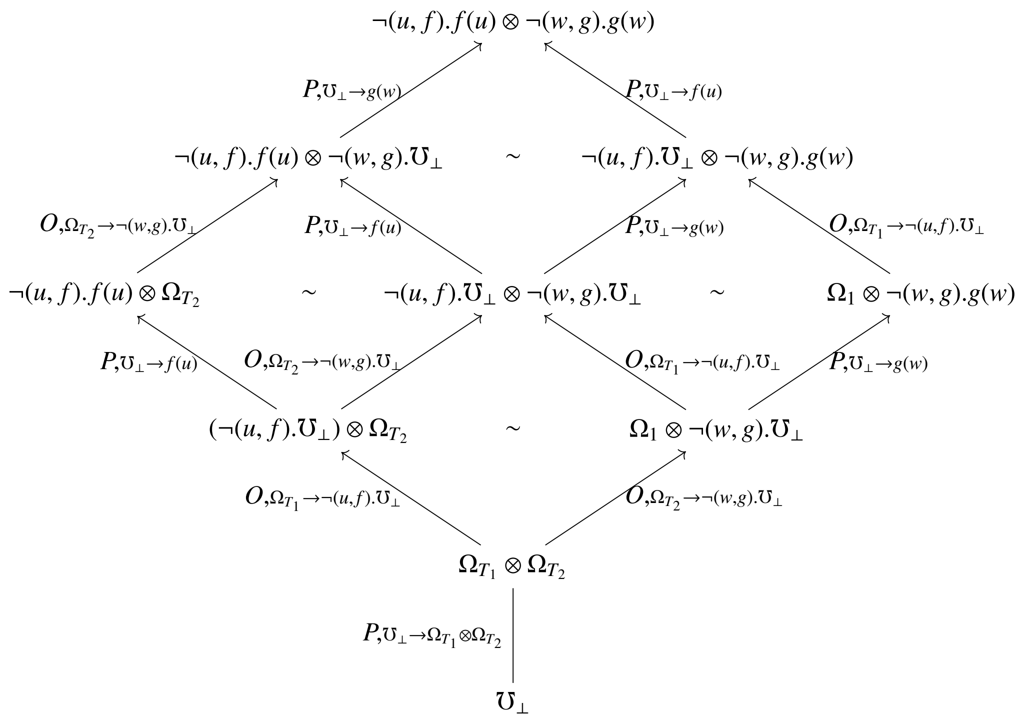


Figure 4.9: Asyncronous graph associated with $[\neg(u,f).f(u) \otimes \neg(w \otimes g).g(w)]_\alpha$

$\neg(x,w,f,g,h).h(x \otimes \neg u.(g(w \otimes \neg v.(f(u \otimes v))))))$ presented in figure 4.8, can be seen as a subgraph of the asynchronous graph of the arena as follows:

- each opponent move $\Omega \to \neg(x_1,...,x_n).\mho_\perp$ in the asynchronous Böhm graph corresponds to moves m such that $\ulcorner \mathsf{m} \urcorner = (\alpha, v, \{x_1,...,x_n\})$, where α is a name brought by proponent in a position corresponding to Ω in the arena.
- each player move move $\mho_\perp \to f(x_1,..,x_n,\mho_\perp,...,\mho_\perp)$ in the asynchronous Böhm graph corresponds to proponent moves m such that $\ulcorner \mathsf{m} \urcorner = (f, v, \{x_1,...,x_n,\alpha_1,..,\alpha_n\})$, where $\alpha_1,..,\alpha_n$ are names of $\mathbb{A}_{\mathsf{cells}}$ in the arena.

The main difference between the both structures is that the $\mho_\perp$ and the $\Omega_\perp$ have been replaced by names in the arena. This reflects on the symmetry of the arenas, and the asymmetry of the λ-calculus. The λ-term puts emphasis on the names brought by opponent, and reflects how the player is going to behave with those names. On the other hand, it does not highlight the fact that the player, in a symmetric setting, would also need to bring new names, that correspond to the different cells opponent could play in.

Furthermore the asynchronous Böhm graph defines a partial order on the moves of the arena: $P_1 < O_1 < \neg_{r,1}P < \neg_{r,2}O < \neg_{l,1}P < \neg_{l,2}O < \neg_3 P$, with certain conditions on the moves, such as the equality of the names $\omega = \rho$ and $\phi = \mu$.

Similarly, we can carry an similar analysis on the type $\neg(A \otimes \neg A) \otimes \neg(B \otimes \neg B)$, whose arena is presented below 4.12. Simplifying its move-structure by forgetting the values we get a graph as in figure 4.13.

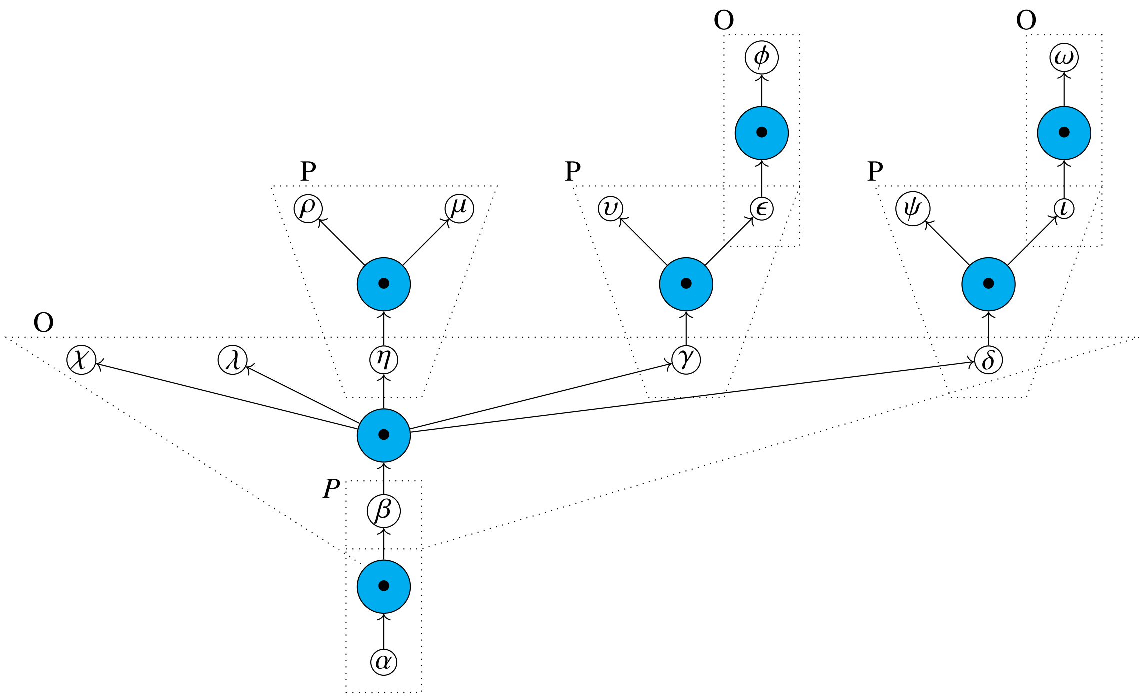


Figure 4.10: Dialogue game of the type: $\neg(X \otimes Y \otimes (\neg(Z \otimes W) \otimes (\neg(Y \otimes \neg W)) \otimes (\neg(X \otimes \neg Z)))$

Now the Böhm tree associated with the term $\vdash \neg(u, f).f(u) \otimes \neg(w, g).g(w)$ corresponds to the partial order on the moves $O_1 \leq P_1$ and $O_2 \leq P_2$, together with the conditions on names $\lambda = \mu$ and $\rho = \chi$. Similarly, it can be seen as a set of sequences respecting the partial order. Among them, the alternated sequences, are enough to faithfully represent the partial order. For instance, in that case, the sequences $\{P_0.O_1.P_1.O_2.P_2, P_0.O_2.P_2.O_1.P_1\}$. The goal of the next chapter 5 is to precisely characterise those sets of sequences that originate from a term, or, equivalently, from a proof.

Let us note that the legal plays are not perfectly fit to describe all elements coming from the λ-calculus. Indeed, the same name might be bound at different locations in the term if the binders have different scopes. For instance, the term $(\neg(f, x).fx) \otimes (\neg(f.x)fx)$ is a well-formed term of type $\neg\neg X \otimes \neg\neg Y$. However, a strategy associated with the α-equivalence of this term would be playing plays that correspond to $[\neg(f, x).fx \otimes \neg(g, y).gy]$, as the legal condition enforces that the cells introduced have all different names. This limitation is harmless and does not prevent the strategies from perfectly modelling the proofs.

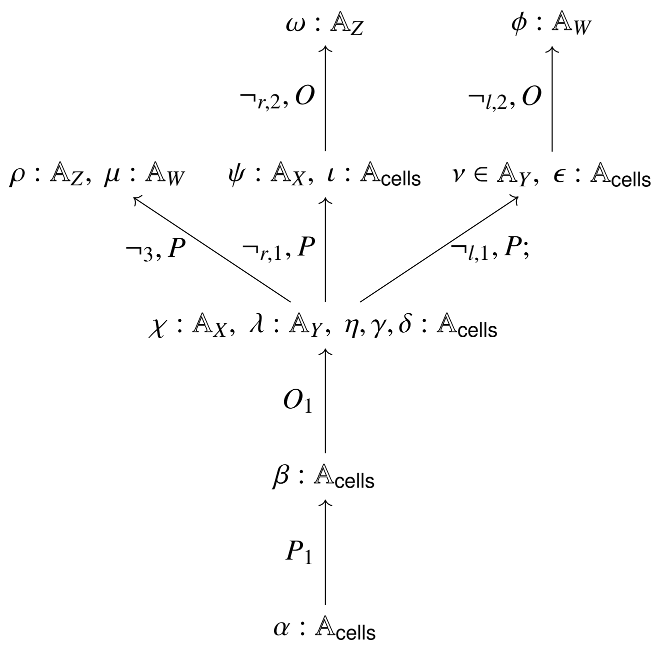


Figure 4.11: Dialogue game of the type: $\neg(X \otimes Y \otimes (\neg(Z \otimes W) \otimes (\neg(Y \otimes \neg W)) \otimes (\neg(X \otimes \neg Z)))$, simplified version

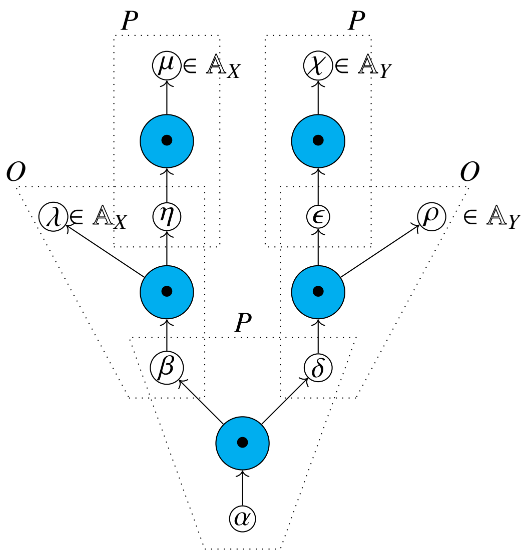


Figure 4.12: Dialogue game of the type:$\neg(X \otimes \neg X) \otimes \neg(Y \otimes \neg Y)$

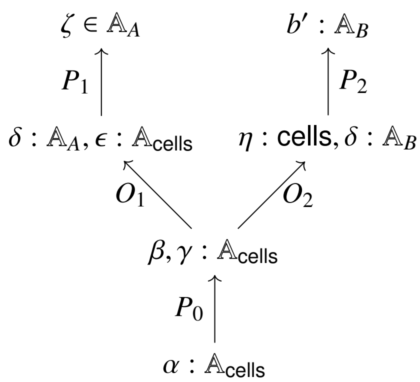


Figure 4.13: Dialogue game of the type:$\neg(A \otimes \neg A) \otimes \neg(B \otimes \neg B)$, simplified version

Chapter 5

Strategies for Tensorial Logic

We embark on the adventure of defining our category of games and morphisms. Along this chapter we will define several categories that are all based on the following recipe.

- The objects are positive dialogue games.
- The morphisms $A \to B$ are strategies on the negative pre-dialogue games $A \triangleright B$.

Strategies are abstract representations of sets of traces canonically associated with the α-equivalence classes of λ-terms. Similarly, they are abstract representations of proofs of tensorial logic. As such, they must satisfy a number of properties already presented before in the literature, chief among them is innocence [64]. Innocence was firstly exposed in the context of game semantics of programming languages [7, 49], though its presentation differs between the different papers. In the work of Abramsky, Jagadeesan and Malacaria, innocence appears as a property on strategies, stating that the way proponent will answer to a play will only depend on its last opponent move. This is called history-freeness, though the term is a bit confusing since opponent encodes, thanks to an index mechanism, a part of the history of the play in its last move. In the work of Hyland and Ong, these mechanisms were made more visible. Some view functions are defined, and these made clear exactly the history that the strategies are allowed to look at in order to produce a move. Finally, it was noticed in [64], that these functions actually state that the strategies are positional: the sets of plays can be seen as forming a graph, and the strategies answers depend only on the positions, not the "history", that is, the way the positions are reached. Furthermore, innocent strategies are perfectly described by the sets of positions they reach in a graph. So one can forget about the dynamic, that is, sequences of moves seen as interactions, and precisely characterise the way the strategies behave through their associated sets of positions.

This discovery is to be related of a long-standing logical paradox, the "staticity" of linear logic. Indeed, one notable feature of linear logic is its decomposition of the intuitionistic arrow ($A \to B \ = !A \multimap B$), that allows one to encode intuitionistic and therefore, the simply typed lambda-calculus, inside linear logic. Hence, it must possess a dynamic flavour; that is, a dynamic process, such that the lambda-calculus forms a sub-system of it. However, the models of linear logic are static, that is, they can be seen as relations, as, for instance, in section 3. This conflictual point of view was the point of departure of numerous works trying to relate static and dynamic semantics [12, 18, 29, 28, 68].

This chapter is mainly an adaptation of the work that Melliès carried in [64, 69] for the nominal structures defined in Chapter 4. The reader acquainted with his work will recognise a similar development, similar definitions and similar properties. Our strategies differ to those presented in these works in several ways. First, as nominal strategies they shall be closed under equivariance and the condition of determinacy is relaxed into the weaker condition of nominal determinacy. Furthermore, additional conditions shall be added for the strategies to be "logical". The equality between names does not translate in any logical property, therefore the strategies should not be able to act on it. This is imposed by closure under typed substitutions. We enable the strategies to deal with the axiom-links between propositional variables (so that they form a model of the free dialogue category over the discrete category VAR) by encoding the links through names: Opponent introduces typed names for negative literals, and Proponent repeats them in positive literals. Two literals are linked if Proponent played the same name as Opponent. As such, the strategy should not be able to produce a move corresponding to a positive atomic-variable on-demand, this one has to be linked to a negative one. This is imposed by the semi-linearity condition, that prevents the strategy from introducing new typed names. As, each typed name played by opponent corresponds to an occurence of a negative litteral, these ones should all be different. This is the frugality condition, introduced in Section 5.3.1.1. This one however should be however dealt with carefully, since it does not compose. Composition is recovered by closing the strategies under typed substitutions, just as in the case of separated relations. Finally, coherence is obtained through sequentiality structures, that were introduced in [69], and slightly adapted here to care for axiom-links and the nominal nature of our strategies. This way, we are able to project the strategies onto "external" positions, and recover nominal relations of the nominal polarised model of linear logic of Section 3. Moreover, this operation allows us to precisely select those relations that come from denotations of proofs. This will the subject of the next section 6.

The strategies are defined the usual way for nominal games. As fully complete strategies for tensorial logic they shall be innocent, total, and transverse. We will see that innocent strategies define families of functions, called sequentiality structures, that inform us of the dynamics of the strategies at a certain point of the interaction. We strengthen those in two ways. First, we force them to take into account all the context, and not only a part of it. This enforces them to be "linear", and not only affine. Furthermore, we encode the behaviour of axiom-links inside these sequentiality structures. That is, we impose that the strategy establishes axiom-links between opposite occurrences of literals. This gives us the perfect candidate for a full-completeness proof.

To start, we introduce the definition of nominal strategy.

Definition 5.1. *Given a negative dialogue game A, a **strategy** $\sigma : A$ is a non-empty set of legal plays of even lengths of the arena (that is, the asynchronous legal graph) associated to A such* *that:*

- *the plays are alternating.*
- *the strategy is closed under prefix. If* $s.\mathsf{m}.\mathsf{n} \in \sigma$ *and then* $s \in \sigma$*.*
- *the strategy is nominal deterministic :*

$$\forall s_1.\mathsf{m}_1.\mathsf{n}_1, s_2.\mathsf{m}_2.\mathsf{n}_2 \in \sigma.\ \ s_1.\mathsf{m}_1 \simeq s_2.\mathsf{m}_2 \ \Rightarrow\ s.\mathsf{m}_1.\mathsf{n}_1 \simeq s_2.\mathsf{m}_2.\mathsf{n}_2.$$

- *the strategy is equivariant :* $\forall s \in \sigma,\ s \simeq t \Rightarrow t \in \sigma$*.*

In our case, since we expect our strategies to be "logical", that is, denotations of proofs, we will also enforce them to be typed coherent.

Definition 5.2. *A strategy is* ***typed coherent*** *if it satisfies the two following conditions:*

- *(Closure under typed substitutions):* $\forall s \in \sigma. \forall e \in \Xi_T.\ e \cdot s \in \sigma$*.*
- *(semi-linearity): The strategy does not introduce typed names:*

$$s.\mathsf{m}.\mathsf{n} \in \sigma \Rightarrow \nu_T(s.\mathsf{m}) = \nu_T(s.\mathsf{m}.\mathsf{n})$$

In the sequel, we will only consider typed coherent strategies, and therefore will often omit to specify typed coherency. The typed coherency we only be dropped once we start speaking about frugal strategies, in Section 5.3.1.1, and this will be clearly written.

Remark 5.3. *In a typed coherent strategy, if we have two moves* $s.\mathsf{m}.\mathsf{n}_1$*,* $s.\mathsf{m}.\mathsf{n}_2$ *such that both belong to the strategy, then* $s.\mathsf{m}.\mathsf{n}_1\ C_{\mathsf{cells}}\ s.\mathsf{m}.\mathsf{n}_2$ *entails* $\mathsf{n}_1 = \mathsf{n}_2$*. Indeed, by nominal determinacy* $s.\mathsf{m}.\mathsf{n}_1 \simeq s.\mathsf{m}.\mathsf{n}_2$*, and, as the two positions are compatible with relation to untyped cells,* $s.\mathsf{m}.\mathsf{n}_1 \simeq_T s.\mathsf{m}.\mathsf{n}_2$*. Particularly, there exists a permutation* π *of* $\mathbb{A}_T$ *such that* π *lets* $s.\mathsf{m}$ *invariant, and* $\pi \cdot \mathsf{n}_1 = \mathsf{n}_2$*. As* $s.\mathsf{m}$ *has strong support, this entails* $\pi \# \nu(s.\mathsf{m})$*. In particular, as* $\nu_T(\mathsf{n}_1) \subseteq \nu_T(s.\mathsf{m})$*, this leads to* $\pi \cdot \mathsf{n}_1 = \mathsf{n}_1$ *and* $\mathsf{n}_1 = \mathsf{n}_2$*.*

Let us remind that given two legal positions x, y then there exists a permutation π of $\mathbb{A}_{\mathsf{cells}}$ and two typed substitutions e_1, e_2 such that $\pi \cdot (e_1 \cdot x)\ C_{\mathsf{post}}\ e_2 \cdot y$. As the strategies are closed under substitutions, given $s, t \in \sigma$, there is $s', t' \in \sigma$ such that $s' \cong s$, $t' \cong t$ and $s'\ C_{\mathsf{post}}\ t'$ (where the relation C_{post} is straightforwardly extended to plays by: $s : \star \twoheadrightarrow x\ C_{\mathsf{post}}\ t : \star \twoheadrightarrow y$ if and only if $x\ C_{\mathsf{post}}\ y$).

We remind that given a play, or path s we write $|s|$ for the length of the sequence s.

5.1 Innocent strategies and their structures

There are two definitions of innocence that are equivalent [64], one by defining a notion of view in asynchronous games and then redefining innocence in a similar way as for "non-asynchronous games", or one in a diagrammatic way that we prefer and present below. These definitions are almost similar to the ones from [64], except that we must take care that the legal positions are compatible when taking the union.

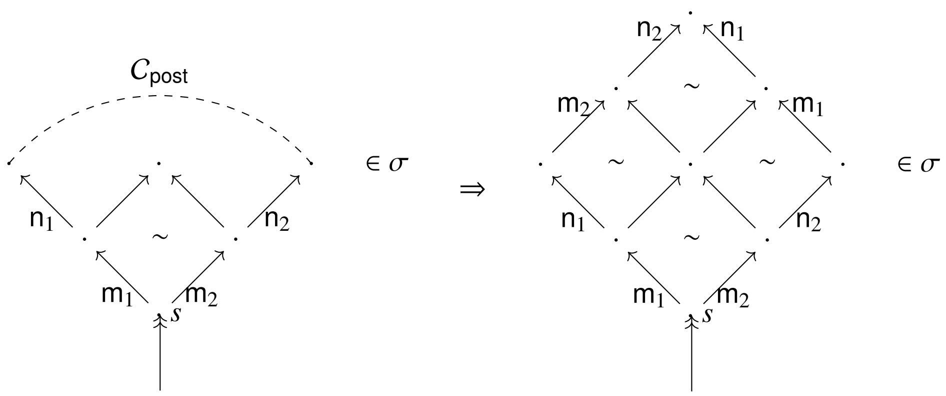
All the definitions and proposition in this chapter are simple adpations of those presented in [64] adapted to our new nominal structures. Due to the subtle differences between them, proofs are repeated. We also folow a similar development as in the original paper. Note that in [64] the event structure does not have a conflict relation (that is, only the multiplicative case is dealt with), although the general case has been already examined in [74].

Let us remind that given two moves m, n if $\mathsf{m} \not\cong \mathsf{n}$ and $\mathsf{m}\ C_{\mathsf{cells}}\ \mathsf{n}$ then $\mathsf{m}\ C_{\mathsf{post}}\ \mathsf{n}$, as proven in 4.49. Consequently, given two plays $s.\mathsf{m}_1.\mathsf{n}_1$ and $s.\mathsf{m}_2.\mathsf{n}_2$ belonging to a strategy σ, if $\mathsf{m}_1 \not\cong \mathsf{m}_2$, $\mathsf{n}_1 \not\cong \mathsf{n}_2$, then there exists a permutation of $\pi \in \mathsf{Perm}(\mathbb{A}_{\mathsf{cells}})$ such that $\pi \cdot s.\mathsf{m}_1.\mathsf{n}_1 = s.\mathsf{m}'_1.\mathsf{n}'_1$ and $s.\mathsf{m}'_1.\mathsf{n}'_1\ C_{\mathsf{cells}}\ s.\mathsf{m}_2.\mathsf{n}_2$, entailing $s.\mathsf{m}'_1.\mathsf{n}'_1\ C_{\mathsf{post}}\ s.\mathsf{m}_2.\mathsf{n}_2$. We remind that $\mathsf{m}_1 \uparrow \mathsf{m}_2$ if $\mathsf{m}_1\ C\ \mathsf{m}_2 \wedge \mathsf{label}(v_1 \cap v_2) = \mathsf{value}$, where v_1, v_2 are the values of the moves $\mathsf{m}_1, \mathsf{m}_2$ respectively.

Definition 5.4. *A (typed coherent) strategy is* ***forward consistent*** *if* $\forall s \in \sigma$ *and* $\mathsf{m}_1, \mathsf{m}_2, \mathsf{n}_1, \mathsf{n}_2$ *such that* $s.\mathsf{m}_1.\mathsf{n}_1,\ s.\mathsf{m}_2.\mathsf{n}_2 \in \sigma$, $\mathsf{m}_1 \neq \mathsf{m}_2$, $\mathsf{m}_1 \uparrow \mathsf{m}_2$ *then* $\mathsf{n}_1 \not\cong \mathsf{n}_2$ *. Moreover, if* $\mathsf{m}_1, \mathsf{n}_1$ *are such that* $s.\mathsf{m}_1.\mathsf{n}_1\ C_{\mathsf{post}}\ s.\mathsf{m}_2.\mathsf{n}_2$ *then we have:*

$$s.\mathsf{m}_1.\mathsf{n}_1. \uparrow s.\mathsf{m}_2.\mathsf{n}_2 \text{ and } s.\mathsf{m}_1.\mathsf{n}_1.\mathsf{m}_2.\mathsf{n}_2 \in \sigma.$$

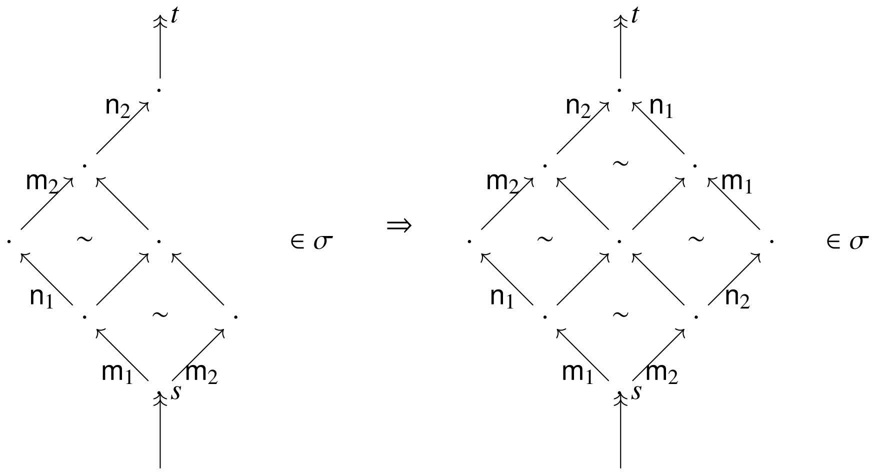
This can be translated to the following diagram :



A strategy is ***backward consistent*** *if:*

$$\forall s.\mathsf{m}_1.\mathsf{n}_1.\mathsf{m}_2.\mathsf{n}_2.t \in \sigma, \text{ such that } \neg(\mathsf{m}_1 \vdash \mathsf{m}_2) \text{ and } \neg(\mathsf{n}_1 \vdash \mathsf{m}_2) \text{ then}$$
$$\neg(\mathsf{m}_1 \vdash \mathsf{n}_2),\ \neg(\mathsf{n}_1 \vdash \mathsf{n}_2) \text{ and } s.\mathsf{m}_2.\mathsf{n}_2.\mathsf{m}_1.\mathsf{n}_1.t \in \sigma.$$

This translates into :



*Finally, a (typed coherent) strategy is **innocent** if it is both backward and forward consistent.*

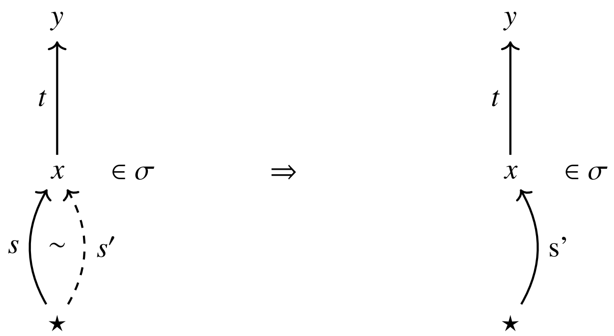
These definitions are standard and orginated from [64]. We check that their main properties remain unchanged within this nominal framework, and provide the proofs.

The sub-sections below are devoted to this. First, we prove that the way an innocent strategy responds to an opponent move depends only on the position reached. Then, we strengthen this property, demonstrating that actually, the strategy can be recovered from the set of positions it reaches. Based on this result, we seek a characterisation of the sets of positions that correspond to innocent strategies. Remaining is to prove that relational and sequential compositions are well-behaved towards each other, or, more precisely, that it is equivalent to relationally compose their sets of positions or to use the classic parallel and hide paradigm on their sets of plays. In order to tackle the proof, we introduce an important property of innocent strategies; they produce a function between accessible cells of their reached positions that tells where a future play might trigger a response. We call this weak sequentiality structure, and explain why it is necessary to strengthen it in order to achieve full completeness. This sequentiality structure can also been encoded as payoffs on positions and paths of the strategies, as done in [66].

5.1.1 Structure of the innocent strategies: positionality

A strategy is said to be positional if the way it reacts depends only on the position reached, and not on the path followed to reach it.

Definition 5.5 ([64])**.** *A strategy is **positional** if for all $s, s' : \star \twoheadrightarrow x \in \sigma$, such that $s \sim s'$, for all $t : x \twoheadrightarrow y$ such that $s \cdot t \in \sigma$, then $s'.t : \star \overset{s'.t}{\twoheadrightarrow} y \in \sigma$. This is drawn as follows.*



Proposition 5.6. *Every nominal innocent strategy is positional.*

The rest of this section is devoted to proving this. To start, we introduce, for a strategy σ, the set $\sigma^\bullet$ defined to be the set of positions it reaches:

$$\sigma^\bullet = \{x \in \mathsf{Legal}(A) \mid \exists s : \star \overset{s}{\twoheadrightarrow} x \in \sigma\}.$$

Second, let us note $\preceq$ the relation on paths $s \preceq t$ if there is a s' such that $s.s' \sim t$. We furthermore refine the homotopy relation into a second one, called $\sim_{OP}$ that acts by permuting only pairs of OP moves, we construct $\sim_{OP}$ in a similar way as we constructed $\sim$, but this time focussing on permutations between alternating paths of length 4, permutating pairs of $O-P$ moves.

Definition 5.7 ([64])**.** *We define the $\diamond_{OP}$ relation between co-initial and co-final paths of length* 4 *as follows:*

$$\mathsf{m}_1.\mathsf{n}_1.\mathsf{m}_2.\mathsf{n}_2 \diamond_{OP} \mathsf{m}_2.\mathsf{n}_2.\mathsf{m}_1.\mathsf{n}_1 \;\Leftrightarrow\; (\mathsf{m}_1.\mathsf{n}_1.\mathsf{m}_2.\mathsf{n}_2) \sim (\mathsf{m}_2.\mathsf{n}_2.\mathsf{m}_1.\mathsf{n}_1) \text{ and } \lambda(\mathsf{m}_i) = -\lambda(\mathsf{n}_i) = -1.$$

We say that this permutation of moves correspond to a single OP-homotopy step. We then define the intermediate $\tilde{\diamond_{OP}}$ to be the augmented relation from $\diamond_{OP}$ between paths of lengths more than four.

$$s.u.t \,\tilde{\diamond_{OP}}\, s.v.t \;\Leftrightarrow\; u \diamond_{OP} v$$

We define $\sim_{OP}$ as the reflexive, transitive and symmetric closure of $\tilde{\diamond_{OP}}$.

As the sequences of the strategies are alternating, and strategies are deterministic, this seems to be an appropriate notion for dealing with homotopy between paths of a strategy. Note however that two alternating paths can be $\sim$ homotopic without being $\sim_{OP}$ homotopic. One can draw a parallel between history freeness and the $\sim_{OP}$ relation: two $\sim_{OP}$ homotopic plays will have same player moves after same opponent moves. Finally, we write $s \preceq_{OP} t$ if there exists s' such that $s.s' \sim_{OP} t$.

Proposition 5.8. *Let σ be an innocent strategy and $s, t \in \sigma$ such that $s \preceq t$. Then $s \preceq_{OP} t$.*

Proof. For a given path u, we write $u_{\leq n}$ for the pre-sequence of u consisting of its n-first moves, and $u_{>n}$ for the path such that $u_{\leq n}.u_{>n} = u$. Furthermore, let us write s' for a path such that $s.s' \sim t$. We will prove the existence of a sequence of plays $t_0 \sim_{OP} t_1 \sim_{OP} t_2 \sim_{OP} t_3 ... \sim_{OP} t_{|s|/2}$, such that $t_0 = t$, and $t_i \in \sigma$, $t_i \sim_{OP} t$, and satisfying for each i, $(t_i)_{\leq 2*i} = s_{\leq 2*i}$. This way, $t_{|s|/2}$ will satisfy the following equality $t_{|s|/2} = s.t' \sim_{OP} t$, entailing $s \preceq_{OP} t$.

The required conditions are obviously satisfied for t_0, so let us assume they are true for t_n and we try proving the existence of t_{n+1}, assuming $2 * n < |s|$. To simplify notations, we write u for the path $s_{>2*n}$, and v for $t_{n,>2*n}$ (that is, such that $t_n = t_{n,\leq 2*n}.v$. Notably, as $s_{\leq 2*n} = t_{n,\leq 2*n}$, u, v are co-initial. In particular, this entails $u.s' \sim v$ since these are both co-initial and co-final.

As s, t_n are in σ, they are alternating. Let us write m for the moves of u and n for the moves of v. Formally, $u = \mathsf{m}_1....\mathsf{m}_n$, and $v = \mathsf{n}_1...\mathsf{n}_o$. Let us consider the first opponent move m_1 of u. As $u.s' \sim v$, m_1 is also a move of v. If it is the first one, then as $s_{\leq 2*n}.\mathsf{m}_1.\mathsf{m}_2 \in \sigma$, $t_{n,\leq 2*n}.\mathsf{n}_1.\mathsf{n}_2 \in \sigma$, σ is deterministic, and $s_{\leq 2*n}.\mathsf{m}_1 = t_{n,\leq 2*n}.\mathsf{n}_1$, this entails $\mathsf{m}_2 \simeq \mathsf{n}_2$. Now as $s.s'$ and t reach the same position, they are in compatible mode for equality, that is, $\forall i, j.\ \mathsf{m}_i \simeq \mathsf{n}_i \Rightarrow \mathsf{m}_i = \mathsf{n}_i$. So $\mathsf{n}_2 = \mathsf{m}_2$, and $t_{n+1} = t_n$.

We deal with the case where m_1 is not the first move of v. We write $v = v_0 = v'.\mathsf{o}_1.\mathsf{o}_2.\mathsf{m}_1.\mathsf{m}_2.v''$. As m_1 appears before o_1 in $u.s'$, and after in v, it entails $\mathsf{m}_1 \Uparrow \mathsf{o}_1$. In particular, there is a tile $\mathsf{m}_1.\mathsf{o}_1 \diamond \mathsf{o}_1.\mathsf{m}_1$. By backward consistency, $t_{n,\leq 2*n}.v \sim_{OP} t_{n,\leq 2*n}.v_1 \in \sigma$, where $v_1 = v'.\mathsf{m}_1.\mathsf{m}_2.\mathsf{o}_1.\mathsf{o}_2.v''$. So, by a sequence of backward consistency steps, we can push $\mathsf{m}_1.\mathsf{m}_2$ as the start of v as we did from v_0 to v_1. This is represented in figure 5.1. This way, we obtain a v_{final} such that $t_{n,\leq 2*n}.v_{final} \in \sigma$, and $t_{n,\leq 2*n}.v_{final} \sim_{OP} t_n$. We deduct, as above, that the two first moves of v_{final} are equal to those of u, and set $t_{n+1} = t_n.v_{final}$. This concludes the proof. □

We also need a second lemma before proving the proposition, that states that strategies are closed under $\sim_{OP}$ homotopy.

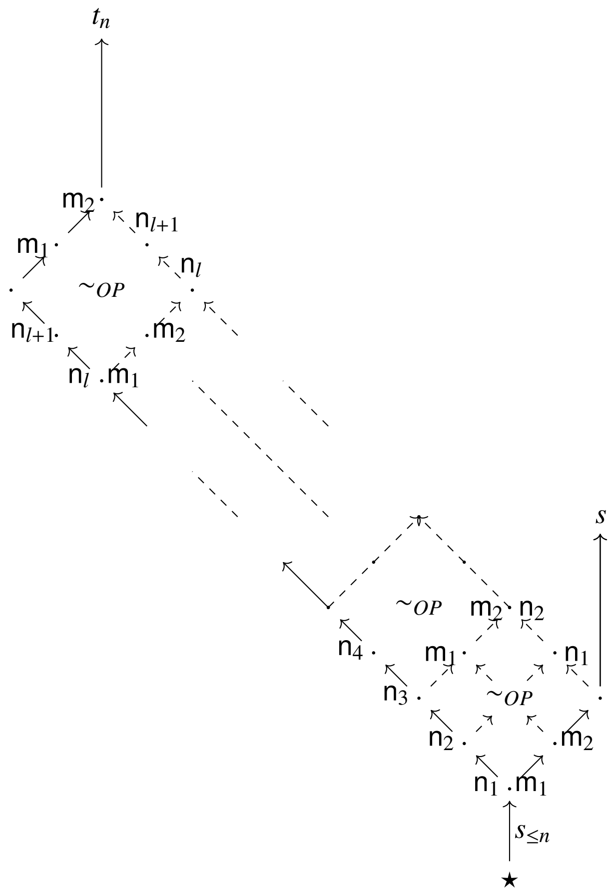
Lemma 5.9. *Let $s \in \sigma$, σ innocent strategy, and $s \sim_{OP} t$. Then $t \in \sigma$. Equivalently, an innocent strategy is closed under OP-homotopy.*

Proof. We do the proof by induction on the number of OP-homotopy steps needed to go from s to t. If it is 0, then $s = t$ and hence the property holds naturally. All we have to prove is that if s is in σ, and $s \xrightarrow{OP} s'$, where $\xrightarrow{OP}$ is a single OP-homotopy step, then s' is in σ.

Suppose $s = s_1.\mathsf{m}_1.\mathsf{n}_1.\mathsf{m}_2.\mathsf{n}_2.s_2$ and $s' = s_1.\mathsf{m}_2.\mathsf{n}_2.\mathsf{m}_1.\mathsf{n}_1.s_2$. Then by definition there is a tile $\mathsf{m}_1 \Uparrow \mathsf{m}_2$ and $\mathsf{n}_1 \Uparrow \mathsf{m}_2$, and using the backward consistency, we deduct that $s.\mathsf{m}_2.\mathsf{n}_2.\mathsf{m}_1.\mathsf{n}_1.s_2$ is in σ. □

We are now in position to prove the positionality of the strategy, that is, proposition 5.6.

Figure 5.1: Sequences of OP-homotopy steps following backward consistency



Proof. Let $s, s' \in \sigma$, $s \sim s' : \star \twoheadrightarrow x$, and $t : x \to y$, such that $s.t \in \sigma$. Then, $s' \leq s$ and hence, as both plays belong to the strategy, $s' \leq_{OP} s$. Furthermore, the length of the two plays being the same $s' \sim_{OP} s$. As this relation is closed under post-composition, $s'.t \sim_{OP} s.t$ and finally, as σ is stable under $\sim_{OP}$ homotopy $s'.t \in \sigma$. □

5.1.2 Structure of the innocent strategies: strong positionality

Innocent strategies satisfy a stronger property than positionality, they are relational. This means they are entirely determined by the set of positions they reach. This property allows us to see them simply as a subset of the set of positions. So a strategy $A \to B$ can alternatively be seen as a relation between some positions of A and some positions of B.

Given a subset $X \subseteq \mathsf{Legal}(A)$ we write $X^{\lightning}$ for the set of plays defined by X.

$$X^{\lightning} = \{s = \mathsf{m}_1.\mathsf{m}_2....\mathsf{m}_{n-1}.\mathsf{m}_n \in \mathsf{Legal}(\mathsf{Plays}((A)) \mid n \text{ even},$$
$$\text{and } \forall i \leq n.(i \text{ even and } \mathsf{m}_1.\mathsf{m}_2....\mathsf{m}_{i-1}.\mathsf{m}_i : \star \twoheadrightarrow x) \Rightarrow x \in X\}.$$

Definition 5.10. *A strategy is **strongly positional** (or relational) if* $\sigma = (\sigma^{\bullet})^{\lightning}$.

In other terms, a relational strategy is a strategy that is both static and dynamic. It can be characterised dynamically as a set of sequences, or statically as a set of positions.

Proposition 5.11. *Every innocent strategy is strongly positional.*

Proof. The inclusion $\sigma \subseteq (\sigma^\bullet)^\lightning$ is clear, so we only need to focus on the reverse inclusion, that we prove by induction on the length of the plays. We pick a play $s \in (\sigma^\bullet)^\lightning$. If the play is of length 0, then there is nothing to prove. So imagine the length of s is now $n+2$, and all the plays of length n of $(\sigma^\bullet)^\lightning$ have been proven to belong to σ. So s can be decomposed as $s = s_1.\mathsf{m}.\mathsf{n} : \star \twoheadrightarrow y$, such that $s_1 : \star \to x$, and as $x \in (\sigma^\bullet)$, together with s_1 is of length n, we already know that $s_1 \in \sigma$. As $y \in (\sigma^\bullet)$, we know that there is a path $t : \star \twoheadrightarrow y \in \sigma$. In particular as $s_1.\mathsf{m}.\mathsf{n}$ and t reach the same position, $s_1.\mathsf{m}.\mathsf{n} \sim t$ and $s_1 \leq t$. As s_1 and t are in σ, we can use proposition 5.8 and infer that $s_1 \leq_{OP} t$. In particular, there are m', n' two moves such that m' is an O-move, n' a P-move and $s_1.\mathsf{m}'.\mathsf{n}' \sim_{OP} t$. As $\mathsf{m}', \mathsf{n}', \mathsf{m}, \mathsf{n} \in t \setminus s_1$, we can gather that $\mathsf{m} = \mathsf{m}'$, $\mathsf{n} = \mathsf{n}'$ and $s_1.\mathsf{m}.\mathsf{n} \sim_{OP} t$, and thus $s = s_1.\mathsf{m}.\mathsf{n} \in \sigma$ as σ is closed under $\sim_{OP}$ homotopy. □

One can easily prove that relational, or strongly positional, entails positional.

Thanks to strong positionality, we can now characterise the strategies as sets of positions, and hence use it to prove compositionality, as well as the associativity of composition. In order to do that, we have to give a precise characterisation of those sets X such that there is a strategy σ that makes the following equality holds: $X = \sigma^\bullet$.

5.1.3 Innocent strategies as sets of positions

We give a characterisation of definable sets, that are, sets X that correspond to sets of positions reached by innocent strategies.

Definition 5.12. *A subset X of $\mathsf{Pos}(A)$ is **definable** if there exists a typed-coherent innocent strategy σ such that $\sigma^\bullet = X$.*

The description of definable sets is laid down in the following theorem. We say that a position y **dominated** by a set of positions X, if there is an $x \in X$, and a path $s : y \twoheadrightarrow x$. In this case we say that y is dominated by x (that is, $y \leq x$), or x (and X) **dominates** y. Similarly, we speak of under-domination in the case where $\exists x \in X, x \leq y$ (that is, there exists a path $s : x \twoheadrightarrow y$).

Theorem 5.13. *A set X of positions is definable if and only if:*

1. *(Root): $\bot \in X$.*
2. *(Legality): $X \subseteq \mathsf{Legal}(A)$.*
3. *(Nominal closure): X is nominal closed, and closed under strict typed substitutions. That is, $\forall x \in X, \forall y. y \cong x \Rightarrow y \in X$.*
4. *(Closure under intersection): $\forall x, y \in X, x \, C \, y \;\wedge\; x \uparrow y \;\Rightarrow x \sqcap y \in X$.*

5. *(Closure under union):* $\forall x, y \in X, x\, C_{\mathsf{post}}\, y \ \wedge \ x \uparrow y \ \Rightarrow x \sqcup y \in X$.
6. *(Preservation of compatibility): Let* $x \in X$, *and two moves* m, m' *such that* $x \xrightarrow{\mathsf{m}} y$ *and* $x \xrightarrow{\mathsf{m}'} y'$, *satisfying* $y \uparrow y'$, $y\, C_{\mathsf{post}}\, y'$ *and* y, y' *are dominated in* X. *Then* $y \sqcup y'$ *is dominated in* X.
7. *(Forward confluence 1): For all* $x \in X$, *if there is an opponent move* $\mathsf{m} : x \to y$ *and* y *is dominated in* X *then there is a unique* $z \in X$, *up to equivalence, such that there is a P-move* n *satisfying* $x \overset{\mathsf{m.n}}{\twoheadrightarrow} z$, *and furthermore* $\nu_T(\mathsf{n}) \subseteq \nu_T(y)$.
8. *(Forward confluence 2) For all* $x \in X$, *if there is an opponent move* $\mathsf{m} : x \to y$ *and* y *is dominated in* X *by* w *then there is a unique P-move* n, *such that* $x \overset{\mathsf{m.n}}{\twoheadrightarrow} z$, *and* $z \in X$, $z \leq w$, *and furthermore* $\nu_T(\mathsf{n}) \subseteq \nu_T(y)$.
9. *(Mutual attraction) : For all* x, y *in* X *such that* y *dominates* x, *either* $x = y$ *or there is an opponent move* $\mathsf{m} : x \to x'$ *and a player move* $\mathsf{n} : y' \to y$ *such that* y' *dominates* x'.

To do the proof we rely on an additional property, namely reverse consistency, presented below. First, note that the dialogues games we have defined are **intuitionistic** in the following sense:

- For every occurrence of move $\mathsf{n} : y \to z$, there is at most one move m that justifies it.
- When there is one, $\lambda(\mathsf{m}) = -\lambda(\mathsf{n})$.

Consequently, a strategy which is backward and forward consistent satisfies automatically an additive property called reverse consistency, and described below.

Definition 5.14. *A strategy is* ***reverse consistent*** *if for all* $s.\mathsf{m}_1.\mathsf{n}_1.\mathsf{m}_2.\mathsf{n}_2.t \in \sigma$, *if* $|s|$ *is even and* $\mathsf{n}_1 \Uparrow \mathsf{m}_2, \mathsf{n}_2$ *then* $\mathsf{m}_1 \Uparrow \mathsf{m}_2, \mathsf{n}_2$, *and* $s.\mathsf{m}_2.\mathsf{n}_2.\mathsf{m}_1.\mathsf{n}_1.t \in \sigma$.

This is displayed diagrammatically in figure 5.2.

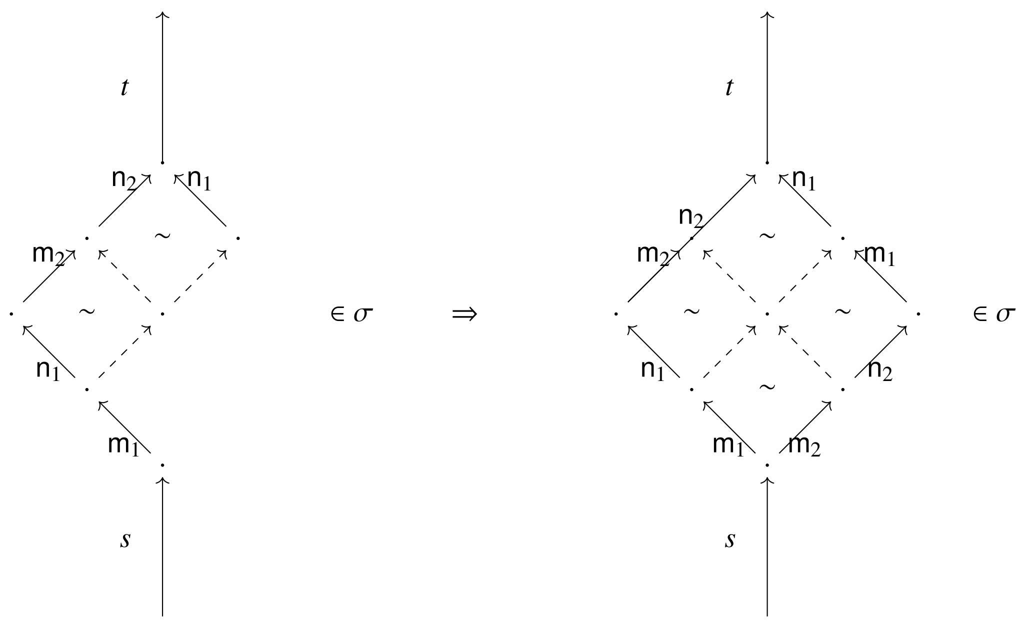


Figure 5.2: Reverse innocence

Proposition 5.15. *Every innocent strategy is reverse consistent.*

Proof. Let $s.\mathsf{m}_1.\mathsf{n}_1.\mathsf{m}_2.\mathsf{n}_2.t$ as in the definition. From prefix closure of σ, we devise that $s.\mathsf{m}_1.\mathsf{n}_1.\mathsf{m}_2.\mathsf{n}_2$ is part of the strategy. We need to prove that $\mathsf{m}_2 \Uparrow \mathsf{m}_1$. We already know that $\mathsf{m}_2 \Uparrow \mathsf{n}_1$. As the game is intuitionist, an opponent move cannot justify another one. That is $\neg(\mathsf{m}_1 \vdash \mathsf{m}_2)$. So the only possible chain of moves between m_1 and m_2 would be $\mathsf{m}_1 \vdash \mathsf{n}_1 \vdash \mathsf{m}_2$. But as $\neg(\mathsf{n}_1 \vdash \mathsf{m}_2)$ it entails $\neg(\mathsf{m}_1 \leq \mathsf{m}_2)$. On the other hand, $\neg(\mathsf{m}_2 \leq \mathsf{m}_1)$ as m_2 appears after m_1 in the play, and $\neg(\mathsf{m}_2 \smile \mathsf{m}_1)$, $\mathsf{m}_2\ C_{\mathsf{post}}\ \mathsf{m}_1$. So, $\mathsf{m}_1 \Uparrow \mathsf{m}_2$. By backward consistency, $s.\mathsf{m}_2.\mathsf{n}_2.\mathsf{m}_1.\mathsf{n}_1$ belongs to σ and by positionality, $s.\mathsf{m}_2.\mathsf{n}_2.\mathsf{m}_1.\mathsf{n}_1.t \in \sigma$ as required. □

We now prove the theorem.

Proof. We first show that there is an innocent strategy σ such that $X = \sigma^\bullet$. We then show that given any innocent strategy τ, its set $\tau^\bullet$ satisfies the properties of a definable set.

We start by showing that there is a set of plays σ, ($\sigma = X^{\lightning}$), such that $X = \sigma^\bullet$. Formally, we need to prove that $(X^{\lightning})^\bullet = X$. We will prove that σ is an innocent strategy afterwards. By definition, $(X^{\lightning})^\bullet \subseteq X$, as the target of every play of $X^{\lightning}$ is an element of X. So it remains to prove the reverse inclusion. Let us prove that every element of X is the target of an alternating path, such that each even-length subpath reaches an element of X. This is true for the root by (1). Let $x \in X$, and let us suppose that we proved the property for every element y of X with $y \leq x$, and we note that, as $\bot \in X$, the set of such y is never empty. Then let y be maximal in X under x. We apply the last property, mutual attraction (9), and hence know that there in an opponent move $\mathsf{m} : y \to y'$ and a player move $\mathsf{n} : x' \to x$ such that x' dominates y'. Now either $x' = y'$, in which case we conclude using the induction hypothesis. Or, using forward confluence (2), there is a player move $\mathsf{n}' : x' \to y''$, and $y'' < x$ is in X. This brings a contradiction as y was supposed to be maximal in X under x.

In this paragraph, we prove that the set of plays $X^{\lightning}$ forms a typed-coherent strategy. As the root is an element of X, the empty play is part of $X^{\lightning}$, as expected. Moreover, thanks to the definition of $\lightning$, the closure under prefix follows. The closure under nominal permutations comes from the fact that X is itself closed under permutations, and similarly for typed substitutions. The remaining property is nominal determinacy. Let $s : \star \overset{s}{\twoheadrightarrow} x, s' : \star \overset{s'}{\twoheadrightarrow} x'$ two plays of $X^{\lightning}$ such that $s \simeq s'$, and hence $x \simeq x'$. Let $\mathsf{m} : x \xrightarrow{\mathsf{m}} y$, $\mathsf{m}' : x' \xrightarrow{\mathsf{m}'} y'$ two opponent moves such that $s.\mathsf{m} \simeq s'.\mathsf{m}'$, and thus $y \simeq y'$. Let $\mathsf{n} : y \to z$, $\mathsf{n}' : y' \to z'$ such that $s.\mathsf{m}.\mathsf{n}$ (resp $s'.\mathsf{m}'.\mathsf{n}'$) $\in X^{\lightning}$. We must prove that $z \simeq z'$. Let us consider π a nominal permutation such that $\pi \cdot s'.\mathsf{m}' = s.\mathsf{m}$. Then by forward consistency (1) there exists a unique, up to nominal permutation, z'', such that there exists a P-move n'' satisfying $s.\mathsf{m}.\mathsf{n}'' : \star \twoheadrightarrow z'' \in X$. In other terms, $\pi \cdot z' \simeq z'' \simeq z$ and therefore $z \simeq z''$, that is, $s.\mathsf{m}'.\mathsf{n}' \simeq s.\mathsf{m}.\mathsf{n}$, and the strategy is nominal deterministic. Furthermore, $\nu(\mathsf{n}') \subseteq \nu(s.\mathsf{m})$ entails that the strategy is semi-linear, and hence, typed-coherent.

This leads us to prove that the strategy $X^{\lightning}$ satisfies the 2 diagrams of innocence. We start with forward consistency.

Let $x \in X$, and $\mathsf{m}_1 : x \to y_1$ an opponent move, such that y_1 is dominated in X. We also pick $\mathsf{m}_2 : x \to y_2$ such that $\mathsf{m}_2 \uparrow \mathsf{m}_1$, $\mathsf{m}_2 \neq \mathsf{m}_1$, y_2 is dominated in X and $y_2\ C_{\mathsf{post}}\ y_1$. As y_1 and y_2 are compatible and dominated in X, so is their union $y_1 \sqcup y_2$ (by (6)). So let us call w an element of X such that w dominates $y_1 \sqcup y_2$ and w minimal among those. Then y_1 is dominated by $\{w\}$, and therefore there is a single move n_1 such that $y_1 \xrightarrow{\mathsf{n}_1} z_1$ and z_1 is dominated by w. Similarly, we can conclude that there is a single move $\mathsf{n}_2 : y \to z_2$, such that $X \ni z_2 \leq w$. Furthermore, z_1 is dominated by w, and so is $z_1 \uplus \{\mathsf{m}_2\}$, and similarly for $z_1 \uplus \{\mathsf{m}_1\}$. Therefore, we conclude the existence of two additional moves $\mathsf{n}_2', \mathsf{n}_1'$ such that $z_1 \uplus \{\mathsf{m}_2\} \uplus \{\mathsf{n}_2'\}, z_2 \uplus \{\mathsf{m}_1\} \uplus \{\mathsf{n}_1'\}$ both lead to positions in X dominated by w. As these positions are both dominated by w, there are compatible, and, their intersection belong in X. We deal with different cases:

- if $\mathsf{n}_1' \neq \mathsf{n}_1, \mathsf{n}_2' \neq \mathsf{n}_2$ then their intersection is $y_1 \sqcup y_2$, which cannot be in X since it is an unbalanced position.
- If $\mathsf{n}_1' = \mathsf{n}_1, \mathsf{n}_2' \neq \mathsf{n}_2$, then their intersection is $y_1 \sqcup y_2 \uplus \{\mathsf{n}_1\}$, which cannot be in X for the same reason as above.
- If $\mathsf{n}_1 \neq \mathsf{n}_1, \mathsf{n}_2' = \mathsf{n}_2$, then this is a similar case as above.
- If $\mathsf{n}_1 = \mathsf{n}_1' = \mathsf{n}_2 = \mathsf{n}_2'$, then similar case as above once again.

So we conclude that $\mathsf{n}_1 = \mathsf{n}_1', \mathsf{n}_2 = \mathsf{n}_2', \mathsf{n}_1 \neq \mathsf{n}_2$, and furthermore $\mathsf{n}_1 \Uparrow \mathsf{n}_2$, since they appear at different orders in different plays. At this stage we have not finished the proof since the $\mathsf{n}_1, \mathsf{n}_2$ were picked in accordance to w, and we shall prove that the proof can be made to work for any $\mathsf{o}_1, \mathsf{o}_2$ chosen such that $s.\mathsf{m}_1.\mathsf{o}_1\ C_{\mathsf{post}}\ s.\mathsf{m}_2.\mathsf{o}_2$, and $s.\mathsf{m}_1.\mathsf{o}_1, s.\mathsf{m}_2, \mathsf{o}_2$ reach positions in X. By nominal determinacy, we automatically have $s.\mathsf{m}_1.\mathsf{o}_1 \simeq s.\mathsf{m}_1.\mathsf{n}_1$, and similarly for $\mathsf{o}_2, \mathsf{n}_2$. We first do the proof in the "easy" case, where $\mathsf{o}_1 \#_{\mathsf{cells}} \mathsf{n}_1, \mathsf{n}_2$ and $\mathsf{o}_2 \#_{\mathsf{cells}} \mathsf{n}_1, \mathsf{n}_2$. Note that $\mathsf{o}_1 \#_{\mathsf{cells}} s.\mathsf{m}_2$ as enforced by the relation C_{post}. In this case, we pick $\pi_1, \pi_2 \in \mathsf{Perm}(\mathbb{A}_{\mathsf{cells}})$ of minimal support such that $\pi_1 s.\mathsf{m}_1.\mathsf{n}_1 = s.\mathsf{m}_1.\mathsf{o}_1$, and similarly for o_2. In particular, as $\nu(\pi_1) \subseteq \nu(\mathsf{n}_1) \cup \nu(\mathsf{o}_1)$, and similarly for π_2, we got $\pi_1 \# \pi_2$. We then have $s.\mathsf{m}_1.\mathsf{o}_1 \leq \pi_1 \cdot w$, and $\pi_2(s.\mathsf{m}_1.\mathsf{o}_1) = s.\mathsf{m}_1.\mathsf{o}_1 \leq (\pi_2 \circ \pi_1) \cdot w$. Similarly, $s.\mathsf{m}_2.\mathsf{o}_2 \leq (\pi_2 \circ \pi_1) \cdot w$. Notably, as π_1, π_2 have disjoint support, $(\pi_1 \circ \pi_2) = (\pi_2 \circ \pi_1)$, and we have found a common bound for $s.\mathsf{m}_1.\mathsf{o}_1, s.\mathsf{m}_2.\mathsf{o}_2$, namely $(\pi_1 \circ \pi_2) \cdot w$. We can then apply the proof as above. To finish the proof, we need to tackle the "hard" case, where the propositions $\mathsf{o}_1 \#_{\mathsf{cells}} \mathsf{n}_1, \mathsf{n}_2$, $\mathsf{o}_2 \#_{\mathsf{cells}} \mathsf{n}_1, \mathsf{n}_2$ might not be true. In that case, we use intermediate $\mathsf{p}_1 \simeq \mathsf{n}_1$, $\mathsf{p}_2, \simeq \mathsf{n}_2$ such that $\mathsf{p}_1, \mathsf{p}_2 \# \mathsf{o}_1, \mathsf{o}_2, \mathsf{n}_1, \mathsf{n}_2$, $s.\mathsf{m}_1.\mathsf{p}_1, s.\mathsf{m}_2.\mathsf{p}_2$ reach positions in X, and $s.\mathsf{m}_1.\mathsf{p}_1\ C_{\mathsf{post}}\ s.\mathsf{m}_2.\mathsf{p}_2$. Relying on the p_i, we can apply the easy case, and obtain an upper bound $w' \simeq w$ for the positions reached by $s.\mathsf{m}_1.\mathsf{p}_1, s.\mathsf{m}_2.\mathsf{p}_2$. Next, we can go from p_i to o_i, noticing that this corresponds once again to the easy case. Eventually, we reach a $w'' \simeq w' \simeq w$ such that $w'' \in X$, and w'' is an upper bound for $s.\mathsf{m}_1.\mathsf{o}_1, s.\mathsf{m}_2.\mathsf{o}_2$. This allows us to deduce the proposition as above.

We prove the backward consistency. Let $s.\mathsf{m}_1.\mathsf{n}_1.\mathsf{m}_2.\mathsf{n}_2.t : \star \twoheadrightarrow w \in X^{\lightning}$ such that $\star \overset{s}{\twoheadrightarrow} x \overset{\mathsf{m}_1.\mathsf{n}_1}{\twoheadrightarrow} y \overset{\mathsf{m}_2.\mathsf{n}_2}{\twoheadrightarrow} z \overset{t}{\twoheadrightarrow} w$, and suppose that $s.\mathsf{m}_2$ is legal, and $\mathsf{m}_2 \Uparrow \mathsf{m}_1, \mathsf{n}_1$. Furthermore, $s.\mathsf{m}_2$ is dominated by $z \in X$. Then let n' be the move such that $s.\mathsf{m}_2.\mathsf{n}' \in X^{\lightning}$ and $s.\mathsf{m}_2.\mathsf{n}' : \star \twoheadrightarrow y'$ is dominated by z. Therefore, $\mathsf{n}' = \mathsf{n}_1$ or $\mathsf{n}' = \mathsf{n}_2$. Note that $y' \uparrow y$ since they are both dominated by z. If $\mathsf{n}' = \mathsf{n}_1$, then as X is closed under compatible intersection, $y' \sqcap y$ is in X. However, in that case, $y \sqcap y' = x \uplus \{\mathsf{n}_1\}$, which is a unbalanced position. So $\mathsf{n}' = \mathsf{n}_2$, and $s.\mathsf{m}_2.\mathsf{n}_2 \in X^{\lightning}$. Finally, we devise that $s.\mathsf{m}_2.\mathsf{n}_2.\mathsf{m}_1.\mathsf{n}_1$ is in $X^{\lightning}$ by forward consistency. Finally, noticing that $s.\mathsf{m}_2.\mathsf{n}_2.\mathsf{m}_1.\mathsf{n}_1$ and $s.\mathsf{m}1.\mathsf{n}_1.\mathsf{m}_2.\mathsf{n}_2$ reach the same position, we can conclude that $s.\mathsf{m}_2.\mathsf{n}_2.\mathsf{m}_1.\mathsf{n}_1.t \in X^{\lightning}$ by definition of $\lightning$, which concludes the proof.

At last, reverse consistency follows from the proposition 5.15

Subsequently, we now have to prove the reverse direction. Let σ be an innocent strategy, and consider the set $X = \sigma^{\bullet}$. The condition (1) ($\perp \in X$) comes from the fact that the empty play is always part of a strategy. The second condition (2) ($X \subseteq \mathsf{Legal}(A)$) is by definition just as the condition (3): its nominal and typed substitutions closure follows directly from the one of σ. Harder to prove are the conditions of closure under intersection (4) and union (5). We start with the first.

Let $x, y \in X$, such that $x \uparrow y$ (and hence $x\ C\ y$), and let $s : \star \twoheadrightarrow x \in \sigma$, and $t : \star \twoheadrightarrow y \in \sigma$. Consider $z = x \sqcap y$, our goal is to show that there is $u : \star \twoheadrightarrow z$ such that $u \preceq_{OP} s, t$. As σ is closed under $\preceq_{OP}$, this will prove that $u \in \sigma$ and hence $z \in \sigma^{\bullet}$. We call M the set of moves that s, t have in common. M is the set of moves that appear in $x \sqcap y$. As both s, t are plays, and hence closed under down-closure, so is M. Therefore, the restriction of s to the moves that appear in M define a play (see proposition 4.56), in the sense that is defines a path starting from the root. We call it u. We prove that $u \preceq_{OP} s, t$, by induction on the pre-sequences v of u of even length. That is, we prove that for every pre-sequence of even length v of u, $v \preceq_{OP} s, t$, and furthermore u is of even-length. Firstly, we notice that the empty sequence ϵ obviously satisfies the property. For the next step, consider it true for an arbitrary pre-sequence of even length v of s'. Then let m the move after v in s', and suppose that m is an O-move. As $v \preceq_{OP} s$, $\exists \mathsf{n}_1.\ v.\mathsf{m}.\mathsf{n}_1 \preceq_{OP} s$, and, equivalently, $\exists \mathsf{n}_2.\ v.\mathsf{m}.\mathsf{n}_2 \preceq_{OP} t$. Now, by nominal determinacy, $\mathsf{n}_1 \simeq \mathsf{n}_2$, and as x, y are compatible for equality, $\mathsf{n}_1 = \mathsf{n}_2$. Hence $v.\mathsf{m}.\mathsf{n}_1$ is a prefix of u and $v.\mathsf{m}.\mathsf{n}_1 \preceq_{OP} s, t$. On the other hand, let us consider that the next move n after v in s is a P-move. Then there exists m_1, s_1, t_1 such that $v.s_1.\mathsf{m}_1.\mathsf{n} \preceq_{OP} s$, $v.t_1.\mathsf{m}_2.\mathsf{n} \preceq_{OP} t$ and $\mathsf{m}_1 \neq \mathsf{m}_2$. Now, we apply a permutation π of $\mathbb{A}_{\mathsf{cells}}$ to x such that $\pi \cdot x\ C_{\mathsf{post}}\ y$. Let us note that this let $x \sqcap y$ (that is, $x \sqcap y = \pi \cdot x \sqcap y$), invariant, and hence we assume without lost of generality that $x\ C_{\mathsf{post}}\ y$. As $x \uparrow y$, $x\ C\ y$, and $t_1 \cap s_1 = \emptyset$ (in the sense that they have no common moves), we can apply forward consistency, and push s_1 along t_1, obtaining a play $v.s_1.t_1 \in \sigma$. At this stage, we can perform a final step of forward consistency on $v.s_1.t_1.\mathsf{m}_1.\mathsf{n}$ and $v.s_1.t_1.\mathsf{m}_2.\mathsf{n}$, relying on $\mathsf{m}_1 \uparrow \mathsf{m}_2$, $\mathsf{m}_1 \neq \mathsf{m}_2$, $\mathsf{m}_1\ C_{\mathsf{post}}\ \mathsf{m}_2$. This one tells us that the after m_1 is non-congruent to the move after m_2, that is $\mathsf{n} \not\cong \mathsf{n}$. Hence this is a contradiction, and this case is excluded. As a result, for all pre-sequences (of even-length) v of u, $v \preceq_{OP} s$, and y is of even-length. Consequently, $u \preceq_{OP} s$ and $u \in \sigma$. As u reaches $x \sqcap y$, this entails $x \sqcap y \in \sigma^{\bullet}$.

Let x, y be two post-compatible positions in X and s, t two plays of σ that target them. Let u be a play of σ that targets $x \sqcap y$. We write $s \sim_{OP} u.s'$ and $t \sim_{OP} u.t'$ and do the induction on the length of s'. If it is null, then $x \leq y$ and $t : \star \twoheadrightarrow x \sqcup y = y$ is the witness play. Otherwise, consider $u.\mathsf{m}.\mathsf{n}.s'' \sim_{OP} s$. Then m is independent of t' as it is not in conflict, not related by the partial order with the moves of t' and post-compatible with t'. Therefore, by pushing the forward consistency diagram of innocence, $u.\mathsf{m}.\mathsf{n}.t' \in \sigma$. Then let z such that $\star \overset{u.\mathsf{m}.\mathsf{n}.t'}{\twoheadrightarrow} z$. It is easily seen that $z \geq y$, and $z \sqcup x \leq y \sqcup x$. As a result, $z \sqcup x = y \sqcup x$, and we can work with the new play $t'' = u.\mathsf{m}.\mathsf{n}.t'$. Thus, we can intersect z and x, and consider the play $u' = u.\mathsf{m}.\mathsf{n}$ that targets this position in σ. However, this time, we consider s'' such that $u'.s'' \sim_{OP} s$ and, as s'' is two moves shorter than s', we apply the induction hypothesis on it.

The sixth condition, preservation of compatibility, is a simple consequence of the forward consistency of $\sigma^\bullet$. Let x a position reached by σ (by a witness play s), and two compatible opponent moves m, m' played from σ, $x \xrightarrow{\mathsf{m}} y$, $x \xrightarrow{\mathsf{m}'} y'$, $y\ C_{\mathsf{post}}\ y'$, such that $s.\mathsf{m}.\mathsf{n}, s.\mathsf{m}'.\mathsf{n}' \in \sigma$. Then n, n' can be chosen such that $\mathsf{n}\ C_{\mathsf{post}}\ \mathsf{n}'$, entailing $\mathsf{n} \Uparrow \mathsf{n}'$. By forward consistency, $s.\mathsf{m}.\mathsf{n}.\mathsf{m}'.\mathsf{n}' \in \sigma$ and reaches a position that dominates $y \sqcup y' = x \oplus \{\mathsf{m}, \mathsf{m}'\}$.

Remaining is to prove the forward confluence properties (7)(8) of X. Let x be a position of the strategy, $x \in \sigma^\bullet$, and let m be an O-move such that $x \xrightarrow{\mathsf{m}} y \in \mathsf{Legal}(A)$, and let $w \in \sigma^\bullet$ that dominates y. Then, by definition, there is a path $t : \star \twoheadrightarrow w \in \sigma$. Let $\sigma \ni s : \star \twoheadrightarrow x$. Then as $x \leq w$, $s \leq t$ and by proposition 5.8 $s \preceq_{OP} t$. So we write $t \sim_{OP} s.t_2$, and we know that $\mathsf{m} \in t_2$. Hence there is a unique n such that $s.\mathsf{m}.\mathsf{n} \preceq_{OP} s.t_2$. So there exists a player move n and a position z such that $\sigma^\bullet \ni z = x \oplus \{\mathsf{m}, \mathsf{n}\}$, and $z \leq w$. Furthermore, if there were two, then by nominal determinacy there would be equivalent, and as both dominated by w, there would be compatible for equality, and hence, equal. So the n, z are unique. Finally, (7) follows straightforwardly from the nominal determinacy of strategies.

We finish with the last property (9) of mutual attraction. Let $x, y \in \sigma^\bullet$, such that $x \leq y$. Then let us prove that there exists a path $x \twoheadrightarrow y$ such that s begins with an O-move. To do that, let consider a path $s : \star \twoheadrightarrow y \in \sigma$, and let $t : \star \twoheadrightarrow x \in \sigma$. As $x \leq y$ then $t \leq s$. Then by proposition 5.8, $t \preceq_{OP} s$, and $s \sim_{OP} t.s' \in \sigma$. Hence, $s' : x \twoheadrightarrow y$ is either empty, or starts with an O-move and finishes with a P-move, which finishes the last case of this proof. □

5.1.3.1 On backward confluence

Definable sets for innocent strategies defined originally in [64], satisfied an additional property, namely backward confluence, that we present below:

- For all $x \in X$, if there is a P-move $\mathsf{n} : y \xrightarrow{\mathsf{n}} x$, and y is under-dominated by $w \in X$, then there is a unique O-move m such that $z \xrightarrow{\mathsf{m}} y$, $z \in X$ and furthermore $w \leq z$.

This property is still satisfied by our strategies.

Proposition 5.16. *Every innocent strategy is backward confluent.*

Proof. Let $x, w \in \sigma^\bullet$, such that $w \leq x$. Let n a P-move, $\mathsf{n} : y \to x$ such that $w \leq y$. We must prove that there exists an unique m O-move, $\mathsf{m} : z \to y$ such that $z \in \sigma^\bullet$ and furthermore w

under-dominates z. We start by proving that there exists a $s \in \sigma$, $s : \star \twoheadrightarrow x$ and the last move of s is n. Let us pick a random $s : \star \twoheadrightarrow x$ in σ, then $s = s'.\mathsf{m}.\mathsf{n}.s''$. If s'' is empty, the proof is over. Otherwise as for any pairs of OP-moves $\mathsf{m}'.\mathsf{n}' \in s''$ we have $\mathsf{m}', \mathsf{n}' \Uparrow \mathsf{n}$, by reverse consistency we deduct that $\mathsf{m}', \mathsf{n}' \Uparrow \mathsf{m}$, and we can push the path s'' down $\mathsf{m}.\mathsf{n}$. Finally, we get $s'.s''.\mathsf{m}.\mathsf{n} \in \sigma$. Furthermore, consider $\sigma \ni t : \star \twoheadrightarrow w$, then as $w \leq x$, $t \leq s$. As both s, t belongs in σ, it entails $t \preceq_{OP} s$. As $\mathsf{n} \notin t$, this leads to $\mathsf{m} \notin t$, and $t \preceq_{OP} s.s''$. This is equivalent to $w \leq z$. We conclude by proving the uniqueness. It they were two possible m, call them m, m', then, as they reach the same position, there exists two plays $\sigma \ni s.\mathsf{m}.\mathsf{n} \sim t.\mathsf{m}'.\mathsf{n} \in \sigma$. Therefore, $s.\mathsf{m}.\mathsf{n} \sim_{OP} t.\mathsf{m}'.\mathsf{n}$ and as a result, $\mathsf{m} = \mathsf{m}'$ (as $\mathsf{n} = \mathsf{n}$). □

Alternatively, we provide a direct proof in the appendix 9.3 that the properties of definable sets entail backward confluence.

5.1.4 Innocent strategies and weak sequentiality structures

Sequentiality structures have been introduced in [69], as a way to strengthen innocent strategies, and most of the properties-definition presented here are directly adapted from this paper. These consists of a set of total functions, relating negative and positive cells. We here introduce weak sequentiality structures, that corresponds to these "partial functions" that an innocent strategy naturally produce. Those will be pivotal in our proof that sequential and relational composition are equivalent.

We say that a cell α **justifies** a move m, written $\alpha \vdash \mathsf{m}$, if $\mathsf{m} = (\alpha, v, S)$. We say that a cell α is **accessible** from a position $x \in \mathsf{Legal}(A)$ in a dialogue game if:

$$\exists \mathsf{m} \in x. \mathsf{m} = (\beta, v, S), \alpha \in S, \text{ and } \neg(\exists \mathsf{m}' \in x. \mathsf{m}' = (\alpha, v', S')) .$$

In other terms, the cell α justifies no move in the position. Recalling the fact that the dialogue game is almost the syntactical tree of the formula, the accessible cells correspond to those sub-formulas that are yet to be explored by the position. We denote A_x the set of accessible cells of the position x. We divide A_x into two subsets, A_x^+ the subset of A_x of cells of positive polarity and A_x^- those of negative polarity. Those of negative polarity are brought by a move of positive polarity and justify moves of negative polarities, and inversely for those of negative polarities.

Definition 5.17. *Let σ be an innocent strategy, and $x \in \sigma^\bullet$. Let $\alpha, \beta \in A_x^-$ two different opponent cells. Let $\sigma \restriction_x \alpha$ be defined as follows:*

$$\sigma \restriction_x \alpha = \{s : x \twoheadrightarrow y \mid \exists s' : \star \twoheadrightarrow x.\ s'.s \in \sigma \wedge \forall \mathsf{m} \in s. \neg(\exists \alpha' \in A_x^- \setminus \{\alpha\}. \alpha' \vdash \mathsf{m})\}$$

*$\sigma \restriction_x \alpha$ **is the restriction of σ above α.***

$\sigma \mid_x \alpha$ is the part of the strategy above x that corresponds to a trigger by opponent of the cell α. Given a move m, we say that $\mathsf{m} \in \sigma \mid_x \alpha$, if there is a sequence $s \in \sigma \mid_x \alpha$ such that m is a move of s.

Lemma 5.18. *The following property holds:*

$$\mathsf{m} \in \sigma \mid_x \alpha \;\Rightarrow\; \forall \mathsf{n} \in \sigma \mid_x \beta . \mathsf{m} \not\cong \mathsf{n}.$$

This lemma is quite a strong property, also called the "separation of contexts". It says that the strategy above two different cells explores two distinct sub-trees.

Proof. We consider a move ϖ of $\sigma \mid_x \alpha$ (we use a Greek letter to distinguish it from the other moves), and let $s \in \sigma \mid_x \alpha$ such that $\varpi \in s$. Now, let t a sequence of $\sigma \mid_x \beta$, with $\beta \neq \alpha$. As we could consider $s' \cong s, t' \cong t$ such that $s' \; C_{\mathsf{post}} \; t$, we assume without lost of generality that $s \; C_{\mathsf{post}} \; t$. In that case $\varpi \cong \mathsf{n} \Rightarrow \varpi = \mathsf{n}$. We will prove that for all moves $\mathsf{m} \in s, \mathsf{n} \in t . \mathsf{m} \Uparrow \mathsf{n}$. As the first move m_1 of s is justified by α, and the first one n_1 of t by β, we already know that $\mathsf{m}_1 \Uparrow \mathsf{n}_1$. By forward consistency $\forall i, j \leq 2 . \mathsf{m}_i \Uparrow \mathsf{n}_j$. So we proved the property for the paths of length 2. The proof is done by induction on the lengths of the paths s and t.

We consider that the length of s is $n+2$, t is of length m, and that we proved the property hold for (n, m). We write s' for the subpath of s that consists of its n^{th} first moves. By using the forward consistent diagram of innocence and the inductive hypothesis, we can push the path t along the s'. We now need to prove that $\mathsf{m}_{n+1} \Uparrow \mathsf{n}$ for every move n of t. As m_{n+1} is an opponent move, its departure cell is brought by a player move. By definition of s, its departure cell cannot be a cell available at x. Hence it is brought up by a player move in s'. As s', t are independent (in the sense that $\forall \mathsf{m} \in s, \mathsf{n} \in t' . \mathsf{m} \Uparrow \mathsf{n}$), the move m_{n+1} is not related by the partial order, not in conflict with the moves of t, and is post-compatible. Therefore, $\mathsf{m}_{m+1} \Uparrow t$ (meaning it is strongly compatible with every move of t). Hence, by repetitive applications of forward consistency $\mathsf{n}_{n+1} \Uparrow t$, finishing the inductive case.

If t is of length $m+2$, the induction is strongly similar, as the role being played by s, t are interchangeable. □

There is a direct corollary to this lemma, that highlights why we speak about separation of contexts.

Corollary 5.19. *Let x a position of $\sigma^\bullet$, and $\alpha \in A_x^-$. Then there is a set, called* $\mathsf{dominion}_x(\alpha) \subseteq A_x^+$ *defined by:*

$$\mathsf{dominion}_x(\alpha) = \{\beta \in A_x^+ \mid \exists \mathsf{m} \in \sigma \mid_x \alpha . \beta \vdash \mathsf{m}\}$$

such that $\sigma \mid_x \alpha$ takes place above $\alpha \sqcup \mathsf{dominion}_x(\alpha)$ and such that $\alpha \neq \beta \Rightarrow \mathsf{dominion}_x(\alpha) \cap \mathsf{dominion}_x(\beta) = \emptyset$.

This corollary enables the following definition.

Definition 5.20. *Given an innocent strategy σ, there exists a family partial function $\{\phi_x : A^+_x \to A^-_x \mid x \in \sigma^\bullet\}$, called* ***weak sequentiality structure****, such that:*

$$\phi_x(\alpha) = \beta \Leftrightarrow \exists \mathsf{m} \in \sigma \restriction_x \alpha, \beta \vdash \mathsf{m}.$$

A function ϕ_x is partial as it is undefined on cells without move above (that is, cells corresponding to $\neg 0$), or even on cells above whom the strategy does not explore. Given a cell $\alpha \in A^-_x$, we say that $\mathsf{dominion}(\alpha)$ is the context **captured** by α. Using the correspondence between untyped cells and formulas of tensorial logic, this corollary is the game semantics counterpart of the following (wrong) logical rule:

$$\frac{\Gamma_1 \vdash A \qquad \Gamma_2 \vdash B}{\Gamma_1, \Gamma_2, \Gamma_3 \vdash A \otimes B}.$$

Indeed, the formulas on the right hand side of the sequents correspond to negative cells, the ones on the right to positive ones. Now, the corollary tells us that the strategy splits in two independent parts, depending on the cell (formula) chosen by opponent. However, as one can clearly notice, the main problem is that the context is *affine*, there might be some parts of the context (that is some positive cells), that might not be explored. This will be targeted in Section 5.3.2.

We introduce an invariant of the sequentiality structure.

Lemma 5.21. *Let $s : \star \twoheadrightarrow x \overset{t}{\twoheadrightarrow} y \in \sigma$, and $\alpha \in A^-_x \cap A^-_y$. Then $\mathsf{dominion}_x(\alpha) = \mathsf{dominion}_y(\alpha)$,*

In other terms, let $x, y \in \sigma^\bullet$ such that $x \leq y$. Then if $\alpha \in A^-_x \cap A^-_y$, it holds that $\mathsf{dominion}_x(\alpha) = \mathsf{dominion}_y(\alpha)$.

Proof. We do the proof in the case where $t = \mathsf{m.n}$. This proof generalises straight away in the general case.

Consider the set of plays s that belongs to $\sigma \restriction_x \alpha$. As α is an accessible cell of both x and y, it means that the play $\mathsf{m.n}$ belongs to $\sigma \restriction_x \beta$ for $\beta \neq \alpha$. We proved above (in the proof of 5.18) that this entails $\mathsf{m.n} \Uparrow \sigma \restriction_x \alpha$. Consequently, the plays of $\sigma \restriction_x \alpha$ can be pushed above $\mathsf{m.n}$ (assuming they satisfy the necessary conditions of C_{post}), and $\sigma \restriction_y \alpha = \{s \in \sigma \restriction_x \alpha \mid s \; C_{\mathsf{post}} \; y\}$. Given a path $s \in \sigma \restriction_x \alpha$, there exists a path s' such that $s' \simeq s$, $s' \; C_{\mathsf{post}} \; y$ and $s' \in \sigma \restriction_x \alpha$. Consequently, if a cell of x justifies a move in $\sigma \restriction_x \alpha$, it also justifies one in $\sigma \restriction_y \alpha$. □

One of the key points of sequentiality structures is this simple property, that states that if $\phi(\alpha) = \beta$, then β has appeared after α in the sequence. This will turn to be a central point to later prove compositionality.

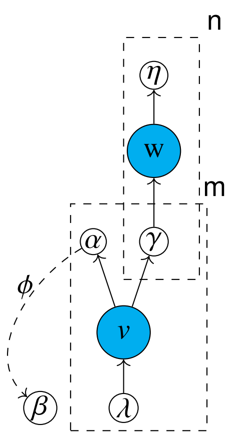
Definition 5.22. *Let σ be an innocent strategy, $s : \star \twoheadrightarrow x \in \sigma$, and let $\alpha \in A_x$. We define $\|s\|_\alpha$ as the length of the minimal prefix $s' : \star \twoheadrightarrow y$ of s such that $\alpha \in A_y$.*

Proposition 5.23. *Let σ be an innocent strategy, $s : \star \twoheadrightarrow x \in \sigma$ and $\alpha, \beta \in A_x$ such that $\alpha \in \mathsf{dominion}(\beta)$. Then $\|s\|_\alpha \leq \|s\|_\beta$.*

Note that, one can equivalently write $\alpha \in \mathsf{dominion}_x(\beta)$ or $\phi_x(\alpha) = \beta$. Hence $\phi_x(\alpha) = \beta \Rightarrow \|s\|_\alpha \leq \|s\|_\beta$.

Proof. Let s, α, β as before. Let $s' : \star \twoheadrightarrow y$ the smallest even prefix of s that reaches a position where β is available. Then, as $\phi_y^{-1}(\beta) = \phi_x^{-1}(\beta)$, it entails that $\alpha \in A_y^+$. Now, as β is a negative cell, is has been brought by a player move. This one has to be the last move of s'. Similarly, as α is a positive cell, it has been brought by an opponent move, and therefore has to be already available before the last move of s'. That is, $\|s'\|_\alpha < \|s'\|_\beta$. Finally, as for any given cell $\gamma \in A_x \cap A_y$, we naturally have $\|s\|_\gamma = \|s'\|_\gamma$, we conclude. □

For instance, let us suppose that α appears at a position y in s. As $\alpha \in A_x^+$, it is introduced by an opponent move m, and we consider the positions $\sigma^\bullet \ni x \xrightarrow{\mathsf{m}} y \xrightarrow{\mathsf{n}} z \in \sigma^\bullet$ right below and above y in s. Suppose that $\phi_z(\alpha) = \beta$ points to a cell available at x. This situation is pictured in the figure below:



This entails there is a sequence in $\sigma \restriction_x \beta$ that contains a move m above α. Consequently, there is a sequence in $\sigma \restriction_x \beta$ that contains a move above λ (as α is above λ), which is in contradiction with the definition of $\sigma \restriction_x \beta$. Therefore, the cell that $\phi(\alpha)$ points to appears after α; it is brought up by n in z.

We end up this section with a final technical lemma.

Lemma 5.24. *Let σ an innocent strategy, and $x, y \in \sigma^\bullet$ such that $x < y$. Let $A_{x<y} \subseteq A_x$ be the subset of cells α available at x such that $\exists \mathsf{m}.\alpha \vdash \mathsf{m}$ and $x \xrightarrow{m}\!\!\twoheadrightarrow y$. Then $\phi_x \upharpoonright A_{x<y}$ is a total, surjective function $A_{x<y}^+ \to A_{x<y}^-$*

Proof. We focus on totality. Let $\alpha \in A^+_{x<y}$. Then let $s' : x \twoheadrightarrow y \in \sigma$. This path has a move m such that $\alpha \vdash \mathsf{m}$. Therefore, $\exists\, \beta \in A^-_{x<y}$ such that $\mathsf{m} \in \sigma \mid \beta$. Then $\phi_x(\alpha) = \beta$ and hence ϕ is total. □

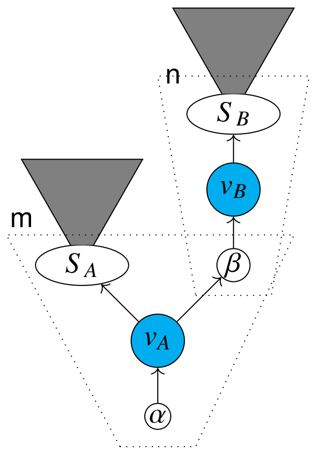
5.2 On composition of innocent strategies

5.2.1 Transverse strategies

We remind that our goal is to have a category, whose objects are the dialogue games, seen as arenas, and whose morphisms $A \to B$ are strategies on $A \triangleright B$. In order to make it work, we have to constrain ourselves to strategies that are transverse. They are those whose positions in $A \triangleright B$ are transverse, as defined in section 4.5.4.

Definition 5.25.
- *A play s of $A \triangleright B = (A \otimes \neg B)^*$, of length more than 2 is* ***transverse*** *if the second move of s belongs in the arena B.*
- *A strategy is* ***transverse*** *if every play in it is transverse.*

Equivalently, a play is transverse if it reaches a transverse position, and a position is transverse if it is reached by a transverse play. Let us remind here the structure of $A \triangleright B$, in the easy case where A, B are simple.



Then any transverse play of length greater than 2 starts with $\mathsf{m}.\mathsf{n}$ (or any congruent moves). Finally, let us remind that in section 4.5.4, we established a bijection:

$$\mathsf{Trans}(A \triangleright B) \simeq \mathsf{Pos}(A) \mathbin{\overline{\otimes}} \mathsf{Pos}(B).$$

However, if we restrict to legal positions, then:

$$\mathsf{Legal}(\mathsf{Trans}(A \triangleright B)) \neq \mathsf{Legal}(A) \mathbin{\overline{\otimes}} \mathsf{Legal}(B).$$

since in the right hand side term, a name of $\mathbb{A}_{\mathsf{cells}}$ might belong to both A and B. Hence, the bijection is as follows:

$$\mathsf{Legal}(\mathsf{Trans}(A \triangleright B)) \simeq \{(x, y) \mid (x, y) \in \mathsf{Legal}(A) \,\overline{\otimes}\, \mathsf{Legal}(B), x \#_{\mathbb{A}_{\mathsf{cells}}} y\}.$$

5.2.2 Relational and sequential compositions

We are in possession of two descriptions of innocent strategies. One is based on the set of positions they reach, the second on the set of sequences that realise them. We now address composition, and prove that relational and sequential composition are equivalent.

5.2.2.1 Sequential composition

We consider interaction of sequences. Given σ, τ define $\sigma \mid \tau$ by :

$$\sigma \mid \tau = \{s \in \mathsf{Legal}(A \triangleright B \triangleright C) \mid s \upharpoonright A \triangleright B \in \sigma, \text{ and } s \upharpoonright B \triangleright C \in \tau\}.$$

We might want to prove the projection of $s \upharpoonright A \triangleright C$ leads to a legal, alternating sequence. Unfortunately, it does not necessarily hold, as the play might be non-alternating. Indeed, consider the following case, where we write O/P to indicate that the polarity of the move is O from the left arena point of view, and P from the right arena point of view:

A	B	C
O		
	P/O	
		P
		O
	O/P	
P		
O		
	P/O	
	O/P	
		O
		P
	P/O	
O		

By projecting this play on $A \triangleright C$, we reach a non-alternating $O-P-O-P-O-O-P-P$ sequence. So we have to select the alternating plays, leading to the following definition.

Definition 5.26. *Given two strategies $\sigma : A \to B$ and $\tau : B \to C$ we define their sequential composition by:*

$$\sigma; \tau = \{s \in \mathsf{Alt}(A \triangleright C) \mid \exists t \in \sigma \mid \tau.\ t \upharpoonright A \triangleright C = s\}.$$

5.2.2.2 Relational Composition

In this paragraph, we briefly recall the definition of relational composition. Given a transverse innocent strategy $\sigma : A \triangleright B$, its set of positions $\sigma^\bullet$ forms a subset of $\mathsf{Legal}(A) \mathbin{\overline{\otimes}} \mathsf{Legal}(B)$. Similarly, if we consider a second transverse innocent strategy $\tau : B \triangleright C$, its set of positions $\tau^\bullet$ forms a set of positions in $\mathsf{Legal}(B) \mathbin{\overline{\otimes}} \mathsf{Legal}(C)$. Using the fact that $\mathsf{Legal}(A) \mathbin{\overline{\otimes}} \mathsf{Legal}(B) \subseteq \mathsf{Legal}(A) \times \mathsf{Legal}(B)$, one can see the set $\sigma^\bullet$ as a relation $\mathsf{Legal}(A) \to \mathsf{Legal}(B)$. We are interested in making use of the relational composition. The only property that needs to be checked is that, given σ and τ as above, does $\sigma^\bullet ; \tau^\bullet \subseteq \mathsf{Legal}(A) \times \mathsf{Legal}(C)$ correspond to a subset of positions of $\mathsf{Legal}(\mathsf{Trans}(A \triangleright C))$. Unfortunately, this is not the case, as the name separation condition is not invariant under composition. For instance, if we compose $a \triangleright b$ with $b \triangleright a$, we obtain $a \triangleright a$, where the name a is reused across the $\triangleright$ operator.

To cope with these minor difficulties, given two subsets $\mathcal{R} \subseteq A \times B$ and $\mathcal{R}' \subseteq B \times C$ we define a new $;_{\mathsf{Rel}}$ for the remaining of this section as follows :

$$\mathcal{R} ;_{\mathsf{Rel}} \mathcal{R}' = \{(x, z) \mid x \#_{\mathsf{cells}} z, \exists y, (x, y) \in \mathcal{R}, (y, z) \in \mathcal{R}'\}$$

Finally, we write $\mathcal{R} \mid_{\mathsf{Rel}} \mathcal{R}'$ for the subset of $A \times B \times C$ defined below:

$$\mathcal{R} \mid_{\mathsf{Rel}} \mathcal{R}' = \{(x, y, z) \mid (x, y) \in \mathcal{R}, (y, z) \in \mathcal{R}', x \#_{\mathsf{cells}} z\}.$$

Finally, given a position (x, y) of $\mathsf{Legal}(A) \triangleright \mathsf{Legal}(B)$, such that $x \#_{\mathbb{A}_{\mathsf{cells}}} y$, we will indifferently deal with it as an element of either the original set or $\mathsf{Legal}(\mathsf{Trans}((A \triangleright B))$. Sometimes, as analogy with the theorem 5.13 we will write X for a definable set of positions.

5.2.2.3 The correspondence

Proposition 5.27. *Let* $(x, y, z) \in \sigma^\bullet \mid_{\mathsf{Rel}} \tau^\bullet$. *Then there is a sequence* $s : \star \twoheadrightarrow (x, y, z) \in \sigma \mid_{\mathsf{Rel}} \tau$.

In the sequel, we write **conflict-freeness** for the property that given a play $s : \star \twoheadrightarrow x \in \sigma$, if there exists a $y \in \sigma^\bullet$ such that $y \geq x$, given any opponent move $\mathsf{m} : x \to x'$ such that $x' \leq y$, there exists a path s' such that $\sigma \ni s.\mathsf{m}.s' : \star \twoheadrightarrow y$. This is a reformulation of forward confluence, and hence every innocent strategy is indeed conflict-free. We also write $(\sigma \mid \tau)^\bullet$ for the set of positions reached by sequences in $(\sigma \mid \tau)$. Moreover, we shall write P-cell, P-move for proponent cell, proponent move, and respectively for O and opponent.

Proof. Suppose this is not the case and let $s : \star \twoheadrightarrow (x', y', z') \in \sigma \mid \tau$ be an even alternating sequence that reaches a maximal position under (x, y, z) in $(\sigma \mid \tau)^\bullet$. Then $s \upharpoonright A \triangleright B \in \sigma$, and, as σ is conflict free, let s' be a sequence such that $(s \upharpoonright A \triangleright B).s' : \star \twoheadrightarrow (x, y) \in \sigma$. Furthermore, let

t' be its counterpart in $B \triangleright C$. Now, if either s' or t' has its first move in either A or C (let us say, s'), then we add to s the two first moves of s'. Either the two are in A, in which case we have reached a higher position of $(\sigma \mid \tau)^\bullet$ under (x, y, z). Or, one is in A and the other in B. In this case, the move in B appears as opponent move for τ. Furthermore, the $B \triangleright C$ part is still under $y \triangleright z$, hence by conflict freeness, we can complete the sequence by the reaction from τ. Then if this move is in C, we stop, and have reached a new position of $\sigma^\bullet \mid_{\mathsf{Rel}} \tau^\bullet$. If it is in B, then by conflict-freeness, we can extend the sequence with a reaction from σ, and so on. This whole process terminates, as any position of B can only have a finite number of moves. Furthermore, the newly created play has alternating projection on $A \triangleright C$. Therefore, in the case where either s', t' has its first move in A or C, then we can reach a higher position under (x, y, z) contradicting the maximality of (x', y', z').

So let suppose that for all sequences s', t' as above, the first moves of s' and t' are in B. By conflict-freeness of σ and τ, it means that every path $x' \twoheadrightarrow x$ (respectively $z' \twoheadrightarrow z$) starts with a P-move, and hence there are only P-cells below x in x' (respectively below z in z'). That is, all paths from (x', y') to (x, y) in σ (respectively from (y', z') to (y, z) in τ) begin with an O-move from y'. We will see that there is a contradiction.

Let ϕ be the sequentiality structure associated with σ, and ψ the sequentiality structure associated with τ . Consider α a available cell of $A_{y'<y}$ that has been introduced last in s. Then suppose that α is an O-cell of B, then it is a P-cell of $B \triangleright C$. Then as $\alpha \in A^+_{(y',z')<(y,z)}$, $\psi_{(y',z)'}(\alpha)$ is well defined. Furthermore, as only P-cells are available at z', $\beta = \psi_{(y',z')}(\alpha)$ is a cell of y'. But then β must have been introduced after α, by proposition 5.23. Furthermore, β belongs in $A_{y'<y}$ by 5.24. However, α was, by definition, a latest cell of $A_{y'\leq y}$ to be introduced, we reach a contradiction. Thus, we can conclude that α is a P-cell of B. In that case, we can repeat the reasoning using ϕ instead of ψ, and get a similar contradiction.

Overall, we reach the conclusion that there exist sequences s' or t' that start with an O-move in A or C, and (x', y', z') is not the highest position under (x, y, z). And therefore, by contradiction, there is a sequence $s : \star \twoheadrightarrow (x, y, z) \in \sigma \mid_{\mathsf{Rel}} \tau$. Furthermore, as the position is legal, the sequence leading to it is legal. □

The reverse direction is straightforward. If there is a sequence $s : \star \twoheadrightarrow (x, y, z)$ then $(x, y, z) \in \sigma^\bullet \mid_{\mathsf{Rel}} \tau^\bullet$, as $(x, y) \in \sigma^\bullet$, $(y, z) \in \tau^\bullet$, and legal interaction sequences lead to legal positions. Therefore, there is a correspondence between the two compositions. We now investigate briefly some properties about the witness of interaction.

5.2.2.4 Some additional properties

Before studying more in depth the relation between sequential and relational composition, we prove this simple proposition.

Proposition 5.28. *Let* (x_1, z_1) *and* (x_2, z_2) *in* $X ;_{\mathsf{Rel}} Y$*, such that* $(x_1, z_1)\ C_{\mathsf{post,cell}}\ (x_2, z_2)$*. Then,* *there exists* y_1, y_2 *such that* $X \mid_{\mathsf{Rel}} Y \ni (x_1, y_1, z_1)\ C_{\mathsf{post,cell}}\ (x_2, y_2, z_2) \in X \mid_{\mathsf{Rel}} Y$.

Proof. Given any position (x, y) of X in $\mathsf{Pos}(A) \overline{\otimes} \mathsf{Pos}(B)$, then the position $(x, y) \in X$ is legal and therefore $x \#_{\mathsf{cells}} y$. Therefore, picking $\pi \in \mathsf{Perm}(\mathbb{A}_{\mathsf{cells}})$ such that $\pi \#_{\mathsf{cells}} x$ then $\pi \cdot (x, y) = (x, \pi \cdot y)$. Furthermore, as X is closed under permutation, $(x, \pi \cdot y) \in X$. So let us consider $(x_1, y_1, z_1), (x_2, y_2, z_2) \in X \mid_{\mathsf{Rel}} Y$ witnessing (x_1, z_1), (x_2, z_2) respectively. Applying appropriate permutations of $\mathbb{A}_{\mathsf{cells}}$ if necessary, we assume $y_1, y_2 \#_{\mathsf{cells}} x_2, z_2, x_1, z_1$. Then we simply apply the proposition 4.51 on the positions y_1, y_2 to conclude the existence of $\pi \in \mathsf{Perm}(\mathbb{A}_{\mathsf{cells}})$ such that such that $\pi \cdot y_1\ C_{\mathsf{post,cell}}\ y_2$. Furthermore, π could be chosen such that $\pi \# x_1, z_1$, and $\pi \cdot y_1 \#_{\mathsf{cells}} x_1, z_1$. This way, we obtain $\pi \cdot (x_1, y_1, z_1) = (x_1, \pi \cdot y_1, z_1)$, and $(x_1, y_1, z_1)\ C_{\mathsf{post,cell}}\ (x_2, y_2, z_2)$ as expected. □

The properties proven in this section will be useful in order to prove that the composition of innocent strategies results in innocent strategies. The first goal is to establish that there is a unique witness of interaction, and then to prove that this one behaves well with regard to compatibility. We prove below that the strategy "preserves the compatibility": the incompatibility between different positions of the strategy has to originate from the opponent. That is, it must be opponent that introduces first a move incompatible with future positions.

Recall that we write $(\mathsf{m} \smile \mathsf{n})$ for the conflict relation, defined by $\mathsf{m} \smile \mathsf{n} \Leftrightarrow \neg(\mathsf{m} \uparrow \mathsf{n})$. Given a path $t = \mathsf{n}_1. \ldots .\mathsf{n}_k$ and a move m, we write $\mathsf{m} \smile t$ if $\exists \mathsf{n} \in t. \mathsf{m} \smile \mathsf{n}$. This extends to paths; $s \smile t$ if $\exists \mathsf{m} \in s, \mathsf{n} \in t. \mathsf{m} conflit \mathsf{n}$. Similarly, given a position x, we write $\mathsf{m} \smile x$ if, seeing x as a set of moves, $\exists \mathsf{n} \in x. \mathsf{m} \smile \mathsf{n}$.

Lemma 5.29. *Let* $x, y \in \sigma^\bullet$ *such that* $x\ C_{\mathsf{post,cell}}\ y$ *and* $x \smile y$, *then given any path* $\sigma \ni s : \star \twoheadrightarrow x$, *the first move* m *of* s *such that* $\mathsf{m} \smile y$ *is an opponent move.*

Proof. Let x' be the greatest member of $\sigma^\bullet$ under x such that $x \uparrow y$. Namely, we define:

$$x' = \bigsqcup \{z \in \sigma^\bullet \mid z \leq x, z \uparrow y\}.$$

By noticing that the set $\{z \in \sigma^\bullet \mid z \leq x, z \uparrow y\}$ is stable under union (since $\sigma^\bullet$ is closed under compatible union 5.1.3, we conclude that x' belongs to it. Similarly, we define y' for y. As $\sigma^\bullet$ is closed under compatible union, $x' \sqcup y', x \sqcup y', x' \sqcup y \in \sigma$. Now let us consider two paths $\sigma \ni s : x' \sqcup y' \twoheadrightarrow x \sqcup y'$ (meaning that there exists $s' : \star \twoheadrightarrow x' \sqcup y'$ such that $s'.s \in \sigma$) and $\sigma \ni t : x' \sqcup y' \twoheadrightarrow x' \sqcup y$. Such paths exists since $x' \sqcup y' \leq x \sqcup y$ and σ is conflict-free. These paths are non-empty, as $x \smile y$ so $x' \neq x$ and $y' \neq y$. Let us name m the moves of s ($s = \mathsf{m}_1.\mathsf{m}_2. \ldots$), and n the moves of t ($t = \mathsf{n}_1.\mathsf{n}_2. \ldots$). Then we shall have $\mathsf{m}_1.\mathsf{m}_2 \smile \mathsf{n}_1.\mathsf{n}_2$, otherwise this would contradict the maximality of x' and y' as parts of x, y that are not in conflict. It should be noted that if $\mathsf{m}_1 \Uparrow \mathsf{n}_1$ (respectively $\mathsf{m}_1 = \mathsf{n}_1$), then, by the forward consistency of innocence, (respectively by nominal determinacy and compatibility) $\mathsf{m}_1.\mathsf{m}_2 \uparrow \mathsf{n}_1.\mathsf{n}_2$ (respectively $\mathsf{m}_1.\mathsf{m}_2 = \mathsf{n}_1.\mathsf{n}_2$), and therefore, $\mathsf{m}_1 \smile \mathsf{n}_1$.

So let us consider a play and a m such as in the lemma, and assume for contradiction that it is a player move. Then, in the play, it belongs to an OP-pair $\mathsf{m}'.\mathsf{m}$, and this one does not belong in x'. Therefore, there is a path of the strategy from $x' \sqcup y' \twoheadrightarrow x \sqcup y'$ that starts with $\mathsf{m}'.\mathsf{m}$, and $\mathsf{m}' \smile y$ as explained above, contradicting that m is the first move of the play to be in conflict with y. □

The previous lemma will help us prove the following proposition, stating the unicity of the witness of interaction.

Proposition 5.30. *Let* $(x, y, z) \in \sigma^\bullet \mid_{\mathsf{Rel}} \tau^\bullet$*, and* $(x, y'z) \in \sigma^\bullet \mid_{\mathsf{Rel}} \tau^\bullet$*, such that* $y\ C_{\mathsf{cells}}\ y'$*, then* $y = y'$*.*

Proof. We already proved earlier in lemma 5.29 that if there were two distinct y, such that $y\ C_{\mathsf{post,cell}}\ y'$, then they must be in conflict. According to the previous lemma, from the σ point of view, opponent must initiate the conflict in B, that is, proponent from the τ point of view. On the other hand, from the τ point of view, it must be the opponent that initiates conflict, that is, the proponent from the σ point of view. This is a dead-end, and there is a unique y.

Finally we note that requiring $y\ C_{\mathsf{cells}}\ y'$ instead of $y\ C_{\mathsf{post,cell}}\ y'$ is enough. Indeed, let us pick two y, y' such that $y\ C_{\mathsf{cells}}\ y'$, and π such that $\pi \cdot y\ C_{\mathsf{post,cell}}\ y'$. Then, in the lemma above, this would imply $\pi \cdot y = y'$. Therefore $y \simeq_{\mathsf{cells}} y'$, and, as $y\ C_{\mathsf{cells}}\ y'$, $y = y'$. In particular, given $(x, y, z), (x, y', z) \in \sigma^\bullet \mid_{\mathsf{Rel}} \tau^\bullet$, then taking $\pi \in \mathsf{Perm}(\mathbb{A}_{\mathsf{cells}})$ such that $\pi \cdot y\ C_{\mathsf{cells}}\ y'$ entails $\pi \cdot y = y'$. That is, $y \simeq_{\mathsf{cells}} y'$. □

The witness of interaction preserves compatibility, as formulated in the below proposition. This lemma is a consequence of the preservation of compatibility by the strategy.

Proposition 5.31. *Let* $(x_1, z_1), (x_2, z_2) \in \sigma^\bullet ;_{\mathsf{Rel}} \tau^\bullet$ *such that* $(x_1, z_1) \uparrow (x_2, z_2)$*. Let* y_1, y_2 *such that* $y_1\ C_{\mathsf{post,cell}}\ y_2$ *and* $(x_1, y_1, z_1), (x_2, y_2, z_2) \in \sigma^\bullet \mid_{\mathsf{Rel}} \tau^\bullet$*. In that case* $y_1 \uparrow y_2$*, and, in particular,* $y_1\ C_{\mathsf{post}}\ y_2$*.*

Proof. Suppose $y_1 \smile y_2$. Let $\sigma \mid \tau \ni s_1 : \star \twoheadrightarrow (x_1, y_1, z_1)$, and let n be the first move of s_1 such that $\mathsf{n} \smile y_2$. Then by the lemma above 5.29 it has to be an opponent from the σ point of view, and an opponent from the τ point of view. So we reach a contradiction, and $y_1 \uparrow y_2$. In particular, by proposition 4.49, $y_1\ C_{\mathsf{post}}\ y_2$. □

5.2.3 Innocent strategies are stable under composition

We find it easier to handle composition through the relational composition of strategies as sets of positions. Our goal is to prove that the composition of two innocent strategies leads to an innocent strategy. For sets of positions, this translates into the following proposition.

Proposition 5.32. *The relational composition of two definable sets leads to a definable set.*

Proof. To begin, let us properly define the notations used along the proof. We consider two definable sets $X \subseteq \mathsf{Legal}(\mathsf{Trans}(A \triangleright B))$ and $Y \subseteq \mathsf{Legal}(\mathsf{Trans}(B \triangleright C))$. If needed, we might consider the strategies σ and τ associated respectively to X and Y. We look at X, Y as sets of positions in $\mathsf{Legal}(A) \mathbin{\overline{\otimes}} \mathsf{Legal}(B)$ and $\mathsf{Legal}(B) \mathbin{\overline{\otimes}} \mathsf{Legal}(C)$.

The first condition (1) to check is $\bot \in X ;_{\mathsf{Rel}} Y$. This comes from $(\bot_A, \bot_B) \in X, (\bot_B, \bot_C) \in Y$, and hence $\bot = (\bot_A, \bot_C) \in X ;_{\mathsf{Rel}} Y$, as $(\bot_A, \bot_B) ;_{\mathsf{Rel}} (\bot_B, \bot_C) = (\bot_A, \bot_C)$.

Next, we need to ensure (2) that $X ;_{\mathsf{Rel}} Y \in \mathsf{Legal}(A \triangleright C)$. This follows from the definition of $;_{\mathsf{Rel}}$, that ensures that the resulting positions are indeed legal.

We now investigate the nominal closure (3) of $X ;_{\mathsf{Rel}} Y$. Given an element $x, z \in X ;_{\mathsf{Rel}} Y$, and $y \in B$ such that $(x, y, z) \in X \mid_{\mathsf{Rel}} Y$, then for all $\pi \in \mathsf{Perm}(\mathbb{A})$, $(\pi \cdot x, \pi \cdot y, \pi \cdot z) \in X \mid_{\mathsf{Rel}} Y$ as X, Y are closed under permutation. Therefore $\pi \cdot (x, y) = (\pi \cdot x, \pi \cdot z)$ belongs in $X ;_{\mathsf{Rel}} Y$. Closure under typed substitutions is dealt with on a equal footing.

The two following properties are closure under compatible intersection (4) and union (5). We tackle both at once. Let $(x_1, z_1), (x_2, z_2) \in X ;_{\mathsf{Rel}} Y$ such that $(x_1, z_1) \uparrow (x_2, z_2)$. For the union, we also assume that $(x_1, z_1)\ C_{\mathsf{post}}\ (x_2, z_2)$. Then let y_1, y_2 such that $(x_1, y_1, z_1)\ C_{\mathsf{cells}}\ (x_2, y_2, z_2) \in X \mid_{\mathsf{Rel}} Y$ (respectively $(x_1, y_1, z_1)\ C_{\mathsf{post,cell}}\ (x_2, y_2, z_2) \in X \mid_{\mathsf{Rel}} Y$ for the union). As proven in the lemma 5.31, $y_1 \uparrow y_2$, and consequently $(x_1, y_1, z_1) \uparrow (x_2, y_2, z_2)$. In particular, $y_1\ C_{\mathsf{post}}\ y_2$. As X, Y are closed under compatible union and intersection, the element $(x_1 \sqcap x_2, y_1 \sqcap y_2, z_1 \sqcap z_2)$ belongs in $X \mid_{\mathsf{Rel}} Y$, just as $(x_1 \sqcup x_2, y_1 \sqcup y_2, z_1 \sqcup z_2)$ (in the case where they are post compatible). So $(x_1 \sqcap x_2, z_1 \sqcap z_2) \in X ;_{\mathsf{Rel}} Y$, $(x_1 \sqcup x_2, z_1 \sqcup z_2) \in X ;_{\mathsf{Rel}} Y$, and therefore $X ;_{\mathsf{Rel}} Y$ is closed under compatible union and intersection.

The sixth property (6) focuses on conservation of compatibility after two opponent moves. This corresponds to the first part of forward consistency, namely that given two opponent moves m, m' starting from a position x of an innocent strategy and such that $\mathsf{m} \Uparrow \mathsf{m}'$, if the strategy reacts by playing two moves n, n' then, taking n, n' such that $\mathsf{n}\ C_{\mathsf{post,cell}}\ \mathsf{n}'$, it entails $\mathsf{n} \Uparrow \mathsf{n}'$. So let (x, y, z) be a position of $\sigma^\bullet \mid_{\mathsf{Rel}} \tau^\bullet$. Let m, m' two moves starting either from x or z, and (x_1, y_1, z_1) (respectively (x_2, y_2, z_2)), the first position of $\sigma^\bullet ;_{\mathsf{Rel}} \tau^\bullet$ above $(x, y, z) \uplus \mathsf{m}$ (respectively $(x, y, z) \uplus \mathsf{m}'$), and such that, furthermore, $(x_1, y_1, z_1)\ C_{\mathsf{post,cell}}\ (x_2, y_2, z_2)$. As, as proven in the lemma 5.29, both σ, τ preserve compatibility, and as $\mathsf{m} \Uparrow \mathsf{m}'$, this entails $(x_1, y_1, z_1) \uparrow (x_2, y_2, z_2)$. In particular, using permutations of $\mathbb{A}_{\mathsf{cells}}$, these could be chosen to be post-compatible. Then, using the compatibility under union, we get that $(x_1, z_1) \sqcup (x_2, z_2)$ forms a common bound for $(x, z) \uplus \{\mathsf{m}\}$ and $(x, z) \uplus \{\mathsf{m}'\}$.

Before proving the next properties, we introduce the following remark. Let $x = (x_A, x_C), y = (y_A, y_C)$ two positions of $X ;_{\mathsf{Rel}} Y$ such that $x < y$, and let us take two positions $x' = (x_a, x_B, x_C)$ and $y' = (y_A, y_B, y_C)$ in $X \mid_{\mathsf{Rel}} Y$ such that $x'\ C_{\mathsf{cells}}\ y'$. Then $x' < y'$. Indeed, by preservation of compatibility, $x' \uparrow y'$ and in particular $x'\ C\ y'$. By stability under compatible intersection, $(x_A \sqcap y_A, x_B \sqcap y_B) \in X$, and $(x_B \sqcap y_B, x_C \sqcap y_C) \in Y$. As $y_A \geq x_A$ and $y_C \geq x_C$, this entails $(x_A, x_B \sqcap y_B, y_C) \in X \mid_{\mathsf{Rel}} Y$. Furthermore, $x_B \sqcap y_B\ C_{\mathsf{cells}}\ x_B$ by definition. Consequently, by proposition 5.30, $x_B \sqcap y_B = x_B$. This entails $y_B \geq x_B$, and hence $x' < y'$.

On the seventh point, we tackle the stability of forward confluence (7)(8). Let $x \in X \,;_{\mathsf{Rel}}\, Y$ such that there is an opponent move $x \xrightarrow{\mathsf{m}} y$ and y is dominated in $X \,;_{\mathsf{Rel}}\, Y$ by w. One needs to prove that there is a unique P-move n such that $x \overset{\mathsf{m.n}}{\twoheadrightarrow} z$, $z \in X \,;_{\mathsf{Rel}}\, Y$, $z \leq w$ and furthermore $\nu_T(\mathsf{n}) \subseteq \nu_T(x) \cup \nu_T(\mathsf{m})$. We suppose (without lost of generality) that m is in A. We use the property of the unique witness of interaction 5.30. Let $x = (x_A, x_C)$, then there is a unique (up to cells permutation) x_B such that $(x_A, x_B, x_C) \in X \,|_{\mathsf{Rel}}\, Y$. As $w \in X \,;_{\mathsf{Rel}}\, Y$, there exists a unique, up to cells permutation, $w' \in X \,|_{\mathsf{Rel}}\, Y$ such that $w' \upharpoonright A \triangleright C = w$. Furthermore, w' can be chosen such that $w'\ C\ (x_A, x_B, x_C)$. Suppose that X answers to m in A, then we conclude easily. In the case where X answers m with a move n_1 in B, then it will answer it uniquely such that it reaches a position dominated by w' Furthermore, this move will respect the type condition on its support. Now, however, it is Y's turn, and for the same reason, Y will answer uniquely by a n_2 whose target position that will remain dominated w'. Furthermore, $\nu_T(\mathsf{n}_2) \subseteq \nu_T(x') \cup \nu_T(\mathsf{m}) \cup \nu_T(\mathsf{n}_1) = \nu_T(x') \cup \nu_T(\mathsf{m})$. This way we can build an alternated sequence of moves alternating between σ and τ, until one of the strategy moves into either A or C. Then the position reached still remains dominated by w'. Projecting it on $A \triangleright C$, we get a desired P-move reaction in $A \triangleright C$. Furthermore, following the uniqueness of the sequence that leads to this P-move, one can gather that this P-move is unique satisfying the conditions. Finally, the type condition follows from $\nu_T(x') = \nu_T(x)$, since each move in B is a P-move from either the left or the right point of view. (7) is proven on an equal footing.

Remaining is the proof of mutual attraction (10). Let $s : \star \twoheadrightarrow y' \in \sigma \,|_{\mathsf{Rel}}\, \tau$ and $t : \star \twoheadrightarrow x' \in \sigma \,|_{\mathsf{Rel}}\, \tau$, such that $x' \leq y'$. We set $x' = (x_A, x_B, x_C)$, $x = x' \upharpoonright A \triangleright C = (x_A, x_C)$, $y' = (y_A, y_B, y_C)$ and $y = y' \upharpoonright A \triangleright C = (y_A, y_C)$ as above. Let $s' = s \setminus x'$ defined by induction as follows:

$$s = \emptyset \Rightarrow s' = \emptyset \qquad s = \mathsf{m}.s'' \wedge \mathsf{m} \in x' \Rightarrow s' = s'' \setminus x \qquad s = \mathsf{m}.s'' \wedge \mathsf{m} \notin x' \Rightarrow s' = \mathsf{m}.(s'' \setminus x)$$

The resulting s' is simply a sequence of moves. It is not a play, as it does not necessarily start from the root. We consider the sequence $t.s'$. Our goal is to show that it is in $\sigma \,|_{\mathsf{Rel}}\, \tau$. First of all, we need to check that $t.s'$ is a play. We check the 3 conditions of proposition 4.56. If $x = \bot$, this is straightforward. In the case where $x \neq \bot$, the first move of t is an initial move, and this is the only one of $t.s$. Furthermore, no moves are repeated in $t.s$. Finally, we need to check the down-closure. Let m be a move in $t.s$. Either this move belongs to t, and, as t is a play, the predecessor of m is in t. Either it belongs to s'. Then either its predecessor belongs in s' as well, and, as s is a play, it appeared before it in the sequence. Or it does not belong to s', in which case we can conclude it belonged to x'. In that case, it appeared in t, and hence before it. So this sequence forms a play. Furthermore, it reaches the position y', which is legal. Therefore the sequence is legal. Now, let us look at $t.s' \upharpoonright A \triangleright B$. We already know that $t \upharpoonright A \triangleright B \in \sigma$. We will prove that $t.s' \upharpoonright A \triangleright B \in \sigma$ by even induction on the length of s'. So let $s' = s_1.\mathsf{m}.s_2$, s_2 being (at this stage), possibly an empty sequence, such that we already know that $t.s_1 \upharpoonright A \triangleright B \in \sigma$. To start, we establish that m is an O-move. Suppose m is a P-move. Then there is a m', O-move such that m' appears right before m in $s \upharpoonright A \triangleright B \in \sigma$. As σ is closed under $\sim_{OP}$ homotopy,

$s \upharpoonright A \triangleright B$ can be reorganized ($s \upharpoonright A \triangleright B \simeq_{OP} u_1.u_2$), such that $u_1 : \star \twoheadrightarrow (x_A, x_B)$. As m does not belong in u_1, it belongs in u_2. From the $\sim_{op}$ property, we deduct that $\mathsf{m}' \in u_2$ as well. Hence m' does not belong in (x_A, x_B). So m' is in $s' \upharpoonright A \triangleright B$, and appears before m. Applying a similar reasoning to $s_1 \upharpoonright A \triangleright B \preceq_{OP} s$ we can deduct that m' is not in $s_1 \upharpoonright A \triangleright B$, otherwise so would be m. Hence the first move after s_1 should be m', not m. From this contradiction, we gather that the first move m after s_1 in $s' \upharpoonright A \triangleright B$ is an O-move. Using the same decomposition as before ($s \upharpoonright A \triangleright B \sim_{OP} s_1 \upharpoonright A \triangleright B.\mathsf{m}.u_2$), we can deduct that there is a player move n appearing after m in s' such that $s_1.\mathsf{m}.\mathsf{m}' \preceq_{OP} s \upharpoonright A \triangleright B$). Therefore, $t.s_1.\mathsf{m}.\mathsf{n} \in \sigma$.

Eventually, we conclude that $t.s' \upharpoonright A \triangleright B$ is in σ, and, symmetrically, that $t.s' \upharpoonright B \triangleright C$ is in τ. So $t.s' \in \sigma \mid_{\mathsf{Rel}} \tau$. By projecting on $A \triangleright C$, we get a sequence $(x_A, x_C) \twoheadrightarrow (y_A, y_C)$. Remaining is simply to prove that this sequence starts with an O-move and finishes with a P-move. Let us look at the sequence $t.s' \in \sigma \mid_{\mathsf{Rel}} \tau$. If the first move of s' would be in B, then from the σ perspective, it would have to be an opponent move, that is, a P-move from the τ perspective, and similarly for τ. Therefore, it is a move in A or C. If it is in A, it would have to be a O-move of $A \triangleright B$, and therefore an O-move of $A \triangleright C$. The same can be said if the first move is in C. On the other hand, if we suppose the last move is in B, we obtain a similar kind of contradiction. Therefore, the last move would be in A, or C, and a P-move from the $A \triangleright C$ perspective. Therefore, $s' \upharpoonright A \triangleright C$ starts with an O-move and finishes with a P-one. This concludes the proof. □

Corollary 5.33. *Innocent, transverse, typed coherent strategies are stable under composition.*

Innocent strategies form the ground structure behind denotation of tensorial logic proofs. However, they do not form a fully complete model. We will have to select those strategies that satisfy additional conditions to achieve the final result. Before presenting those, we examine how the weak sequentiality structures that come with innocent strategies behave with composition.

5.2.3.1 The category Inn

In order to conclude that the innocent, transverse, typed-coherent strategies form a category, we simply need to present strategies that act as identity morphisms. These are called **copy-cat** strategies.

Given an arena A, we define the transverse, innocent strategy typed-coherent $\mathsf{copycat}_A : A \triangleright A$ as follows, where we tag the two occurrences of A in $A \triangleright A$ as A_1, A_2 respectively.

$$\mathsf{copycat}_A = \mathsf{Alt}(\mathsf{Legal}(\{s \in \mathsf{Play}(A_1 \triangleright A_2) \mid s \upharpoonright A_1 \simeq_{\mathsf{cells}} s \upharpoonright A_2\}))$$

where we recall that we write $\simeq_{\mathsf{cells}}$ to signify that there must exist $\pi \in \mathsf{Perm}(\mathbb{A}_{\mathsf{cells}})$ such that $\pi \cdot s \upharpoonright A_1 = s \upharpoonright A_2$. The $\mathsf{copycat}$ strategy can also be described through its set of positions:

$$\mathsf{copycat}^\bullet_A = \mathsf{Legal}(\{(x, y) \in (\mathsf{Trans}(A_1 \triangleright A_2)) \mid x \simeq_{\mathbb{A}_{\mathsf{cells}}} y\})$$

Indeed, every position (x, y) such that $x \simeq y$ can be reached by an alternating sequence.

Proposition 5.34. $\mathsf{copycat}_A$ *is a transverse, typed-coherent, innocent strategy.*

Proof. We work with $\mathsf{copycat}_A$ as a set of sequences. By definition $\mathsf{copycat}_A$ is closed under nominal equivalence and strict substitutions. Furthermore, given $s \in \mathsf{copycat}_A$, and m in, let us say, A_1 such that $s.\mathsf{m}$ is legal, then $s.\mathsf{m}.\mathsf{n} \in \mathsf{copycat}_A$ with $\mathsf{n} \simeq_{\mathsf{cells}} \mathsf{m}$ in A_2. Hence, $s.\mathsf{m}.\mathsf{n}$ is legal, and $\mathsf{copycat}_A$ is closed by even prefix obviously. Furthermore, the condition $s \upharpoonright A_1 \simeq_{\mathsf{cells}} s \upharpoonright A_2$ ensure that all the sequences in it are even-length, and that the strategy is nominal deterministic.

To simplify thinks, given a move m in A_1, we write m^c for its equivalent one in A_2, and reversely. That is, $(\mathsf{m}^c)^c = \mathsf{m}$. Finally, given s in $\mathsf{copycat}_A$, and $\mathsf{m}_1, \mathsf{m}_2$ such that $\mathsf{m}_1 \Uparrow \mathsf{m}_2$ and $s.\mathsf{m}_1, s.\mathsf{m}_2$ are legal and post compatible, then $s.\mathsf{m}_1.(\pi_1 \cdot \mathsf{m}_1^c).\mathsf{m}_2.(\pi_2 \cdot \mathsf{m}_2^c) \in \mathsf{copycat}_A$, (where $\pi_1, \pi_2 \in \mathsf{Perm}(\mathbb{A}_{\mathsf{cells}})$ are picked such that the sequence is legal), and therefore the strategy is forward consistent.

Similarly, if a sequence $s.\mathsf{m}_1.\mathsf{m}_2.\mathsf{n}_1.\mathsf{n}_2.t$ is in $\mathsf{copycat}$, then so is $s.\mathsf{n}_1.\mathsf{n}_2.\mathsf{m}_1.\mathsf{m}_2.t$ and therefore the strategy is backward and forward consistent, that is, innocent. □

Finally, it is straightforward, especially by looking as $\mathsf{copycat}$ as a set of positions, that $\mathsf{copycat}$ acts as the identity: $\sigma; \mathsf{copycat} = \sigma$ and $\mathsf{copycat}; \sigma = \sigma$. So overall, we got this final proposition.

Proposition 5.35. Inn *is a category,*

- *whose objects are positive dialogue games that arise as denotations of formulas of tensorial logic.*
- *whose morphisms are transverse, innocent, typed-coherent strategies of negative dialogue games of the form $A \triangleright B$.*

5.2.4 Composition of weak sequentiality structure

We now address the composition of weak sequentiality structure. Let (x, y) be a position of $\mathsf{Trans}(A \triangleright B)$, seen as $\mathsf{Pos}(A) \overline{\otimes} \mathsf{Pos}(B)$. A weak sequentiality structure $\varphi_{(x,y)}$ is a partial function $\varphi_{(x,y)} : A_x^- \uplus B_y^+ \to A_x^+ \uplus B_y^-$ where we remind that A_x^- are the O-cells of A available at x (and respectively for the other ones). Then, given two innocent transverse strategies $\sigma : A \to B$ and $\tau : B \to C$, let assume φ and ψ are their assigned weak sequentiality structure. Then one might wonder if the weak sequentiality structure of $\sigma; \tau$ can be computed directly from ϕ, ψ.

Let $w = (x, y, z) \in (\sigma^\bullet ;_{\mathsf{Rel}} \tau^\bullet)$, then to compute $(\varphi; \psi)_{(x,z)} : A_x^- \uplus C_z^+ \to A_x^+ \uplus C_z^-$, we can rely on:

$$\varphi_{(x,y)} : A_x^- \uplus B_y^+ \to A_x^+ \uplus B_y^-$$
$$\psi_{(y,z)} : B_y^- \uplus C_z^+ \to B_y^+ \uplus C_z^-$$

We prove the following theorem.

Proposition 5.36. *Let $\varphi, \psi, w, x, y, z$ as before. Then the sequentiality structure of $\sigma;\tau$ at w is a subset of:*

$$\mathsf{Tr}^{B_y^+ \uplus B_y^-}_{A_x^-, C_z^+, A_x^+, C_z^-}((\varphi_{(x,y)} \uplus \psi_{(y,z)}); (\mathsf{id}_{A_x^+} \uplus s_{B_y^-,B_y^+} \uplus \mathsf{id}_{C_z^-})) : A_x^- \uplus B_y^+ \uplus B_y^- \uplus C_z^+ \to A_x^+ \uplus B_y^+ \uplus B_y^- \uplus C_z^-$$

where the trace is taken from the traced symmetric monoidal category $(\mathsf{pSet}, \uplus, \emptyset)$ *of sets and partial functions, with monoidal product the disjoint union of sets, and unit the empty set. The morphism $s_{B_y^-,B_y^+} : B_y^- \uplus B_y^+ \to B_y^+ \uplus B_y^-$ is the morphism coming from the symmetry. That is, denoting $\chi_{x;z}$ the weak sequentiality structure of $\sigma;\tau$,*

$$\chi_{(x,z)}(\alpha) = \beta \;\Rightarrow\; \mathsf{Tr}^{B_y^+ \uplus B_y^-}_{A_x^-, C_z^+, A_x^+, C_z^-}((\varphi_{(x,y)} \uplus \psi_{(y,z)}); (\mathsf{id}_{A_x^-} \uplus s_{B_y^-,B_y^+} \uplus \mathsf{id}_{C_z^+}))(\alpha) = \beta$$

We remind below what is the canonical trace in the category of partial functions. Given $f : A \times C \to B \times C$, we compute $\mathsf{Tr}^C_{A,B}(f)$ as follows :

- if $i \in A$ and $f(i)$ is defined, $f(i) \in B$ then $\mathsf{Tr}^C_{A,B}(f(i)) = f(i)$.
- if $i \in A$, and $f(i)$ is undefined, then $\mathsf{Tr}^C_{A,B}(f)(i)$ is undefined.
- If $i \in A$, $f(i)$ is defined and $f(i) \in C$ then $\mathsf{Tr}^C_{A,B}(f)(i) = \mathsf{feedback}(f)(i)$.
- Given $i \in C$ we get $\mathsf{feedback}(f)(i) = \begin{cases} \mathsf{feedback}(f)(f(i)) \text{ if } f(i) \text{ is defined } \wedge f(i) \in C \\ f(i) \text{ if } f(i) \text{ is defined } \wedge f(i) \in B \\ \text{is undefined if } f(i) \text{ is undefined} \end{cases}$.

Proof. Let $\alpha \in A_x^- \uplus C_z^+$, such that there exists $\beta \in A_x^+ \uplus C_z^-$ satisfying $\alpha \in \mathsf{dominion}_{\sigma;\tau,(x,z)}(\beta)$. Suppose without lost of generality that β is in A_x^+, the case where β is in C_z^+ is dealt with on a equal footing. Let $w = (x, y, z)$ witnessing (x, z). $\alpha \in \mathsf{dominion}_{\sigma;\tau,(x,z)}$ means that there exists a path $s : w \to w' \in \sigma \mid_{\mathsf{Rel}} \tau$, with $s \upharpoonright A \triangleright B \in \sigma$, $s \upharpoonright B \triangleright C \in \tau$, s starts from the cell β, with $\alpha \vdash \mathsf{m} \in s$, and such that $\forall \gamma \in (A_x^+ \uplus C_z^- \setminus \{\beta\}) \neg(\exists \mathsf{m} \in s.\gamma \vdash \mathsf{m})$. Suppose that α is in C_z^+, the case where $\alpha \in A_x^+$ being dealt with on a equal footing. Then, by the fact that $s \upharpoonright B \triangleright C$ is above no O-cell of z, we can deduct the fact that $\psi_{(y,z)}(\alpha) = \alpha_2$ is in B_y^+. Now, the O-move that triggered m, such that $\alpha_2 \vdash \mathsf{m}$ and $\mathsf{m} \in s$, is actually a P-move from the σ point of view. This P-move has been triggered by a O-move from the cell $\varphi_{(x,y)}(\alpha_2) = \alpha_3$. Now α_3 is either a in B_y^- or in A_x^+. If it is in A_x^+ then this is automatically β as β is the only cell in A_x^+ that is explored by s. If it is in B_y^-, then we redo the same reasoning, and conclude that this is triggered from the τ point of view from a O-move above $\psi_{(y,z)}(\alpha_3)$ and $\psi_{(y,z)}(\alpha_3)$ is in B_y^+. Going on like this we obtain a sequence:

$$\begin{aligned} \alpha_1 &= \alpha \\ \alpha_{i+1} &= \psi(\alpha_i) \text{ if } i \text{ is odd} \\ \alpha_{i+1} &= \varphi(\alpha_i) \text{ if } i \text{ is even} \end{aligned}$$

sequence that stops when $\varphi_{(x,y)}(\alpha_i) = \beta$. Now, let $f : A_w \to \mathbb{N}$ defined by $f(\alpha) = \|s\|_\alpha$. Then f is strictly decreasing along $\phi_{(x,y)}$ and $\psi_{(y,z)}$. That is, given (α) such that $\phi_{(x,y)}(\alpha)$ is defined then

$f(\phi_{(x,y)}(\alpha)) < f(\alpha)$ (and equivalently for $\varphi_{(y,z)}$). This ensures us that the sequence above is finite and eventually stops. So we can conclude that $\beta = \mathsf{Tr}^{B_y^+ \uplus B_y^-}_{A_x^-, C_z^+, A_x^+, C_z^-}(\varphi_{(x,y)} \uplus \psi_{(y,z)}; (\mathsf{id}_{A_x^-} \uplus s_{B_y^-, B_y^+} \uplus \mathsf{id}_{C_z^+}))(\alpha) = \beta$. □

Unfortunately the reverse inclusion might not be true. Suppose that there exists a cell α in A_x^+ such that α has three different cells in its $\mathsf{dominion}_{\sigma,(x,y)}$: two in A_x^- and only one in B_y^+. The strategy reacts to a move in α by playing a player move in B_y^-, and depending on the reaction of the opponent, it explores one of the two cells of $\mathsf{dominion}(\alpha)$ located in A, or the other. Then as the context explored depends on the reaction of opponent, when post-composed with another strategy, it will never explore one the cells that was in $\mathsf{dominion}(\alpha)$, (as the reaction of the opponent in B is now encoded into τ, that is deterministic). Hence the dominion of the composite strategy is now restricted to one of the two original cells of A.

This is related to the difficulty of modelling the sequence $\otimes - \&$ by a strategy in linear logic. For instance, consider the sequent $\vdash \Gamma, (A \& B) \otimes \top$. The strategy, playing the tensor $\otimes$, will bring two cells; one corresponding to the & on the left hand side (call it α), and one corresponding to $\top$. Now, calling Γ' the context explored by $\sigma \mid \alpha$, and $\Gamma'' = \Gamma \setminus \Gamma'$, it would be tempting to conclude that the strategy corresponds to the rule :

$$\frac{\vdash \Gamma', A \,\&\, B \qquad \vdash \Gamma'', \top}{\vdash \Gamma, (A \,\&\, B) \otimes \top}$$

Equivalently, in tensorial logic, this would be :

$$\frac{\dfrac{\Gamma', A \oplus B \vdash \bot}{\Gamma' \vdash \neg(A \oplus B)} \qquad \vdash \Gamma'', \top}{\vdash \Gamma, \neg(A \oplus B) \otimes \top}$$

However, the problem is that the context explored might depend whether the strategy decides to explore the left hand side, or the right hand side of the formula $A \,\&\, B$. That is, the context explored by the set of plays of the strategy whose first moves (at this stage) belong in A might be different to the one explored by the set of plays of the strategy whose first moves (at this stage) lie in B. That would lead to the following sequent deduction,

$$\frac{\vdash \Gamma_1', A \qquad \vdash \Gamma_2', B}{\vdash \Gamma', A \,\&\, B}$$

where $\Gamma' = \Gamma_1' \cup \Gamma_2'$, or, equivalently

$$\frac{\Gamma_1', A \vdash \bot \qquad \Gamma_2', B, \vdash \bot}{\Gamma', A \oplus B \vdash \bot}$$

in tensorial logic. However, in both logics, it is required that $\Gamma_1 = \Gamma_2$.

5.3 Refining innocent strategies

Innocent strategies suffer defects that prevent them from forming a fully complete for tensorial logic. First, they are affine, meaning that a part of the context might not be explored and therefore might be discarded. For instance, there is an innocent strategy $A \otimes A \triangleright A$. This problem is also apparent in the confusion between 1 and $\top$. From the game perspective, they are equal (both accepts only one move), whereas they fundamentally differ from the logical point of view. The second problem comes from the fact that on every type there is a strategy, namely, the trivial one whose only play is the empty play. In order to exclude those cases we must focus on strategies that are able to answer every opponent query. Those are called total.

Therefore, we add two properties to our strategies. They shall be total, and with strong sequentiality structures that prevent them from being affine and that encode well the structure of the atomic types.

5.3.1 Totality

A strategy is total if it can always answer an opponent move.

Definition 5.37. *A strategy is* ***total*** *if for all* $s \in \sigma$ *and for all* m *such that* $\lambda(\mathsf{m}) = -1 \wedge s.\mathsf{m} \in \mathsf{Legal}(A)$ *then* $\exists \mathsf{n}.\ s.\mathsf{m}.\mathsf{n} \in \sigma$.

Similarly, this property can be encoded on definable sets.

Definition 5.38. *A definable set* X *is* ***total*** *if* $\forall x \in X. \forall \mathsf{m}. (\lambda(\mathsf{m}) = -1 \wedge x \uplus \{\mathsf{m}\} \in \mathsf{Legal}(A)) \Rightarrow x \uplus \{\mathsf{m}\}$ *is dominated in* X.

There is a straightforward equivalence between the sequential definition of totality and its static one. More important is to prove that the total strategies are stable under composition. This relies on our arenas being finite in the sense that any play on them can only have a finite number of moves.

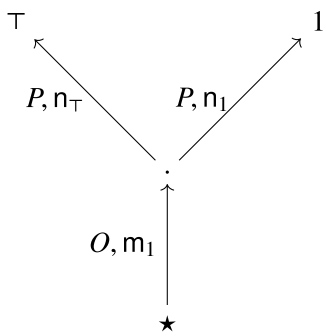
Proposition 5.39. *The total strategies of* Inn *form a sub-category of* Inn. *That is, the composite of two total strategies is total and the identity strategy is total.*

Proof. Consider two strategies $\sigma : A \triangleright B$ and $\tau : B \triangleright C$ such that both are total. Let $s : A \triangleright C \in \sigma;\tau$. Let m a O-move of $A \triangleright C$ such that $s.\mathsf{m}$ is legal. Consider t a sequence of $\sigma \mid \tau$ such that $t \upharpoonright A \triangleright C = s$. Suppose, without lost of generality, that $\mathsf{m} \in A$. Then, as σ is total, there is a m_2 such that $t \upharpoonright A \triangleright B.\mathsf{m}.\mathsf{m}_2 \in \sigma$. If $\mathsf{m}_2 \in A$, we conclude that $s.\mathsf{m}.\mathsf{m}_2 \in \sigma;\tau$. Otherwise, $\mathsf{m}_2 \in B$, where from τ point of view, it appears as a opponent move. Then τ reacts to it with m_3. If $\mathsf{m}_3 \in C$, the $s.\mathsf{m}.\mathsf{m}_3 \in \sigma;\tau$. Otherwise, it is in B where it appears as an O-move from σ point of view. Following this reasoning, we obtain a sequence $\mathsf{m}_1. \dots .\mathsf{m}_n$ of moves in B, until one of the strategies decides to play in A, C. As there are only finite chains in B, such a sequence always terminates, and one of the strategy σ, τ always reacts in A, C with m_{n+1} leading to a play $s.\mathsf{m}.\mathsf{m}_{n+1} \in \sigma; \tau$. Finally, the copycat strategy is obviously total. □

Therefore, total strategies form a sub-category of Inn.

Definition 5.40. *The category* TotInn *has same objects as* Inn *and morphisms total, typed-coherent, transverse, innocent strategies.*

It should however being noted that totality and maximality are two different notions. A total strategy might not reach a maximal position. For instance, let us consider the arena associated with the sequent $\neg 1 \vdash \neg 0$, whose simplified tree is displayed below:



This sequent has a proof, namely:

$$\dfrac{\dfrac{}{0, \neg 1 \vdash \bot}\ 0}{\neg 1 \vdash \neg 0}\ \text{Right } \neg$$

This proof corresponds to the total strategy $\{\emptyset, \mathsf{m}_1.\mathsf{n}_\top\}$, where the $\mathsf{n}_\top$ move is the denotation of the triggering of the 0-rule. However, this strategy does not reach the maximal position $\{\mathsf{m}_1, \mathsf{n}_\top, \mathsf{n}_1\}$. In linear logic, this would correspond to a proof of the sequent $\vdash 1, \top$.

5.3.1.1 On frugality

A play is frugal if opponent never introduces twice the same name. In particular, it never introduces twice the same typed name. As typed cells are maximal (they never justify another move), this can be formalised by focussing on cells available at the target of the play and of positive polarities.

Definition 5.41. *A play $s : \star \twoheadrightarrow x$ is **frugal** if it is legal and furthermore :*

$$\forall \alpha, \alpha' \in A_x^+ . \alpha \neq \alpha' \Rightarrow \alpha \#_T \alpha' \ .$$

This notably entails:

$$\forall \mathsf{m}_i, \mathsf{m}_j \in s.(i \neq j \wedge \lambda(\mathsf{m}_i) = \lambda(\mathsf{m}_j) = -1) \Rightarrow \ulcorner S(\mathsf{m}_i)\urcorner \# \ulcorner S(\mathsf{m}_j)\urcorner.$$

Concerning the cell names, the legality condition already ensures that each move by opponent brings different names. So the frugality only constraints the strategy regarding typed names. Given a strategy σ, we call $\mathsf{frugal}(\sigma)$ its set of frugal sequences.

$$\mathsf{frugal}(\sigma) = \{ s : \star \twoheadrightarrow x \mid \forall \alpha, \alpha' \in A_x^+ . \alpha \neq \alpha' \Rightarrow \alpha \#_T \alpha' \}$$

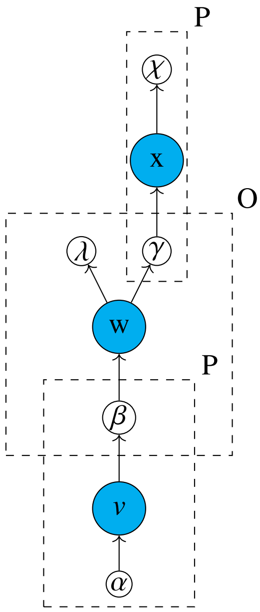
Equivalently, we say that a position is **frugal** if it is the target of a frugal sequence. Given $X = \sigma^\bullet$, we write $\mathsf{frugal}(X)$ for its set of frugal positions.

Just as we can project a proof of tensorial logic into a proof of linear logic, we would like to project our strategies to morphisms coming from a categorical model of linear logic. More specifically, we would like to project our strategies to the linear polarised nominal relational model. In it, the tensor is modelled by the polarised separated product, that does not let the same name appears twice in two different occurrences of atomic type of same polarity.

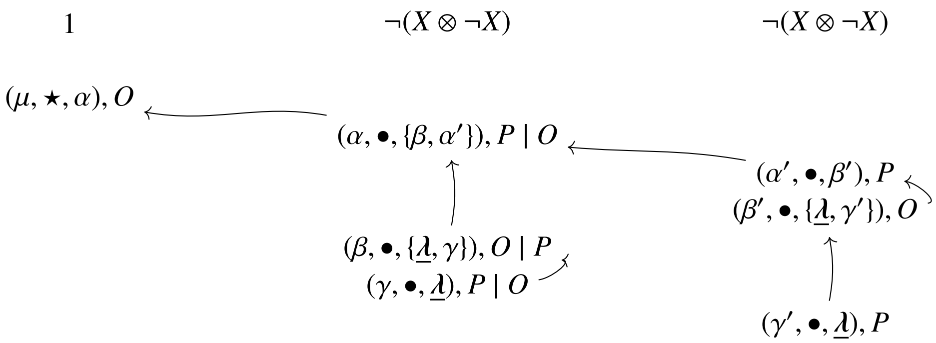
The reason why we do not enforce frugality at the level of strategies is that it does not compose well. This has to be put in relation with section 3.4.3, where we showed that the composition of separated polarised relations could not be defined as the usual relational composition. As example, let us consider the composition of the strategies corresponding to the following cut.

$$\dfrac{\dfrac{\dfrac{\dfrac{X \vdash X}{\neg X, X \vdash \bot}}{\neg X \otimes X \vdash \bot}}{1 \vdash \neg(\neg X \otimes X)} \qquad \dfrac{\dfrac{\dfrac{\dfrac{X \vdash X \qquad \dfrac{\dfrac{X \vdash X}{\neg X, X \vdash \bot}}{\neg X \vdash \neg X}}{X, \neg X \vdash X \otimes \neg X}}{X \otimes \neg X \vdash X \otimes \neg X}}{X \otimes \neg X, \neg(X \otimes \neg X}}{\neg(X \otimes \neg X) \vdash \neg(X \otimes \neg X)}}{1 \vdash \neg(X \otimes \neg X)} \text{Cut}$$

First, let us display the structure of the dialogue game corresponding to $\neg(X \otimes \neg X)$.



where λ, χ have type $\mathbb{A}_X$. The idea is that the proponent will copy the name introduced by opponent to display a copycat link. So if we look from the $(1 \triangleright \neg(X \otimes \neg X))$ point of view, that corresponds to the left hand side of the cut, player will simply copy the λ passed by opponent. On the other hand, in the arena $(\neg(X \otimes \neg X)) \triangleright (\neg(X \otimes \neg X))$, as player will also follow a copy-cat strategy, it will copy back and forth the λ. So from its point of view, it seems like opponent has played twice the same name, hence breaking the frugality condition. The play is displayed below, where the arrow between move emphasize the relation $\vdash$ between them, and where the typed names are bold and underlined.



To solve this issue, we have adopted the same technique as for linear relations: we have considered from the start strategies that are closed under typed nominal substitutions. Given σ be a set of plays, we define $\hat{\sigma}$ by:

$$\hat{\sigma} = \{e \cdot s \mid e \in \Xi_T,\ s \in \sigma\}$$

and equivalently, given X a set of positions, we define:

$$\hat{X} = \{e \cdot x \mid e \in \Xi_T,\ x \in X\}.$$

One can easily check that $(\widehat{\sigma})^\bullet = (\widehat{\sigma^\bullet})$. The first important lemma of this section states that our strategies are well-defined by their subset of frugal plays. We remind that a play is semi-linear if proponent never introduces a typed name in it, and a strategy is semi-linear if all the plays in it are. Semi-linearity is a sub-property of typed-coherency 5.2.

Proposition 5.42. *Let σ be an innocent, typed-coherent, total strategy. Then $\sigma = \widehat{\mathsf{frugal}(\sigma)}$.*

Proof. We start by the left to right inclusion. The proof is done by induction on the lengths of the plays s of σ. Suppose $s = s'.\mathsf{m}.\mathsf{n}$, and $s' = e_1 \cdot t'$, where $t' \in \mathsf{frugal}(\sigma)$, and the only action of e_1 is to merge names of t'. That is $\nu(s') \subseteq \nu(t')$, and $\nu(e_1) \subseteq \nu_T(t')$. Furthermore, let us assume that $(\nu_T(t') \setminus \nu_T(s')) \cap \nu_T(\mathsf{m}) = \emptyset$ without loss of generality (one can always do this assumption since if there were some names in this set, one could get rid of them by applying a typed permutation to t'). In the case where $t'.\mathsf{m}.\mathsf{n}$ is frugal, then, as the only action of e_1 is to map names of $\nu_T(t')$ to names of $\nu_T(s')$, and $(\nu_T(t') \setminus \nu_T(s')) \cap \nu_T(\mathsf{m}) = \emptyset$, we can conclude that $e_1 \cdot \mathsf{m} = \mathsf{m}$. Furthermore, by typed-coherence, $\nu_T(\mathsf{n}) \subseteq (\nu_T(s') \cup \nu_T(\mathsf{m}))$. Therefore, $e_1 \cdot \nu_T(\mathsf{n}) = \mathsf{n}$. Finally, let us assume that $t'.\mathsf{m}.\mathsf{n}$ is not frugal. Then this entails that m brings names appearing in t', or brings several cells filled with the same name. In either case, we pick a m' such that m' is frugal as a move, $\mathsf{m}' \#_T s, t$, and such that there is a typed substitution e_2 such that $e_2 \cdot \mathsf{m}' = \mathsf{m}$, and $\nu(e_2) \subseteq \nu_T(\mathsf{m}) \cup \nu_T(\mathsf{m}')$. Finally, let us consider a n_1 such that $t.\mathsf{m}'.\mathsf{n}_1 \in \sigma$. Such a move n_1 exists since σ is total. Now, $e_1.e_2 \cdot (t.\mathsf{m}'.\mathsf{n}_1) = s.\mathsf{m}.(e_1.e_2) \cdot \mathsf{n}_1 \simeq s.\mathsf{m}.\mathsf{n}$ by nominal determinacy. Let us consider $\pi \in \mathsf{Perm}(\mathbb{A})$ such that $\pi \cdot (s.\mathsf{m}.(e_1.e_2) \cdot \mathsf{n}_1) = s.\mathsf{m}.\mathsf{n}$. As σ is semi-linear $\nu_T(e_1.e_2 \cdot \mathsf{n}_1) \subseteq \nu_T(s.\mathsf{m})$. Moreover, as $\mathsf{n}_1, s.\mathsf{m}$ have strong support, we deduce that $\nu(\pi) \# \nu_T(s.\mathsf{m})$ and we conclude that π is a permutation of $\mathbb{A}_{\mathsf{cells}}$. As π and e_1, e_2 have different supports they commute. Finally, calling $\mathsf{n}' = \pi \cdot \mathsf{n}$, we got $t.\mathsf{m}'.\mathsf{n}' \in \mathsf{frugal}(\sigma)$ and $e_1.e_2 \cdot (t.\mathsf{m}'.\mathsf{n}') = s$.

The reverse inclusion is automatic, since $\mathsf{frugal}(\sigma) \subseteq \sigma$, and σ is closed under typed substitutions. □

One now can define a category with nominal positive games as objects and innocent, semi-linear, total, frugal strategies as morphisms, where the composition is defined through closure under typed substitutions as in section 3. Given a typed-coherent innocent strategy, its frugal restriction is not innocent anymore, but almost. To remedy for that, we introduce frugal innocence.

Definition 5.43. *A strategy is **frugal forward consistent**, if $\forall s \in \sigma$ and $\mathsf{m}_1, \mathsf{m}_2, \mathsf{n}_1, \mathsf{n}_2$ such that $s.\mathsf{m}_1.\mathsf{n}_1,\ s.\mathsf{m}_2.\mathsf{n}_2 \in \sigma$, $\mathsf{m}_1 \neq \mathsf{m}_2$, $\mathsf{m}_1 \uparrow \mathsf{m}_2$ then $\mathsf{n}_1 \not\equiv \mathsf{n}_2$. Moreover, if $\mathsf{m}_1, \mathsf{n}_1$ are such that $s.\mathsf{m}_1.\mathsf{n}_1\ C_{\mathsf{post}}\ s.\mathsf{m}_2.\mathsf{n}_2$ and $\mathsf{m}_1.\mathsf{n}_1 \# \mathsf{m}_2.\mathsf{n}_2$ then we have $s.\mathsf{m}_1.\mathsf{n}_1 \uparrow s.\mathsf{m}_2.\mathsf{n}_2 \in \sigma$ and $s.\mathsf{m}_1.\mathsf{n}_1.\mathsf{m}_2.\mathsf{n}_2 \in \sigma$. A strategy is **frugal innocent** if it is frugal forward consistent and backward consistent.*

We then have the following straightforward property.

Proposition 5.44. *A strategy σ is typed coherent innocent if and only if $\mathsf{frugal}(\sigma)$ is semi-linear frugal innocent. Furthermore, σ is transverse, total if and only if $\mathsf{frugal}(\sigma)$ is.*

The proof is obvious. This allows us to define the following category.

Definition 5.45. Frugal *is the category with objects positive dialogue games and morphisms frugal innocent, semi-linear, total, transverse, frugal strategies. The composition of morphisms is defined as follows:*

$$\sigma;\tau = \mathsf{frugal}(\widehat{\sigma};\widehat{\tau})$$

or, equivalently,

$$X;Y = \mathsf{frugal}(\hat{X} ;_{\mathsf{Rel}} \hat{Y})$$

For completeness, we prove below that the above definition makes sense.

Proposition 5.46. Frugal *is well-defined, that is, if σ, τ are frugal innocent, semi-linear, total, transverse, frugal strategies, then so is $\sigma;\tau$.*

Proof. As σ, τ are frugal innocent, semi-linear, total, transverse, frugal strategies, then $\widehat{\sigma}, \widehat{\tau}$ are morphisms of TotInn. Therefore, $\widehat{\sigma};\widehat{\tau}$ is a morphism of TotInn and $\mathsf{frugal}(\widehat{\sigma};\widehat{\tau})$ is a frugal innocent, semi-linear, total, transverse, frugal strategy. □

Associativity of composition follows from the one of typed-coherent innocent strategies, and the identity morphisms are defined to be $\mathsf{id}_A = \mathsf{frugal}(\mathsf{copycat}_A)$. Equivalently, a direct method can be proposed, by relating arenas to lists, strategies with semi-linear nominal relations, and the closure under nominal permutations with tracing of partial injective functions. More about this is given in the appendix 9.2.

Therefore, in the next paragraph, we focus on strategies of Frugal.

5.3.2 Innocent strategies and strong structures of sequentiality

Sequentiality structures have been introduced in [69], and we simply adapt them to take care of our alternative nominal structure, notably through the "well-typed condition", that is our main contribution. Similarly, the properties to be found here are mostly adaptations of the ones presented in the original paper.

The key to definability for strategies lies in their sequentiality structures. Indeed, each move by player will correspond to a sequence $\bigoplus \bigotimes$ of global positive connectives, and in particular, the $\bigotimes$ splits the formulas on the left hand side of the sequent into a partition. The division of context taking place at the level of formulas in the proof corresponds to the division of context taking place at the level of cells by the strategy. However, the problem with innocent strategies is that their sequentiality structures are weak: their functions are partial. Notably, given a cell with no move above, (corresponding to a unit), there is no way to establish what part of the context will be captured by it. Another way to state it, is that there is no difference between ¬0 and 1 from the strategy point of view, both are interpreted by a single player move. There is no indication for the fact that ¬0 might capture a context, while 1 cannot. Equivalently, we have to enforce the fact that a proponent move in X can only capture a context of type X. In order to solve that, we extend the notion of innocent strategy to equip them with a "strong" sequentiality structure, that is not partial anymore. If the innocence of the strategy captures the underlying structure of the proof, then the sequentiality structure makes sure it stays logical, and, in particular, captures the structure of the leaves (axiom and 0-rule). .

Let us recall that we work within the category Frugal, as defined in the above section. Therefore, the strategies are not closed under typed substitutions anymore.

Definition 5.47. *A strategy with sequentiality structure (σ, ϕ) is an innocent strategy σ together with a family of sequentiality functions $\phi = \{\phi_x : A_x^+ \to A_x^- \mid x \in \sigma^\bullet\}$ such that :*

- *ϕ is closed under permutations: $\phi_{\pi \cdot (x)} = \pi \cdot \phi_x$.*
- *ϕ_x is a total function.*
- *$\forall s : \star \twoheadrightarrow x \xrightarrow{m} \xrightarrow{n} y \in \sigma$, for all $\alpha \in A_x^- \cap A_y^-. \phi_x^{-1}(\alpha) = \phi_y^{-1}(\alpha)$.*

This definition of strong sequentiality structure is coherent with the definition of weak sequentiality structure given in the context of innocent strategies. We recall that the action of name permutations on functions is defined by:

$$(\pi \cdot \phi)(x) = \pi \cdot (\phi(\pi^{-1} \cdot x))$$

Proposition 5.48. *Let (σ, ϕ) be an innocent strategy with sequentiality structure, and ψ the weak sequentiality structure canonically associated to σ. Then $\psi \subseteq \phi$ (where the inclusion takes place at the level of partial functions).*

Proof. Let x be a position of $\sigma^\bullet$, α a negative cell available at x, and m an opponent move above α. Moreover, let us consider β, a positive cell available at x, such that there is a play in $\sigma \mid_x \alpha$ that contains a move n above β. That is, $\psi(\beta) = \alpha$. Let $s = \mathsf{m}.s'\mathsf{n}$ a play of $\sigma \mid_x \alpha$ that starts with m and finishes with n. Consider all the negative cells available at x but α. By definition of $\sigma \mid_x \alpha$, they are not explored by the play s, and therefore remain available all along it. Consequently, the set of cells $\phi_x^{-1}(A_x^- \setminus \alpha)$ is still present at y, the target position of $\mathsf{m}.s'.\mathsf{n}$. As n explores above β, β is not available at y and therefore $\beta \notin \phi_x^{-1}(A_x^- \setminus \alpha)$. As the function is total, this entails $\beta \in \phi_x^{-1}(\alpha)$, that is, $\phi_x(\beta) = \alpha$. □

Since our new sequentiality structure is an extension of the canonical one imposed by the innocence of the strategy, it approximately satisfies the same properties. Among them, Proposition 5.23 remains true. That is, given $s : \star \twoheadrightarrow x \in \sigma$, if $\phi_x(\alpha) = \beta$, (where $\alpha, \beta \in A_x$), then $\|s\|_\alpha < \|s\|_\beta$. The proof is exactly the same as the one of the original proposition 5.23.

We furthermore need to make sure that the context captured by a cell is coherent. That is, an untyped cell can capture any context, but a typed cell can only capture a context of the same type, and, by linearity, only a single cell. We therefore impose the following conditions.

Definition 5.49. *Given (σ, ϕ) a strategy with sequentiality structure, we say that ϕ is* ***well-typed*** *if:*

- $\forall x \in \sigma^\bullet$, $\forall \alpha \in A_x^+$, $\nu(\ulcorner\alpha\urcorner) \in \mathbb{A}_T \Rightarrow (\forall \pi \in \mathsf{Perm}(\mathbb{A}_T).\nu(\ulcorner\phi_{\pi\cdot x}(\pi\cdot\alpha)\urcorner) = \pi\cdot\nu(\ulcorner(\phi_x(\alpha))\urcorner))$. *That is* $\nu(\ulcorner\phi_x(\alpha)\urcorner) \cap \mathbb{A}_T \subseteq \nu(\ulcorner\alpha\urcorner)\cap\mathbb{A}_T$.
- $\forall x \in \sigma^\bullet$, $\forall \alpha \in A_x^-.(\nu(\ulcorner\alpha\urcorner)\cap\mathbb{A}_T \neq \emptyset \Rightarrow |\phi_x^{-1}(\alpha)| = 1)$.

One important consequence of this definition is that player cannot play a move in an atomic formula X if opponent has not played in another X before. Then, player will play the same name α as opponent played before, establishing a copy-cat link. In other terms, the copy-cat links are oriented, from negative to positive literals. The strategies with well-typed sequentiality structures are automatically semi-linear. Indeed, if player plays a cell with a name of $\mathbb{A}_T$, then from the first condition, it points to a positive cell of same type through ϕ, and this cell has the same name. Finally, as we know that sequentiality structures are such that if $\phi(\alpha) = \beta$ then α appears before β, then we can devise that if proponent plays a typed name, then this one was brought by opponent before. Finally, as the strategy is frugal, opponent never plays twice the same name. Now, if player would want to play twice the typed same name, then it would mean that there would be two positive cells that point to it through ϕ. However, that is prevented thanks to the second condition.

Note that, as in a vertex v, only the final cell $\ulcorner v\urcorner$ can be a typed cell, then it is equivalent to require $\nu(\ulcorner\alpha\urcorner)\cap\mathbb{A}_T \neq \emptyset$ and $\nu(\alpha)\cap\mathbb{A}_T\neq\emptyset$. This entails this small simplification.

Lemma 5.50. *The well-typed conditions are equivalent to the following ones:*

- $\forall x \in \sigma^\bullet.\forall\alpha\in A_x^+.\nu(\phi_x(\alpha))\cap\mathbb{A}_T \subseteq \nu(\alpha)\cap\mathbb{A}_T$.
- $\forall x \in \sigma^\bullet.\forall\alpha\in A_x^-.(\nu(\alpha)\cap\mathbb{A}_T\neq\emptyset \Rightarrow |\phi_x^{-1}(\alpha)| = 1)$.

5.3.2.1 Composition of sequentiality structures

To start this section, let us recall that weak sequentiality structures compose through tracing, but not exactly. That is, the composition of two weak sequentiality structures ϕ and ψ along a position of interaction (x, y, z) is defined at the level of the strategy, and is included in:

$$\mathsf{Tr}^{B_y^+\uplus B_y^-}_{A_x^-,C_z^+,A_x^+,C_z^-}((\varphi_{(x,y)}\uplus\psi_{(y,z)});(\mathsf{id}_{A_x^+}\uplus s_{B_y^-,B_y^+}\uplus\mathsf{id}_{C_z^-})) : A_x^-\uplus B_y^+\uplus B_y^-\uplus C_z^+ \to A_x^+\uplus B_y^+\uplus B_y^-\uplus C_z^-)$$

the inclusion being, in the general case, strict. After a brief discussion, it was highlighted that the reason for this strictness might be pointed to a wrong logical behaviour with regard to the &-rule by innocent strategies :

$$\frac{\Gamma_1 \vdash A \qquad \Gamma_2 \vdash B}{\Gamma_1 \cup \Gamma_2 \vdash A \,\&\, B}$$

in linear logic, or

$$\frac{\Gamma_1, A \vdash \bot \qquad \Gamma_2, B \vdash \bot}{\Gamma_1 \cup \Gamma_2, A \oplus B \vdash \bot}$$

in tensorial logic.

This behaviour follows from the fact that a definition of sequentiality structure only through "what is explored" is not strong enough. Our new definition does not encounter this problem, what is explored by the strategy on the left or on the right might differ, but this is not taken into account by the structure.

As expected, composition of sequentiality structures is defined through tracing. One already knows that this is coherent with their restriction to weak sequentiality structure. That is, given ϕ, ψ sequentiality structures associated with two strategies $\sigma : A \to B$ and $\tau : B \to C$, we already know that the weak sequentiality structure associated with $\sigma;\tau$ is a family of sub-functions of $\mathsf{Tr}(\phi \uplus \psi; \mathsf{id} \uplus s \uplus \mathsf{id})$.

Definition 5.51. *Let $(\sigma, \varphi) : A \to B$ and $(\tau, \psi) : B \to C$ two strategies with sequentiality structures. We define their composition $(\sigma, \varphi); (\tau, \psi)$ by $(\sigma; \tau, \varphi;_{\mathsf{seq}} \psi)$, where given a legal position (x, y, z) of $\sigma^\bullet \mid_{\mathsf{Rel}} \tau^\bullet \subseteq \mathsf{Pos}(A \triangleright B \triangleright C)$:*

$$(\varphi;_{\mathsf{seq}} \psi)_{(x,z)} = \mathsf{Tr}^{B_y^+ \uplus B_y^-}_{A_x^-, C_z^+, A_x^+, C_z^-}(\varphi_{(x,y)} \uplus \psi_{y,z}; (\mathsf{id}_{A_x^+} \uplus s_{B_y^-, B_y^+} \uplus \mathsf{id}_{C_z^-})) : A_x^- \uplus B_y^+ \uplus B_y^- \uplus C_z^+ \to A_x^+ \uplus B_y^+ \uplus B_y^- \uplus C_z^-$$

We need to prove that this leads to a sequentiality structure, that is, that the thus defined function is total. Furthermore, we prove that if the two sequentiality structures are well-typed, then so is their composition. Finally, it is necessary to check that two distinct witnesses of interaction lead to the same final sequentiality structure.

Lemma 5.52. *The function*

$$\mathsf{Tr}^{B_y^+ \uplus B_y^-}_{A_x^-, C_z^+, A_x^+ C_z^-}(\varphi_{(x,y)} \uplus \psi_{y,z}; (\mathsf{id}_{A_x^+} \uplus s_{B_y^-, B_y^+} \uplus \mathsf{id}_{C_z^-})) : A_x^- \uplus B_y^+ \uplus B_y^- \uplus C_z^+ \to A_x^+ \uplus B_y^+ \uplus B_y^- \uplus C_z^-$$

is total.

Proof. We simply need to show that the partial function $\varphi_{x,y} \uplus \psi_{y,z} \upharpoonright B_y^- \uplus B_y^+; s_{B_y^-, B_y^+}$ is nilpotent. It is true since, given $s : \star \twoheadrightarrow (x, y, z)$, and the function $||.||$ that to each cell α in B gives the length of the minimal subsequence $s' \leq s \uparrow B$ such that s' introduces α, then $||\varphi_{x,y}(\alpha)|| < ||\alpha||$, and equally for ψ. Therefore, given any cell of B_y, there is no infinite chain following repetitive applications of the function $\phi_{x,y} \uplus \psi_{y,z} \upharpoonright B_y^- \uplus B_y^+; s_{B_y^-, B_y^+}$. □

Lemma 5.53. *If ϕ, ψ are well-typed, then so is their composition.*

Proof. We only need to show that the two properties defining well-typedness are stable under iterated applications of ϕ and ψ (as given any cell $\alpha \in A^{+}_{(x,z)}$, there is a sequence of alternating ϕ, ψ such that $(\phi;_{\mathsf{seq}} \psi)(\alpha) = ..\phi\circ\psi\circ...(\alpha)$). Let us assume without loss of assumption that the last function of the previous sequence is ϕ. Then $\nu(\alpha)\cap\mathbb{A}_T \supseteq \nu(\phi_{(x,y)}(\alpha))\cap\mathbb{A}_T \supseteq \nu(\psi_{(y,z)}\circ\phi_{(x,y)}(\alpha))\cap \mathbb{A}_T \supseteq ... \supseteq (\phi_{x,y};_{\mathsf{seq}} \psi_{(y,z)}(\alpha)$. We proceed similarly to show that $|\phi_{(x,y)};_{\mathsf{seq}} \psi^{-1}_{(y,z)}(\alpha)| = 1$ when $\alpha \cap \mathbb{A}_T \neq \emptyset$. □

Finally, before being able to conclude, we need a final lemma.

Lemma 5.54. *Let $y \neq y'$ such that $(x, y, z), (x, y', z) \in \sigma^{\bullet} |_{\mathsf{Rel}} \tau^{\bullet}$. Then:*

$$\begin{aligned}&\mathsf{Tr}^{B^+_y \uplus B^-_y}_{A^+_x, C^-_z, A^-_x, C^+_z}((\varphi_{(x,y)} \uplus \psi_{y,z}); (\mathsf{id}_{A^+_x} \uplus s_{B^-_y, B^+_y} \uplus \mathsf{id}_{C^-_z}))\\ = &\mathsf{Tr}^{B'^+_y \uplus B'^-_y}_{A^+_x, C^-_z, A^-_x, C^+_z}((\varphi_{(x,y')} \uplus \psi_{y',z}); (\mathsf{id}_{A^+_x} \uplus s_{B'^-_y, B'^+_y} \uplus \mathsf{id}_{C^-_z}))\end{aligned}$$

Proof. We know that there is a unique witness of interaction, up to cell nominal permutations. Hence $y \simeq_{\mathsf{cells}} y'$. Furthermore, as (x, y) is legal, $x\#_{\mathsf{cells}}y$, and similarly $y\#_{\mathsf{cells}}z$. So it follows that there exists a permutation π such that $\pi \cdot (x, y, z) = (x, y', z)$. Then the lemma follows from ϕ, ψ being nominal, as well of the trace operator. That is, given $f : y \times x \to z \times x$ a generic function:

$$\begin{aligned}\mathsf{Tr}^x_{y,z}(f)(\alpha) &= (\pi \cdot \mathsf{Tr})^x_{y,z}(f)(\alpha)\\ &= \pi \cdot (\mathsf{Tr}^{\pi^{-1}\cdot x}_{\pi^{-1}\cdot y, \pi^{-1}\cdot z}(\pi^{-1} \cdot f))(\alpha)\\ &= \pi \cdot (\mathsf{Tr}^{\pi^{-1}\cdot x}_{\pi^{-1}\cdot y, z}(\pi^{-1} \cdot f)(\pi^{-1} \cdot \alpha))\end{aligned}$$

In our case,

$$\begin{aligned}f &= \varphi_{(x,y)} \uplus \psi_{(y,z)}; (\mathsf{id}_{A^+_x} \uplus s_{B^-_y, B^+_y} \uplus \mathsf{id}_{C^-_z})\\ \pi^{-1} \cdot f &= \varphi_{\pi^{-1}\cdot(x,y)} \uplus \psi_{\pi^{-1}\cdot(y,z)}; (\mathsf{id}_{A^+_{\pi^{-1}\cdot x}} \uplus s_{B^-_{\pi^{-1}\cdot y}, B^+_{\pi^{-1}\cdot y}} \uplus \mathsf{id}_{C^-_{\pi^{-1}\cdot z}})\end{aligned}$$

So let us pick π such that $\pi \cdot (x, y, z) = (x, y', z)$. Then as α is a cell of (x, z) and $\pi \cdot (x, z) = (x, z)$, we got $\pi^{-1} \cdot \alpha = \alpha$. Finally, to simplify, writing $\mathsf{Tr}_{x,y,z}(\phi_{x,y} \mid \psi_{y,z})$ for $\mathsf{Tr}^{B^+_y \uplus B^-_y}_{A^-_x, C^+_z, A^+_x, C^-_z}((\varphi_{(x,y)} \uplus \psi_{y,z}); (\mathsf{id}_{A^+_x} \uplus s_{B^-_y, B^+_y} \uplus \mathsf{id}_{C^-_z}))$ we got:

$$\begin{aligned}\mathsf{Tr}_{x,y'z}(\phi_{(x,y')} \mid \psi_{(y',z)})(\alpha) &= \mathsf{Tr}_{\pi\cdot(x,y,z)}(\phi_{\pi\cdot(x,y)} \mid \psi_{\pi\cdot(y,z)})(\pi \cdot \alpha)\\ &= \mathsf{Tr}_{\pi\cdot(x,y,z)}(\pi \cdot \phi_{(x,y)} \mid \pi \cdot \psi_{(y,z)})(\pi \cdot \alpha)\\ &= \pi \cdot (\mathsf{Tr}_{x,y,z}(\phi_{(x,y)} \uplus \psi_{(y,z)})(\alpha))\\ &= \mathsf{Tr}_y(\phi_{(x,y)} \uplus \psi_{(y,z)})(\alpha)\end{aligned}$$

where the last equality holds since $\mathsf{Tr}_{x,y,z}(\phi_{(x,y)} \uplus \psi_{(y,z)})(\alpha)$ is a cell of (x, z). □

So finally we can conclude that we have a category of arena games and transverse, innocent, total, frugal strategies with well-typed sequentiality structures as morphisms. These strategies will prove to be sound and fully complete for tensorial logic. We name this category $\mathsf{TTSFInn}$.

Definition 5.55. TTSFInn *is the category with objects positive dialogue games and morphisms transverse, total, semi-linear, frugal, innocent strategies with well-typed strong sequentiality structures on negative dialogue games $A \triangleright B$. We will call such strategies **brave**.*

We will explore this category in the next section. We will notably present the associated copycat morphisms, and will expose its categorical structure.

Chapter 6

Full Completeness

Full completeness was first introduced in [5] for multiplicative linear logic (with mix rule). A model is fully complete if it is sound and every morphism corresponds to a proof. This amounts to a full functor from the category of proofs (modulo equivalence) to the categorical model investigated, such that every object in the categorical model lies in the image of the functor.

However, in that seminal paper, just as in many others [60, 50, 24], the functor was not only full but also faithful. That is, there was an equivalence of categories. In that case, there is a one to one correspondence between proofs modulo equivalence and the morphisms of the category. These results were all obtained following the same recipe. They relied on known canonical representations of proofs modulo equivalence for several fragments of linear logic, all under the form of proof nets. This was discussed in 2.2.3. One can then establish an equivalence between the proof nets and the morphisms of the category. In this case, the model is an instance of a free category on the discrete category VAR. In our case, we will deal with tensorial logic, and we will rely on $\mathsf{TENS}_{\mathsf{foc-glob}}$ to distinguish between classes of equivalence of proofs. We recall that $\mathsf{TENS}_{\mathsf{foc-glob}}$ is the focalised fragment of tensorial logic with global connectives, introduced in 2.3.2.1.

In the first section of this chapter, we present the interpretation of tensorial logic into our category TTSFInn, defined at the end of the previous chapter in definition 5.55, and prove that TTSFInn is a sound model for it. Furthermore, we prove that there is a one-to-one correspondence between the morphisms of TTSFInn and the proof invariants of tensorial logic. That is, TTSFInn is an instance of the free dialogue category with sums on the discrete category VAR.

Following this result, we recall that tensorial logic and linear logic are strongly linked, especially at a syntactic level. As each proof of tensorial logic can be seen as a proof of linear logic (and reversely), we want to use the category TTSFInn to interpret linear logic. Once done, we define a mapping from this interpretation to the nominal relations. This allows us to prove a full completeness result for MALL, based on a refinement of linear nominal polarised relations. However, the final model obtained this way is unfortunately not the free star-autonomous category with products on VAR. That is, the functor from the proofs to the categories of relations is only full, but not faithful: two proofs might be the interpreted by the same morphism.

Once again, this work is mainly an adaptation of the work that Melliès carried in [66, 69, 73]. In [66], Melliès did introduce a full-completeness for linear logic without axiom-links following a similar recipe, however relying on a notion of payoff instead of sequentality structures. This one also ecompassed exponentials, whereas we don't consider them here. In [69], the result was re-established using sequentiality structures (this time, without exponentials) instead of payoff. In [73], a model of the free-multiplicative dialogue category (hence, corresponding to axiom-links in the case of VAR) was presented, however this one did not include the additive fragment. Here, we display the free dialogue (with additive structure) category over VAR, and prove that the projection extends indeed the original full-completeness result for multiplicative-additive linear logic to include axiom-links.

6.1 Interpretation of proofs and soundness

In order to prove that we can soundly interpret tensorial logic in TTSFInn, we need to prove that TTSFInn forms a dialogue category with sums, with the ability to model the axiom rule on literals. We remind that $A \triangleright B = (A \otimes \neg B)^*$. We present below the pre-arena $X \triangleright X$, where X is an atomic variable.

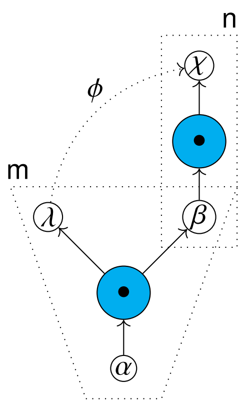


Figure 6.1: Dialogue game of $X \triangleright X$

Then the strategy σ interpreting the axiom is the strategy with maximal plays m.n, $\ulcorner \mathsf{m} \urcorner . \ulcorner \mathsf{n} \urcorner = (\alpha, \bullet, \{\mathsf{inl}(\lambda), \mathsf{inr}(\beta)\}).(\mathsf{inr}(\beta), \bullet, \{\lambda\})$, and with sequentiality structure $\phi(\lambda) = \lambda$. It is a transverse, total, frugal strategy with strong sequentiality structure. We recall that such a strategy is called brave. We remind that a frugal play is legal. That is, frugality entails legality.

More generally, we present the copy-cat strategy, that acts as the identity in the TTSFInn category. Given an arena A, we define the brave strategy $\mathsf{copycat}_A : A_1 \triangleright A_2$ as follows:

$$\mathsf{copycat}_A = \{s \in \mathsf{Play}(A) \mid s \text{ frugal, alternated, even-length and } s \upharpoonright A_1 \simeq_{\mathsf{cells}} s \upharpoonright A_2\}.$$

The semi-linearity and totality are automatic. The $\mathsf{copycat}_A$ strategy can also be described through its set of positions:

$$\mathsf{copycat}^{\bullet}_A = \{(x, y) \in \mathsf{Frugal}(\mathsf{Trans}(A \triangleright A)) \mid x \simeq_{\mathbb{A}_{\mathsf{cells}}} y\}$$

Finally, its structure of sequentiality is described as the identity function between negative and positive cells. That is, given $(x, y) \in \mathsf{copycat}_A$, and $\pi \in \mathsf{Perm}(\mathbb{A}_{\mathsf{cells}})$ such that $\pi \cdot x = y$, then $\phi_{x,y} : A^+_y \uplus A^-_x \to A^+_x \uplus A^-_y$ is defined as follows:

$$\alpha \in A^-_y \Rightarrow \phi_{x,y}(\alpha) = \pi^{-1} \cdot \alpha \in A^+_x$$
$$\alpha \in A^+_x \Rightarrow \phi_{x,y}(\alpha) = \pi \cdot \alpha \in A^-_y$$

Then given $\sigma : A \triangleright B$, we have $\sigma; \mathsf{copycat}_B = \sigma$ and $\mathsf{copycat}_A; \sigma = \sigma$.

We deal in the sub-sections below with the categorical structure. Several distinct steps are needed. First, we prove that the category is monoidal. Then, we establish the existence of a natural isomorphism between $C(A \otimes B, \bot)$ and $C(A, \neg B)$, allowing us to conclude that it is a dialogue category. Finally, we tackle the coproduct. We also take advantage of this presentation to expose the denotation function from proofs to strategies.

6.1.1 TTSFInn is monoidal

The monoidal functor is, of course, defined by $\otimes$ on objects. Given two morphisms $\sigma : A \to C$ and $\tau : B \to D$, that is, two strategies $\sigma : (A \triangleright C)$ and $\tau : (B \triangleright D)$, one needs to define a strategy $\sigma \otimes \tau : (A \otimes B \triangleright (C \otimes D))$. We present in figure 6.2 the structure of the arena $(A \otimes B \otimes \neg(C \otimes D))$, using the standard decomposition of each of the arena A, B, C, D into a sum of simple games; $A = \bigoplus A_i$ for instance. We denote by a_i (resp b_i, c_i, d_i) the initial values of A_i (resp B_i, C_i, D_i).

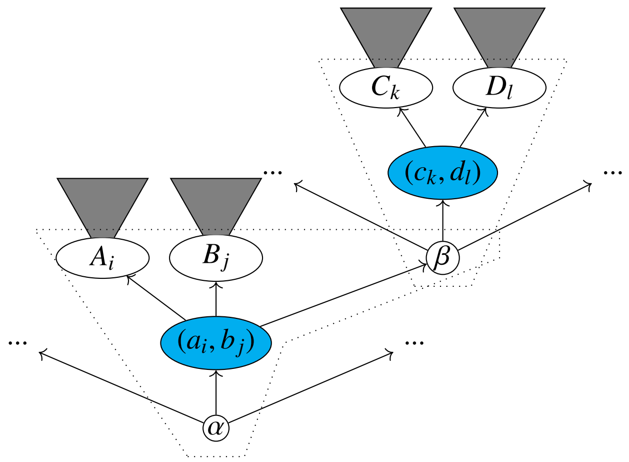


Figure 6.2: Structure of $A \otimes B \triangleright C \otimes D$.

Then, one can see that, as the strategy $\sigma \otimes \tau$ is transverse, the two first moves will be of the shape $(\alpha, (a_i b_i), S_{A_i} \uplus S_{B_j} \uplus \beta).(\beta, (c_i, d_i), S_{C_k} \uplus S_{D_l})$, so are isomorphic to $(\mathsf{m}_1, \mathsf{m}_2).(\mathsf{n}_1, \mathsf{n}_2)$, where $\mathsf{m}_1.\mathsf{n}_1 \in \sigma$ and $\mathsf{m}_2.\mathsf{n}_2 \in \tau$. Furthermore, after two moves, the cells available are $S_{A_i} \uplus S_{B_i} \uplus S_{C_i} \uplus S_{D_i}$, and the sequentiality structure yields a function $\phi_{\{\mathsf{m}_1 \uplus \mathsf{n}_1\}} \uplus \psi_{\{\mathsf{m}_2 \uplus \mathsf{n}_2\}}$, where ϕ is the sequentiality structure associated with σ, and ψ the one associated with τ. Furthermore $\phi_{\mathsf{m}_1.\mathsf{m}_2}$ has domain of definition and image in $S_{A_i} \uplus S_{C_k}$, whereas $\psi_{\mathsf{n}_2.\mathsf{n}_2}$ acts upon $S_{B_j} \uplus S_{D_l}$.

We make that precise using the equations from section 4.5.4.

$$\begin{aligned}
\mathsf{Trans}(A \otimes B \triangleright C \otimes D) &\simeq \mathsf{Pos}(A \otimes B) \,\overline{\otimes}\, \mathsf{Pos}(C \otimes D) \\
&\simeq (\mathsf{Pos}(A) \,\overline{\otimes}_{\,\mathbb{A}_{\mathsf{cells}}} \mathsf{Pos}(B)) \;\overline{\otimes}\; (\mathsf{Pos}(C) \,\overline{\otimes}_{\,\mathbb{A}_{\mathsf{cells}}} \mathsf{Pos}(D)) \\
&\simeq (\mathsf{Pos}(A) \,\overline{\otimes}\, \mathsf{Pos}(C)) \;\overline{\otimes}_{\,(\mathbb{A}_{\mathsf{cells}} \times \mathbb{A}_{\mathsf{cells}})}\; (\mathsf{Pos}(B) \,\overline{\otimes}\, \mathsf{Pos}(D)) \\
&\simeq \mathsf{Trans}(A \triangleright C) \;\overline{\otimes}_{\,(\mathbb{A}_{\mathsf{cells}} \times \mathbb{A}_{\mathsf{cells}})}\; \mathsf{Trans}(B \triangleright D)
\end{aligned}$$

So given $\sigma : A \triangleright C$, $\tau : B \triangleright C$, we define $\sigma \otimes \tau$ first by looking at its set of positions, where we remind that ϖ_1 is the function on positions that to a position returns the cell it is rooted on.

$$\begin{aligned}
(\sigma \otimes \tau)^\bullet &\simeq \mathsf{Frugal}(\mathsf{Legal}((\sigma^\bullet \,\overline{\otimes}_{\,(\mathbb{A}_{\mathsf{cells}} \times \mathbb{A}_{\mathsf{cells}})} \tau^\bullet)) \\
&= (\bot_{A \triangleright C} \times \bot_{B \triangleright D}) \;\uplus\; \mathsf{Frugal}(\{((x_A, x_C), (y_B, y_D)) \mid \varpi_1(x_1) = \varpi_1(y_B),\ \varpi_1(x_C) = \varpi_1(y_D), \\
&\qquad (x_A, x_C) \in \sigma^\bullet \setminus \{\bot\}, (y_B, y_D) \in \tau^\bullet \setminus \{\bot\}, (\nu(x_A, x_C) \setminus \nu(\varpi_1(x_A), \varpi_1(x_C)) \#_{\mathbb{A}_{\mathsf{cells}}} (y_B, y_D)\})
\end{aligned}$$

Equivalently, it can be presented by the set of plays:

$$\sigma \otimes \tau = \{s \in A \otimes B \triangleright C \otimes D \mid s \text{ frugal, even-length, alternated}, s \upharpoonright A \triangleright C \in \sigma, s \upharpoonright B \triangleright D \in \tau\}$$

Finally, any position of $(\sigma \otimes \tau)$ being isomorphic to a position (x, y), where $x \in \sigma^\bullet$ and $y \in \tau^\bullet$, one can describe the structure of sequentiality straightforwardly by :

$$\phi_{\sigma \otimes \tau,\, (x,y)} \simeq \phi_{\sigma,x} \uplus \phi_{\tau,y}.$$

The strategy hence obtained is frugal by definition, transverse and with a strong sequentiality structure. Innocence can be straightforwardly checked following innocence of σ, τ, and relying on the property that every two moves, except the two initial ones, that happen in $A \triangleright B$ and $C \triangleright D$ respectively, are independent when projecting in $A \otimes B \triangleright C \otimes D$.

As the arena for $A \otimes (B \otimes C)$ is the same as the one for $(A \otimes B) \otimes C$ (by associativity of $\overline{\otimes}_{\,\mathbb{A}_{\mathsf{cells}}}$), the monoidal product satisfies associativity as required. What remains is to tackle the units. Given an arena A, we have to give description of the morphisms $A \to A \otimes I$ and $A \otimes I \to A$ satisfying the required equations. However, $A \otimes I$, $I \otimes A$ and A are isomorphic arenas. Hence, the strategies λ, ρ are simply copy-cat like strategies.

For instance, we give below the interpretation of the introduction rule of I. The sequent $\vdash I$

is interpreted by $I \triangleright I$. The unique non-empty play in the strategy interpreting it consists of the sole possible couples of moves $\mathsf{m}.\mathsf{n}$ as in Figure 6.3. That is $[\![\pi_I]\!] = \{\emptyset, \mathsf{m}.\mathsf{n}\}$ where $\emptyset$ is the empty sequence.

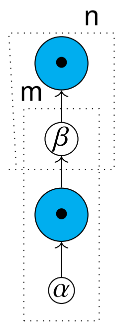


Figure 6.3: Dialogue game of $I \triangleright I$.

The monoidal product allows us to interpret the right tensor rule. The proof:

$$\dfrac{\dfrac{\pi_1}{\Gamma \vdash A} \qquad \dfrac{\pi_2}{\Delta \vdash B}}{\Gamma, \Delta \vdash A \otimes B}$$

is interpreted by $[\![\pi_1]\!] \otimes [\![\pi_2]\!] : [\![\Gamma]\!] \otimes [\![\Delta]\!] \to [\![A]\!] \otimes [\![B]\!]$.

6.1.2 TTSFInn is a dialogue category

The goal is to prove that there is a natural isomorphism between the two following hom-set functors:

$$\mathsf{TTSFInn}(A \otimes B, \bot) \simeq \mathsf{TTSFInn}(A, \neg B).$$

In that case, this translates into:

$$\mathsf{strat}((A \otimes B \otimes (\neg\neg I))^*) \simeq \mathsf{strat}((A \otimes \neg\neg B)^*)$$

where $\mathsf{strat}(C)$ denotes the set of appropriate strategies on the arena C.

We display how those two arenas look like. We start with $(A \otimes B \otimes (\neg\neg I))^*$, whose dialogue game is presented in figure 6.4.

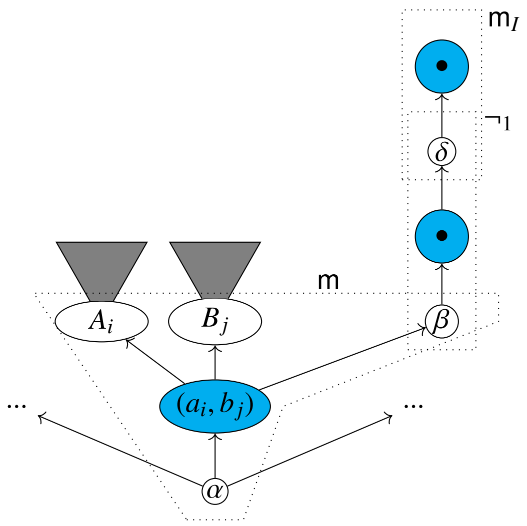


Figure 6.4: Dialogue game for $(A \otimes B \otimes (\neg\neg I))^*$.

The transverse plays of this arena start with a move m, with $\ulcorner \mathsf{m} \urcorner \simeq (\alpha, (a_i, b_j), S_{A_i} \uplus T_{B_j} \uplus \{\beta\})$. Then, as the play is transverse, it has to answer by the unique (up to nominal equivalence) proponent move justified by β, which is a negation move $\ulcorner \mathsf{m}_\neg \urcorner = (\beta, \bullet, \delta)$. Finally, it is the opponent's turn to play, and opponent plays in the unique negative cell available, δ, playing the unique possible move $\ulcorner \mathsf{m}_I \urcorner = (\delta, \bullet)$. At this stage, the β branch of the dialogue tree, that corresponds to $\neg\neg I$ is full, and the proponent must play a move above $S_{A_i} \uplus T_{A_j}$.

We now turn to $(A \otimes \neg\neg B)^*$, whose dialogue game is drawn in Figure 6.5.

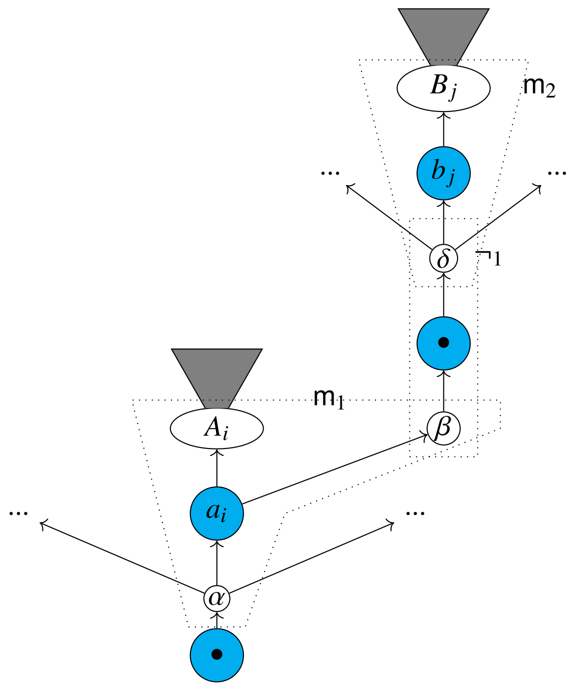


Figure 6.5: Dialogue game for $(A \otimes (\neg\neg B))^*$

A transverse play of $A \triangleright \neg B$ starts with a move m_1, with $\ulcorner \mathsf{m}_1 \urcorner \simeq (\alpha, a_i, S_{A_i} \uplus \beta)$. Then, as the play is transverse, the next move must be the unique (up to equivalence) move justified by β, that is a negation move $\ulcorner \neg \urcorner = (\beta, \bullet, \{\delta\})$. Now, opponent answers with an initial move of B, of the shape $\ulcorner \mathsf{m}_2 \urcorner = (\delta, b_j, T_{B_j})$. Finally, it is proponent's turn to play, and it must play a move above one of the cell of $T_{B_j} \uplus S_{A_i}$.

Therefore, the isomorphisms between strategies must relate plays of the shape $\{(\mathsf{m}_1, \mathsf{m}_2).\neg_1.\mathsf{m}_I.s\}$ and the ones of the shape $\{\mathsf{m}_1.\neg_1.\mathsf{m}_2.s\}$. However, we must pay attention to the legality. Formally, given $\sigma \in \mathsf{TTSFInn}(A \otimes B, \bot)$ we define $\tau \in \mathsf{TTSFInn}(A, \neg B)$ the corresponding strategy by:

$$\tau = \mathsf{Legal}(\{\emptyset, \mathsf{m}_A.\neg_1, \mathsf{m}_A.\neg_1.\mathsf{m}'_B.s \mid (\mathsf{m}_A, \mathsf{m}_b).\neg_1.\mathsf{m}_I.s \in \sigma,\ \mathsf{m}'_B \simeq_{\mathsf{cells}} \mathsf{m}_B\})$$

And, we present the reverse direction:

$$\sigma = \mathsf{Legal}(\{\emptyset, (\mathsf{m}_A \times_{\mathbb{A}_{\mathsf{cells}}} \mathsf{m}_B).\neg_1, (\mathsf{m}_A \times_{\mathbb{A}_{\mathsf{cells}}} \mathsf{m}_B).\neg_1.\mathsf{m}_I.s \mid (\mathsf{m}_A.\neg_1.\mathsf{m}'_B.s) \in \tau,\ \mathsf{m}_B) \simeq_{\mathsf{cells}} \mathsf{m}_B\})$$

Furthermore, let us give a brief description of the two sets of positions of the different arenas:

$$\begin{aligned} \mathsf{Trans}((A \otimes B) \triangleright \bot) &\simeq \mathsf{Pos}(A \otimes B) \mathbin{\overline{\otimes}} \mathsf{Pos}(\bot) \\ \mathsf{Trans}(A \triangleright \neg B) &\simeq \mathsf{Pos}(A) \mathbin{\overline{\otimes}} \mathsf{Pos}(\neg B) \\ \mathsf{Trans}^*(A \triangleright \neg B) &\simeq \mathsf{Pos}^*(A) \times_{\mathbb{A}_{\mathsf{cells}}} (\mathsf{Event}(\top) \times^{\mathbb{A}_{\mathsf{cells}}} \mathsf{Pos}(B)) \end{aligned}$$

We recall that we write $\times^{\mathbb{A}_{\mathsf{cells}}}$ for the following variant of the fibred product between moves: $(\neg \times^{\mathbb{A}_{\mathsf{cells}}} \mathsf{m})$ denotes the product of moves $(\neg, \mathsf{m})$ imposing that the final cell of $\neg$ is the initial cell of m, where $\neg$ denotes an event of $\mathsf{Event}(\top)$, that is, a move of the form $(\alpha, v, \{\beta\})$ where $\alpha, \beta \in \mathbb{A}_{\mathsf{cells}}$. This was introduced in Section 4.5.4. For every position of x of $\mathsf{Trans}(A \otimes B \triangleright \bot)$ reached by a play of length strictly more than 2 belonging to a strategy σ, then writing $x = (x_A, x_B, x_\bot)$, every cell available at x belongs in (x_A, x_B). Furthermore, a play of length more than 4 of a strategy $\tau : A \triangleright \neg B$ will reach a position $(x_A, \neg \times^{\mathbb{A}_{\mathsf{cells}}} x_B)$ (where we write $\neg$ for the events of $\mathsf{Event}(\top)$), where x_B is not the empty position. We write $f : A_{(x_A, \neg \times^{\mathbb{A}_{\mathsf{cells}}} x_B)} \to A_{(x_A, x_B)}$ for the canonical bijection between the available cells of $(x_A, \neg \times^{\mathbb{A}_{\mathsf{cells}}} x_B)$ and those of (x_A, x_B). Now, giving a position $(x_A, \neg \times^{\mathbb{A}_{\mathsf{cells}}} x_B)$ a position of τ, and $(x_A, x_B, x_\bot)$ a position of σ, we define ψ the sequentiality structure of ψ, in function of ϕ, the sequentiality structure of σ, as follows:

$$\psi_{(x_A, \neg \times^{\mathbb{A}_{\mathsf{cells}}} x_B)}(\alpha) = f^{-1}(\phi_{(x_A, x_B, x_\bot)}(f(\alpha)))$$

And straightforwardly, we could express the reverse direction:

$$\phi_{(x_A, x_B, x_\bot)}(\alpha) = f(\psi_{(x_A, \neg \times^{\mathbb{A}_{\mathsf{cells}}} x_B)}(f^{-1}(\alpha)))$$

As the sequentiality structures ϕ, ψ are equivariant, the choice of $x_\perp$ in one case, or the move $\neg$ in the other does not change their behaviour, and therefore these functions are well-defined.

Overall, we obtain a bijection between the set of brave strategies of $A \otimes B \triangleright \perp$ and $A \triangleright \neg B$. Note that in similar way we could have obtained a bijection between the strategies of $\mathsf{TTSFInn}(A \otimes \neg B, \perp)$ and the morphisms of $\mathsf{TTSFInn}(A, B)$.

This allows us to define the denotation of the right negation. Given a proof π as below:

$$\dfrac{\dfrac{\pi'}{\Gamma, A \vdash \perp}}{\Gamma \vdash \neg A}$$

then the interpretation of π is also $g([\![\pi']\!])$, with g being the isomorphism $\mathsf{TTSFInn}(\bigotimes \Gamma \otimes A, \perp) \xrightarrow{g} \mathsf{TTSFInn}(\bigotimes \Gamma, \neg A)$.

The left negation case is dealt with on an equal basis noticing that there is an injection:

$$\mathsf{strat}(A \otimes \neg B)^* \simeq \mathsf{strat}(A \otimes \neg B \otimes \neg\neg I)^*$$

This injection is not a bijection since the second move of a transverse strategy of $A \triangleright B$ must be in B, whereas a strategy of $(A \otimes \neg B) \triangleright \perp$ can, after playing one move in $A \otimes \neg B$, and two in $\neg\neg I$, start exploring A and ignore $\neg B$. Given a proof π as follows:

$$\dfrac{\dfrac{\pi'}{\Gamma \vdash A}}{\Gamma, \neg A \vdash \perp}$$

then, calling g the injection $\mathsf{TTSFInn}(\bigotimes \Gamma, A) \xrightarrow{g} \mathsf{TTSFInn}(\bigotimes \Gamma \otimes \neg A, \perp)$, the interpretation of π is $g(\pi')$.

6.1.3 TTSFInn has finite coproducts

Finally, let us study the sum structure of $\mathsf{TTSFInn}$. As expected, given A, B objects of $\mathsf{TTSFInn}$, the object corresponding to their coproduct is $A \oplus B$. We need to prove that $A \oplus B$ indeed satisfies the universal property of the coproduct. Namely, that there exist two morphisms $\mathsf{inl} : A \to A \oplus B$ and $\mathsf{inr} : B \to A \oplus B$, such that for every object C, for every pair of morphisms $\sigma_A : A \to C$ and $\sigma_B : B \to C$, there exists a unique morphism $\tau : A \oplus B \to C$ such that $\mathsf{inl}; \tau = \sigma_A$ and $\mathsf{inr}; \tau = \sigma_B$. We display in figure 6.6 below the associated diagram.

So let us consider two strategies $\sigma_A : A \triangleright C$ and $\sigma_B : B \triangleright C$, and let us look at the structure of $A \oplus B \triangleright C$, displayed in Figure 6.7. It is clear that either the first move is going to be played in A, and then the strategy will follow σ, or it will be played in B, and then the strategy will play as τ. More precisely, we study formally the correspondence, establishing a isomorphism between

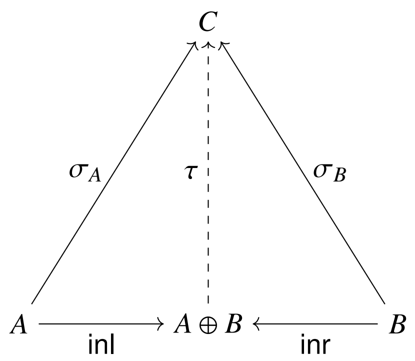


Figure 6.6: Coproduct diagram of $A \oplus B$

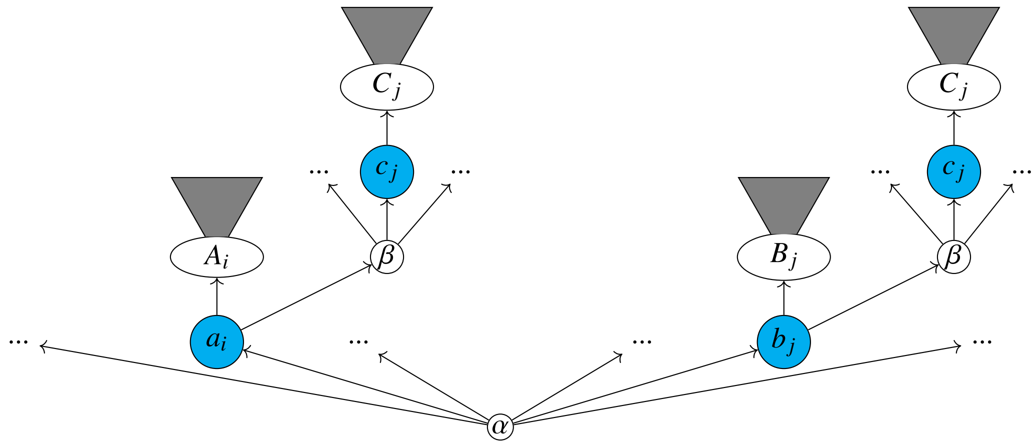


Figure 6.7: Dialogue game of $A \oplus B \triangleright C$

.

the set of transverse positions of $(A \oplus B) \triangleright C$, and the appropriate sum of the sets of transverse positions of $A \triangleright B$ and $A \triangleright C$.

$$\begin{aligned}
\mathsf{Trans}((A \oplus B) \triangleright C) &\simeq \mathsf{Pos}(A \oplus B) \,\overline{\otimes}\, \mathsf{Pos}(C) \\
&\simeq (\mathsf{inl}(\mathsf{Pos}(A)) \,\overline{\uplus}\, \mathsf{inr}(\mathsf{Pos}(B))) \,\overline{\otimes}\, \mathsf{Pos}(C) \\
&\simeq (\mathsf{inl}(\mathsf{Pos}(A)) \,\overline{\otimes}\, \mathsf{Pos}(C)) \,\overline{\uplus}\, (\mathsf{inr}(\mathsf{Pos}(B) \,\overline{\otimes}\, \mathsf{Pos}(C)) \\
&\simeq \mathsf{inl}(\mathsf{Pos}(A) \,\overline{\otimes}\, \mathsf{Pos}(C)) \,\overline{\uplus}\, \mathsf{inr}(\mathsf{Pos}(B) \,\overline{\otimes}\, \mathsf{Pos}(C)) \\
&\simeq \mathsf{inl}(\mathsf{Trans}(A \triangleright C)) \,\overline{\uplus}\, \mathsf{inr}(\mathsf{Trans}(B \triangleright C))
\end{aligned}$$

Therefore, using the above isomorphism, the strategy $\sigma_A \oplus \sigma_B$ is defined by:

$$(\sigma_A \oplus \sigma_B)^\bullet \simeq \mathsf{inl}(\sigma_A^\bullet) \,\overline{\uplus}\, \mathsf{inr}(\sigma_B^\bullet)$$

The strategy thus defined is innocent, total, frugal, and transverse, since the two original strategies are. Finally, defining $\mathsf{inl} : A \triangleright A \oplus B$ to be the identity strategy between A and the A part of $A \oplus B$, and similarly for inr, one can straightforwardly see that $\tau = (\sigma_A \oplus \sigma_B)$ makes the diagram of figure 6.6 commutes.

Furthermore, given a strategy $\sigma : A \oplus B \triangleright C$, then $\sigma^\bullet \simeq (\sigma^\bullet \upharpoonright \mathsf{inl}(A \triangleright C)) \overline{\uplus} (\sigma^\bullet \upharpoonright \mathsf{inr}(B \triangleright C))$, and hence there is a decomposition of σ into two strategies. Furthermore, these two strategies satisfy straightforwardly all the necessary conditions.

Finally, we study the unit of the addition. As $\mathsf{Pos}(A \oplus 0) \simeq \mathsf{Pos}(A)$, since $\mathsf{Pos}(0)$ has only one position, namely the empty one, one can define two isomorphisms $A \triangleright A \oplus 0$ and $A \oplus 0 \triangleright A$, that essentially act as the copy-cat strategy.

For instance, let us give the denotation of the proof π:

$$\frac{}{\Gamma, 0 \vdash C} \text{Left } 0$$

Let us write $\Gamma = F_1, ...F_n$. This proof will be interpreted as a strategy in the pre-arena $(F_1 \otimes ... \otimes F_n \otimes 0 \otimes \neg C)^* \simeq 0^*$. Hence this proof will be interpreted by the strategy that has as unique play the empty sequence.

The sum allows to define the denotation of the two right $\oplus$-rules, and the left $\oplus$-rule. For instance, let us consider the proof π below:

$$\frac{\dfrac{\pi'}{\Gamma \vdash A}}{\Gamma \vdash A \oplus B} \text{right-}\oplus_1$$

Then, $[\![\pi]\!] = [\![\pi']\!]; \mathsf{inl}$, where $\mathsf{inl} : A \to A \oplus B$. Similarly, the right $\oplus_2$ rule is interpreted by post-composition with inr. Likewise, the left $\oplus$-rule, is interpreted by the sum. For instance, the proof π below:

$$\frac{\dfrac{\pi_1}{\Gamma, A_1 \vdash B} \qquad \dfrac{\pi_2}{\Gamma, A_2 \vdash B}}{\Gamma, A_1 \oplus A_2 \vdash B}$$

is interpreted as follows. Let us write β the associativity morphism:
$\beta : [\![\Gamma]\!] \otimes ([\![A_1]\!] \oplus [\![A_2]\!]) \to ([\![\Gamma]\!] \otimes [\![A_1]\!]) \oplus ([\![\Gamma]\!] \otimes [\![A_2]\!])$. Then $[\![\pi]\!] = \beta; ([\![\pi_1]\!] \oplus [\![\pi_2]\!])$,

We conclude this section by recapitulating that all the points we proved along this section together allow us to conclude that $\mathsf{TTSFInn}$ organises itself as a category that can soundly interpret axiom-links and forms a dialogue category with sums. Therefore, it is a sound model of tensorial logic with propositional variables.

6.2 Full completeness for propositional tensorial logic

The goal of this section is to prove that the category $\mathsf{TTSFInn}$ is the free dialogue category with products on VAR. This is the content of the proposition below.

Proposition 6.1. *There is a correspondence between the equivalence classes of proofs of* $\Gamma \vdash A$ *and the morphisms of* $\mathsf{TTSFInn}([\![\Gamma]\!], [\![A]\!])$.

This entails strong completeness, and more. That is, the functor from the proof invariants of tensorial logic to $\mathsf{TTSFInn}$ is not only full, but also faithful. The proof of the proposition relies on the following lemma.

Lemma 6.2. *Let* $\neg B_1, \neg B_2, ..., \neg B_n \vdash \neg A$ *a sequent. Then there is a one to one correspondence between :*

- *The equivalence classes of proofs of the sequent focussing on one* B_i *and the transverse strategies of* $([\![\neg B_1]\!] \otimes [\![\neg B_2]\!] .... \otimes [\![\neg B_n]\!] \otimes [\![A]\!])^*$.
- *The equivalence classes of proofs of the sequent and the strategies of* $([\![B_1]\!] \otimes [\![\neg B_2]\!] ... \otimes [\![\neg B_n]\!] \otimes [\![A]\!])^*$.

What we meant by focussing on one B_i is that, taking a focalised proof in the equivalence class, it will behave by focussing on one of the B_i at the beginning of its first synchronous phase, as displayed in the proof below.

$$\dfrac{\dfrac{\dfrac{\dfrac{\pi}{\neg B_1, ..., \neg B_{i-1}, \neg B_{i+1}, ..., \neg B_n, A'_1, .., A'_m \vdash B_i}}{\neg B_1, \neg B_2, ..., \neg B_n, A'_1, .., A'_m \vdash \bot}}{\neg B_1, \neg B_2, ..., \neg B_n, A \vdash \bot}}{\neg B_1, \neg B_2, ..., \neg B_n \vdash \neg A}$$

Now, there is a syntactic equivalence between the focalised proofs of $B_1, ..., B_n \vdash A$ and the focalised proofs of $\neg A \vdash \neg(B_1 \otimes ... \otimes B_n)$, that will be focussing on A, as displayed below:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\pi'}{B'_1, ..., B'_m \vdash A}}{\neg A, B'_1, ..., B'_m \vdash \bot}\ \text{Asynchronous phase}}{\neg A, B_1, ..., B_n \vdash \bot}\ \otimes}{\neg A, B_1 \otimes .... \otimes B_n \vdash \bot}}{\neg A \vdash \neg(B_1 \otimes ... \otimes B_n)} \qquad \sim \qquad \dfrac{\dfrac{\pi'}{B'_1, ..., B'_m \vdash A}}{B_1, .., B_n \vdash A}\ \text{Asynchronous phase}$$

Therefore, following the lemma, there is a one-to-one correspondence between the equivalence classes of proofs of $\neg A \vdash \neg(B_1 \otimes ... \otimes B_n)$ and the transverse strategies of $([\![B_1 \otimes ... \otimes B_n]\!] \otimes [\![\neg A]\!])^*$ that is, the transverse strategies of $[\![\Gamma]\!] \triangleright [\![A]\!]$, where $\Gamma = B_1, ..., B_n$. Hence proving the lemma 6.2 entails proving the proposition 6.1.

We work with proofs in $\mathsf{TENS}_{\mathsf{foc,glob}}$. We remind that within this fragment, two proofs are equivalent if and only if they are equal.

Proof of lemma 6.2. We tackle the two points of the lemma at once. The proof is done by induction on the maximal length of the sequences of σ. Let us note that since σ is equivariant and frugal, the names chosen on the moves of the sequent we pick to deal with the induction case will not matter. So let us start with the case where σ simply has the empty sequence. Then, as the strategy is total, this implies that the whole pre-arena has no moves. Hence the pre-arena's dialogue game consists of a set of untyped cells, and hence is 0. So we need to solve the following arena equation:

$$(\neg B_1 \otimes ... \otimes \neg B_n \otimes A) = 0.$$

that has solutions (n+1)-tuples $(B_1, .., B_n, A)$ of the form $(B_1, ..., B_n, 0)$ for any $B_1, .., B_n$. This corresponds to a sequent :

$$\neg B_1, ..., \neg B_m \vdash \neg 0$$

which has the unique following proof:

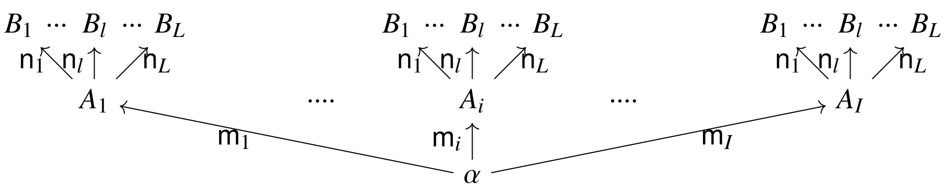
$$\text{Right } 0 \frac{\overline{\neg B_1, ..., \neg B_n, 0 \vdash \bot}}{\neg B_1, ..., \neg B_n \vdash \neg 0}$$

This settles the base case.

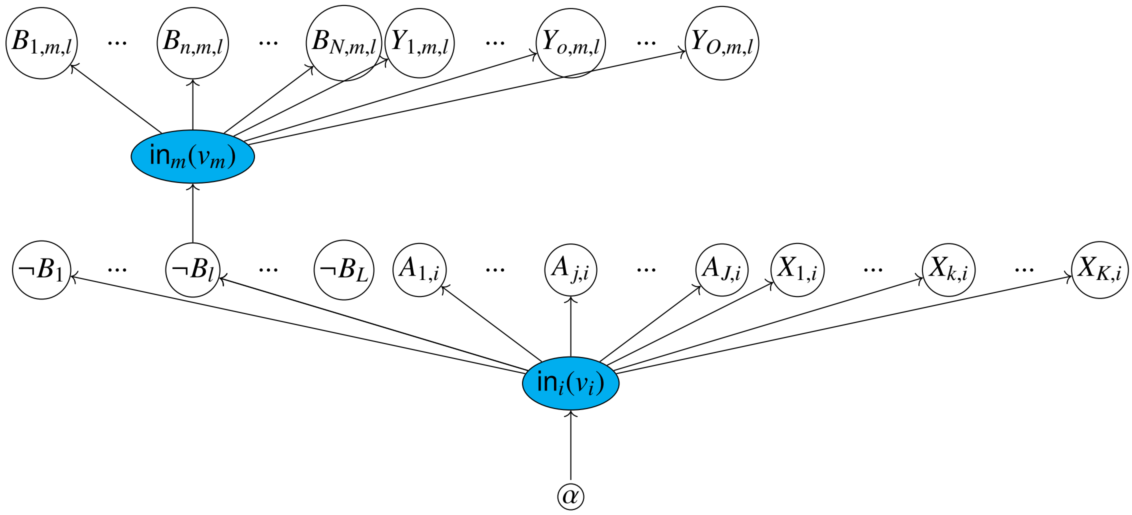
We now move to the inductive case. So suppose $s \in \sigma$ is a sequence of maximal length, $s = \mathsf{m}.\mathsf{n}.s'$. We adopt the following notation convention: given two indices i, j, we write $i \mid j$ to say that the set from which i ranges depends of the index j, and we will explicit the ranging sets only when necessary. For instance, we would write $\bigoplus_i \bigotimes_{i|j}$ for $\bigoplus_{i\in I} \bigotimes_{j\in I_j}$. Furthermore, as indices range through downward closed sets of positive natural numbers, we write with an uppercase the upper bound to which they range. That is, j ranges from 1 to J.

Each B_l is either isomorphic to 0, or a sum $B_l = \bigoplus_{m|l}(\bigotimes_{n|m,l} \neg B_{n,m,l} \bigotimes_{o|m,l} Y_{o,m,l} \bigotimes_{u|m,l} I)$ where $Y_{o,m,l}$ are atomic types, and $A = \bigoplus_i(\bigotimes_{j|i} \neg A_{j,i} \bigotimes_{k|i} X_{k,i} \bigotimes_{v|i} I)$, or is isomorphic to 0. The totality of the strategy prevents the case where there is a unique B, and this one is 0, as player would not be able to answer to the opponent move. This would correspond to the case $\neg 0 \vdash \neg A$, and, in essence, to $A \vdash 0$. That is, totality of the strategy is the counterpart of the absence of right induction rule for 0. In the following, to make it more readable, we denote each cell by the sub-formula it encompasses.

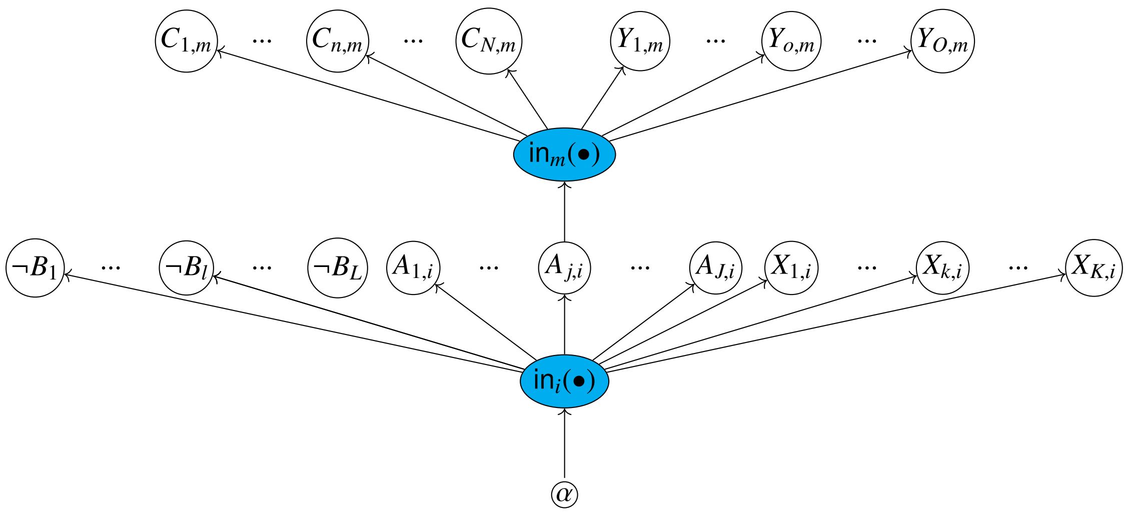
The structure of the two first moves of s will be as follows, in the transverse case:



where each $\mathsf{m}_i.\mathsf{n}_l$ move has the following structure :



However, in the non transverse case, the move n_l can also be above one of the $A_{j,i}$. Then, writing $A_{j,i} = \bigoplus_n (\bigotimes_m \neg C_{n,m} \bigotimes_o Y_{o,n})$, we can get two moves as displayed in the drawing below.



As the two cases are similar, we focus on the transverse one.

The opponent is going to play a value with cells $\{\neg B_l \mid l \in L\} \cup \{A_{j,i} \mid j \in J_i\} \cup \{\alpha \mid \alpha \in X_{k,i}, k \in K_i\}$ for a given i that he would have chosen, and writing $\alpha \in X_{k,i}$ for $\alpha \in \mathbb{A}_{X_{k,i}}$. As the strategy is transverse, the player will answer in one of the cells B_l, and will play a value with cells $\{B_{n,m,l} \mid n \in N_{m,l}\} \cup \{\alpha \mid \alpha \in Y_{o,m,l}, o \in O_{m,l}\}$ for a given m, l that he will pick. He will furthermore play a sequentiality function from the opponent cells just introduced to his own cells. This corresponds to a set of global focalised rules in the proof, sets out in Figure 6.2, where the four first negative rules starting from the root (Right $\neg$, Left $\bigoplus$, Left $\bigotimes$, Left I), are bound to the opponent move m, and the four next together with the axiom-links to the proponent move n (Left $\neg$, Right $\bigoplus$, Right $\bigotimes$, Right I, axioms).

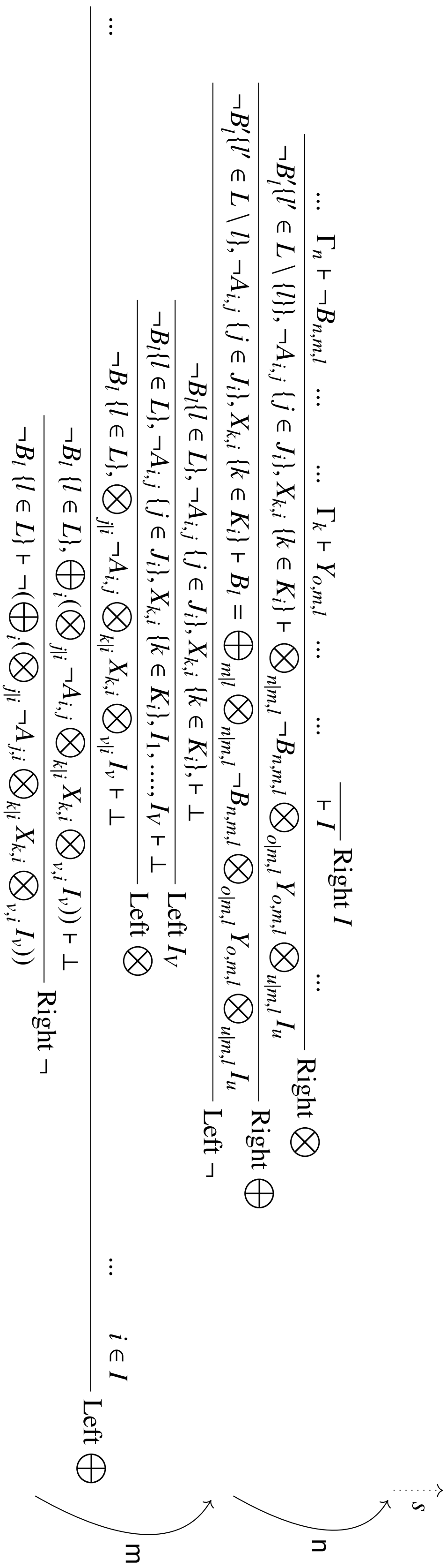
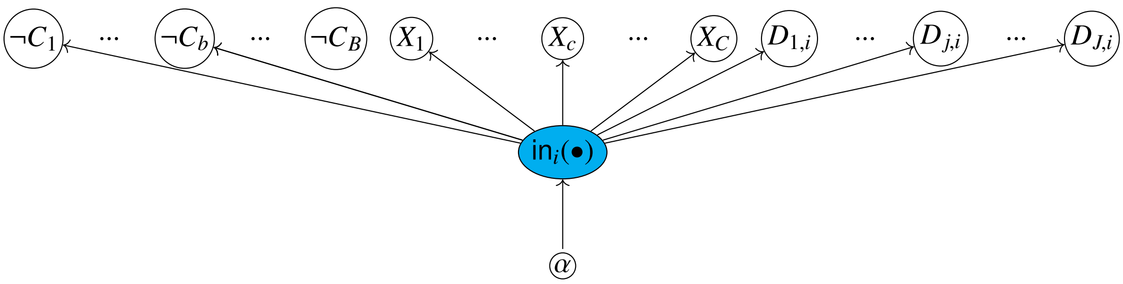


Figure 6.8: Definability: induction case for m.n.s

Each Γ_n corresponds to the dominion of the negative cell that represents $B_{n,m,j}$, or $Y_{o,m,l}$. Now the only cell that can match a cell coming from $Y_{o,m,l}$ on the right hand side is one of the same type, so the Γ_k are of the form $X_{k,j}$, with $X_{k,j} = Y_{o,m,l}$, and each $X_{k,j} \vdash Y_{o,m,l}$ corresponds to the application of an axiom rule. Note that if one of the B_l is 1, then it cannot capture any context. This would then be an application of the right unit rule.

By the lemma of separation of contexts 5.18, one can now focus on each of the branch individually. Furthermore, one can see that every sequence above the cells $\alpha \cup \mathsf{dominion}(\alpha)$ corresponding to $\Gamma_n = \neg C_1, ..., \neg C_b, X_1, ..., X_c \vdash \neg B_{n,m,l}$ in the strategy can be faithfully translated as a sequence in $\neg C_1 \otimes ... \otimes \neg C_b \otimes X_1 \otimes ... \otimes X_c \otimes B_{n,m,l}$, and hence can be seen as an interpretation of a proof $\neg C_1 \otimes ... \otimes \neg C_b \otimes X_1 \otimes ... \otimes X_c \vdash \neg B_{n,m,l}$. Indeed, let us look at the structure of the first move of the arena $\neg C_1 \otimes ... \otimes \neg C_b \otimes X_1 \otimes ... \otimes X_c \otimes B_{n,m,l}$. To simplify things, we assume $B_{n,m,l} = \bigoplus_i (\bigotimes_j D_{i,j})$.



Then the sequences of σ_β can translated as sequences in $\neg C_1 \otimes ... \otimes \neg C_b \otimes X_1 \otimes ... \otimes X_c \otimes B_{n,m,l}$, with the first move of opponent in $\neg C_1 \otimes ... \otimes \neg C_b \otimes X_1 \otimes ... \otimes X_c \otimes B_{n,m,l}$ filling the cells of $C_1, .., C_b, X_1, .., X_c$ with the cells of $\mathsf{dominion}(\beta)$.

Therefore, we can apply the induction hypothesis, and conclude that the sequences of σ define a unique global focalised proof, that is, a proof of $\mathsf{TENS}_{\mathsf{foc-glob}}$. □

Overall, we have obtained a perfect abstract representation and characterisation of the proofs of tensorial logic through nominal strategies in sequential, asynchronous games.

6.3 The case for MALL

6.3.1 Interpretation

We remind here that every proof of tensorial logic can be translated into a proof of linear logic, and reversely. Namely, we remind the proposition 2.10 below. In this property, $\mathcal{P}$ denotes a set of positive formulas of linear logic, $\mathcal{N}$ a set of negative ones, and $\mathcal{X}$ a set of negative atomic formulas.

Proposition 6.3. *Every proof of* $\vdash \mathcal{P}, \mathcal{N}, \mathcal{X};$ *(respectively* $\vdash \mathcal{P}, \mathcal{N}, \mathcal{X}; P$*) in weakly focussed linear logic induces a proof of the sequent* $\neg(\mathcal{P})^F, (\mathcal{N}^\perp)^F, \mathcal{X}^\perp \vdash \perp$ *(resp* $\neg(\mathcal{P})^F, (\mathcal{N}^\perp)^F, \mathcal{X}^\perp \vdash (P)^F$*) in tensorial logic, and reciprocally, where* $\mathcal{P}$ *is the subset of positive formulas of* Γ*,* $\mathcal{X}$ *its subset of negative atomic formulas,* $\mathcal{N}$ *its subset of negative formulas that are not in* $\mathcal{X}$*. Furthermore* $(\Pi)^F = \perp$ *if* (Π) *is empty,* $(P)^F$ *in the case where* $(\Pi) = P$ *is positive, and* $\neg(M^\perp)^F$ *in the case where* $\Pi = M$ *is negative.*

To simplify things, given a weakly focalised linear logic sequent $\vdash \Gamma; \Pi$, we write $(\Gamma)^F \vdash (\Pi)^F$ for the appropriate translation into tensorial logic, where Π is either a single positive formula or the empty sequent. Then let π be a proof of linear logic. We can translate it into a proof $(\pi)^F$ of tensorial logic. This proof $(\pi)^F$ can be given a interpretation $[\![\pi]\!]_{\mathsf{TTSFInn}} \in \mathsf{TTSFInn}$. However, we remind that this interpretation is not a categorical functor, as we night have two proofs π, π' of linear logic such that $\pi \sim \pi'$, but $[\![(\pi)^F]\!]_{\mathsf{TTSFInn}} \neq [\![(\pi')^F]\!]_{\mathsf{TTSFInn}}$.

Therefore, the right translation from linear to tensorial logic should be along the following lines:

$$[\![\pi]\!] = \{[\![(\pi')^F]\!]_{\mathsf{TTSFInn}} \mid \pi' \sim \pi\}$$

Therefore, one needs to define an equivalence relation of strategies of tensorial logic, relating strategies that denote the same proof of linear logic.

Definition 6.4. *Two strategies* $\sigma, \sigma' : A$ *are equivalent, written* $\sigma \sim_1 \sigma'$ *if there exists* $\pi, \pi' : (A)^I$ *such that* $[\![(\pi)^F]\!]_{\mathsf{TTSFInn}} = \sigma$*,* $[\![(\pi')^F]\!]_{\mathsf{TTSFInn}} = \sigma'$ *and* $\pi \sim \pi'$*.*

At this point, one should look for an invariant, that is, a function f together with a set S, such that the image of f lies in S, and $\pi \sim_1 \pi' \Rightarrow f([\![(\pi)^F]\!]_{\mathsf{TTSFInn}}) = f([\![(\pi)'^F]\!]_{\mathsf{TTSFInn}})$. More precisely, we look for a categorical invariant, that is, a functor F from the category $\mathsf{TTSFInn}$ to a star-autonomous category, such that $(\sigma \sim_1 \sigma') \Rightarrow F(\sigma) = F(\sigma')$. Such a functor yields the ground of a fully complete denotational semantics of proofs of linear logic.

6.3.2 About the quotient

The difference between tensorial and linear logic lies in the non-involutive negation. As each negation is interpreted by a move in the category of games, one would like to project moves onto a flat domain. The basic idea is to project onto maximal positions. Indeed, one can notice that, given a strategy $\sigma : F$, and the same strategy double-lifted with two negation moves $\neg\neg\sigma : \neg\neg F$, then the two will reach the same maximal positions. This is coherent with the fact that the negation is involutive in linear logic, thus $(\neg\neg F)^I = F^{\perp,\perp} = F$, and the strategy $\neg\neg\sigma$ and σ should correspond to the same proof of linear logic. That is $(\neg\neg\sigma)^I = (\sigma)^I$.

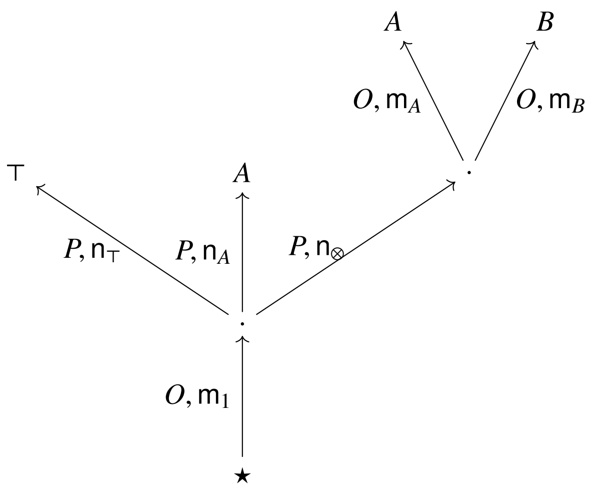
However, we present below an example highlighting why we sometimes should identify strategies that do not reach the same set of maximal positions. For instance, let us consider the following two proofs of linear logic :

$$\frac{\dfrac{}{\vdash, \top, A, A^{\perp} \otimes B^{\perp}} \qquad \dfrac{}{\vdash 1}}{\vdash \top \otimes 1, A, A^{\perp} \otimes B^{\perp}}$$

and

$$\frac{\dfrac{}{\vdash A^{\perp}, A} \qquad \dfrac{\dfrac{}{\vdash \top, B^{\perp}} \qquad \dfrac{}{\vdash 1}}{\vdash \top \otimes 1, B^{\perp}}}{\vdash \top \otimes 1, A, A^{\perp} \otimes B^{\perp}}$$

Let us display below the underlying structure of the dialogue-game associated with the conclusion formula.



The strategy corresponding to the second proof has two maximal plays associated with it, namely $\mathsf{m}_1.\mathsf{n}_{\otimes}, \mathsf{m}_A.\mathsf{n}_A.\mathsf{m}_B.\mathsf{n}_{\top}$ and $\mathsf{m}_1.\mathsf{n}_{\otimes}.\mathsf{m}_B.\mathsf{n}_{\top}.\mathsf{m}_A.\mathsf{n}_A$. These two reach the same maximal position. On the other hand, the strategy associated with the first proof has a unique maximal play $\mathsf{m}_1.\mathsf{n}_{\top}$. As a result, we have two different strategies reaching two different sets of maximal positions. However, they correspond to two proofs that are equivalent.

Indeed, if we suppose that they are not equivalent, this would imply that there are (at least) two morphisms $1 \to (\top \otimes 1) \parr A \parr A^{\perp} \otimes B^{\perp}$. But as $\top \otimes 1 \simeq \top$, and, for any formula F, $\top \parr F \simeq \top$, this would imply that there are (at least) two distinct morphisms $1 \to \top$. Or, as $\top$ is terminal, there is only one.

The reason behind it is to be looked for in the time where the proof decides to use the $\top$ rule. In tensorial logic, as the model is dynamic, the moment when we decide to use it makes a difference. On the other hand, in linear logic, the model being flat, these two proofs will be confounded. Therefore, we have to focus only on those maximal positions that are significant.

Definition 6.5 ([69, 66])**.** *An **external position** is a position such that no untyped cells are available.*

Therefore, no move can happen from an external position, and it is maximal. Given a strategy σ of $\mathsf{TTSFInn}$, we write $\sigma^{\mathsf{external}}$ for its set of external positions.

Definition 6.6. *We define the equivalence relation σ_2 between strategies by relating strategies having the same set of external positions:*

$$\sigma \sim_2 \tau \;\Leftrightarrow \sigma^{\mathsf{external}} = \tau^{\mathsf{external}}$$

The proj function is designed with external positions in mind. Indeed, proj is undefined on $\top, 0$, and, by extension, on maximal non-external positions. This has to be put in relation with relations, where the unit is also interpreted by $\emptyset$, and hence any multiplicative formula with a additive unit in it has denotation the empty-set. Likewise, for every set of positions x , $\mathsf{proj}(\top).\mathsf{proj}(x) = \emptyset$ (where here $\top$ denotes the set of maximal positions of the dialogue game interpreting $\top$). Hence, the above equation translates formally as $\mathsf{proj}(\sigma) = \mathsf{proj}(\tau)$. That is:

$$\sigma \sim_2 \tau \Leftrightarrow \mathsf{proj}(\sigma) = \mathsf{proj}(\tau)$$

The purpose of the next section 6.3.3 is to make sure that this invariant is a sound one. That is, $\sigma \sim_1 \tau \Rightarrow \sigma \sim_2 \tau$.

6.3.3 Quotient and star autonomy

Proposition 6.7. *Given a formula A of linear logic, the function $\mathsf{proj}_A : [\![(A)^F]\!]_{\mathsf{TTSFInn}} \to [\![A]\!]_{\mathsf{NomLinRel}}$, defined in section 4.5.6 is an invariant of the interpretations of proofs. That is, the following diagram commutes.*

$$\begin{array}{ccc}
\pi : A & \xrightarrow{\quad (.)^F \quad} & (\pi)^F : (A)^F \\
\Big\downarrow {\scriptstyle [\![\cdot]\!]_{\mathsf{NomLinRelPol}}} & & \Big\downarrow {\scriptstyle [\![\cdot]\!]_{\mathsf{TTSFInn}}} \\
[\![\pi]\!]_{\mathsf{NomLinRelPol}} & \xleftarrow[\quad \mathsf{proj}_A \quad]{} & [\![(\pi)^F]\!]_{\mathsf{TTSFInn}}
\end{array}$$

The demonstration is a proof by induction on the last rule of π. This could also have been proven by categorical means, relying on proj forming a functor of dialogue categories with sums. Indeed the translation $(.)^F$ has been conceived such that for every functor of dialogue categories

$F : \mathsf{Dial} \to \mathsf{Star}$, where Star is a star-autonomous category seen as a dialogue category, for every denotation function $[\![.]\!] : \mathsf{Tens} \to \mathsf{Dial}$, then $F \circ [\![(.)^F]\!] : \mathsf{MALL} \to \mathsf{Star}$ is a denotation function for linear logic proofs, that is, a star-autonomous functor. The proof is along the same lines as the proof of the proposition, and this property entails $\sigma \sim_1 \tau \Rightarrow \sigma \sim_2 \tau$.

Proof. We start the proof by treating the leaves cases. We first tackle the axiom, then the I-rule. For convenience, we deal with the $\bot$-rule straight after. At last, we present the $\top$-rule.

The axiom proof of $\pi :\vdash X^\bot; X$ is sent to the relation $\{(a,-1).(a,1) \mid a \in \mathbb{A}_X\}$. On the other hand $(\pi)^F$ is the axiom proof of tensorial logic $(\pi)^F : X \vdash X$. It is interpreted as in section 6.1, and one can clearly see that $\mathsf{proj}([\![(\pi)^F]\!]) = \{(\lambda,-1).(\lambda,1) \mid \lambda \in \mathbb{A}_X\}$, as expected.

If the proof only consists of a I rule, introducing $\vdash I$, then it is translated as the relation $(\bullet,-1).(\bullet,1) : [\![I]\!] \to [\![I]\!]$. This proof is translated into tensorial logic as the proof $\vdash I$ as well, that is interpreted as the game of figure 6.3, reaching the unique maximal position x of $I \otimes \neg I$. Hence, this position is sent by proj onto $(\bullet,-1).(\bullet,1)$.

The other multiplicative unit is $\bot$. Let us consider a proof π whose last rule is a $\bot$-rule introduction:

$$\dfrac{\dfrac{\pi'}{\vdash \Gamma, ; A}}{\vdash \Gamma, \bot; A}$$

Then the interpretation of π is $\{x_\Gamma.(\bullet,-1).x_A \mid x_\Gamma.x_A \in [\![\pi']\!]_{\mathsf{NomLinRelPol}}\}$. Now let us consider the translation of π into tensorial logic.

$$\dfrac{\dfrac{(\pi')^F}{(\Gamma)^F \vdash (\Pi)^F}}{(\Gamma)^F, I \vdash (\Pi)^F}$$

The first move of the strategy σ interpreting π is the same as the strategy σ' interpreting π', but the projection now differs, and takes the left I into account. That is, we now have $\mathsf{proj}([\![(\pi)^F]\!]^\bullet_{\mathsf{TTSFInn}}) = \{(x_\Gamma.(\bullet,-1).x_A \mid x_\Gamma.x_A \in [\![(\pi')^F]\!]_{\mathsf{NomLinRelPol}}\}$ as expected.

If the last rule of the proof of π is a Foc rule:

$$\dfrac{\dfrac{\pi'}{\vdash \Gamma; P}}{\vdash \Gamma, P;}\ \text{Foc}$$

The proof π is translated into the same relation as π', plus a atom on the right hand side that would correspond to the unit of the $⅋$, that is $\bot$; $[\![\pi]\!]_{\mathsf{NomLinRelPol}} = [\![\pi']\!]_{\mathsf{NomLinRelPol}}.(\bullet,-1)$. Then it is translated into:

$$\dfrac{\dfrac{(\pi')^F}{\Gamma^F \vdash (P)^F}}{\Gamma^F, \neg(P)^F \vdash \bot}\text{ Left }\neg$$

Hence, written $\sigma = [\![(\pi)^F]\!]_{\mathsf{TTSFInn}}$ and $\sigma' = [\![(\pi')^F]\!]_{\mathsf{TTSFInn}}$, $\mathsf{proj}(\sigma^\bullet) = \mathsf{proj}(\sigma'^\bullet).(\bullet, -1)$, as we could see from the interpretation of the negation in terms of strategies, given in Section 6.1.2.

The case for unfoc is similar to the foc case.

$$\dfrac{\dfrac{\pi'}{\Gamma, M;}}{\Gamma; M}\text{ unfoc}$$

The the proof of π is detonated as follows:

$$[\![\pi]\!]_{\mathsf{NomLinRelPol}} = \{(x_\Gamma.x_M) \mid (x_\Gamma.x_M.(\bullet, -1)) \in [\![\pi']\!]_{\mathsf{NomLinRelPol}}\}$$

It is translated by $(.)^F$ into:

$$\dfrac{\dfrac{(\pi')^F}{\Gamma^F, (M^\perp)^F \vdash \bot}}{\Gamma^F \vdash \neg(M^\perp)^F}\text{ Right }\neg$$

whose interpretation is again given in section 6.1.2. Then again, one can notice that proj acts as follows:

$$\mathsf{proj}(\sigma^F) = \{(x_{\mathcal{P}}.x_{\mathcal{N}}.x_N) \mid (x_{\mathcal{P}}.x_{\mathcal{N}}x_M.x_{\mathcal{X}}, (\bullet, -1)) \in \mathsf{proj}(\sigma'^\bullet)\}$$

hence making the diagram commutes.

If the last rule of π is a $⅋$ on the left hand side, then its interpretation as a sequent remains unchanged. That is, $[\![\pi]\!] = [\![\pi']\!]$. On the other hand, it is translated as the application of a left $\otimes$-rule on $(\pi')^F$, that also leaves the strategy, and the arena, unchanged. So $\sigma = \sigma'$ and hence $\mathsf{proj}(\sigma) = \mathsf{proj}(\sigma')$. We remind the rule and its translation in the table below.

$$\dfrac{\dfrac{\pi'}{\vdash \Gamma, M, N; \Pi}}{\vdash \Gamma, M ⅋ N; \Pi}\ ⅋ \qquad \dfrac{\dfrac{(\pi')^F}{\Gamma^F, (M^\perp)^F, (N^\perp)^F \vdash (\Pi)^F}}{\Gamma^F, (M^\perp)^F \otimes (N^\perp)^F \vdash (\Pi)^F}\text{ Left }\otimes$$

since $((M ⅋ N)^\perp)^F = (M^\perp \otimes N^\perp)^F = ((M^\perp)^F \otimes (N^\perp)^F))$

$$\dfrac{\dfrac{\pi'}{\vdash \Gamma, P, Q; \Pi}}{\vdash \Gamma, P ⅋ Q; \Pi}\ ⅋ \qquad \dfrac{\dfrac{(\pi')^F}{\Gamma^F, \neg(P)^F, \neg(Q)^F \vdash (\Pi)^F}}{\Gamma^F, \neg(P)^F \otimes \neg(Q)^F \vdash (\Pi)^F}\text{ Left }\otimes$$

since $((P ⅋ Q)^\perp)^F = (P^\perp \otimes Q^\perp)^F = (\neg(P)^F \otimes \neg(Q)^F)$

We now deal in the case where the last rule of π is a $\otimes$ rule. Then it is invariably translated by $(.)^F$ into a $\otimes$-rule. We remind here the rule and its translation:

$$\dfrac{\dfrac{\pi_1}{\vdash \Gamma; Q} \qquad \dfrac{\pi_2}{\vdash \Gamma; P}}{\vdash \Gamma, \Gamma'; P \otimes Q}\otimes \qquad \dfrac{\dfrac{(\pi_1)^F}{(\Gamma)^F \vdash (P)^F} \qquad \dfrac{(\pi_2)^F}{(\Delta)^F \vdash (Q)^F}}{(\Gamma)^F, (\Delta)^F \vdash (P)^F \otimes (Q)^F}\text{Right}\otimes$$

$$\text{since } (P \otimes Q)^F = P^F \otimes Q^F.$$

$$\dfrac{\dfrac{\pi_1}{\vdash \Gamma; M} \qquad \dfrac{\pi_2}{\vdash \Delta; N}}{\vdash \Gamma, \Delta; M \otimes N}\otimes \qquad \dfrac{(\Gamma)^F \vdash \neg(M^\perp)^F \qquad (\Delta)^F \vdash \neg(N^\perp)^F}{(\Gamma)^F, (\Delta)^F \vdash \neg(M^\perp)^F \otimes \neg(N^\perp)^F}\otimes$$

$$\text{since } (M \otimes N)^F = (\neg(M^\perp)^F \otimes \neg(N^\perp)^F).$$

Therefore, the interpretation of π is $[\![\pi_1]\!] \otimes [\![\pi_2]\!]$. That is:

$$[\![\pi]\!] = \{(x_\Gamma.x_\Delta.x_P.x_Q) \mid (x_\Gamma.x_P) \in [\![\pi_1]\!], (x_\Delta.x_Q) \in [\![\pi_2]\!],\ (x_\Gamma, x_P)\#_{\mathsf{pol}}(x_\Delta, x_Q)\}$$

Furthermore, the strategy $\sigma = \sigma_1 \otimes \sigma_2$, will reach the maximal positions $\sigma^\bullet \star_{\mathsf{pol}} \tau^\bullet$, as $\sigma^\bullet \simeq \mathsf{Frugal}(\sigma_1^\bullet \,\overline{\otimes}_{\mathbb{A}_{\mathsf{cells}}}\, \sigma_2^\bullet)$, therefore:

$$\mathsf{proj}(\sigma) = \{(x_\Gamma.x_\Delta.x_P.x_Q) \mid (x_\Gamma.x_P) \in \mathsf{proj}(\sigma_1),\ (x_\Delta.x_Q) \in \mathsf{proj}(\sigma_2),\ (x_\Gamma.x_P)\#_{\mathsf{pol}}(x_\Delta.x_Q)\}$$

The case where the formula on the right hand side is M, N is dealt on a equal footing. We now treat the case where the last rule of π is a &.

$$\dfrac{\dfrac{\pi_1}{\vdash \Gamma, P; \Pi} \qquad \dfrac{\pi_2}{\vdash \Gamma, Q; \Pi}}{\vdash \Gamma, P \,\&\, Q; \Pi}\& \qquad \dfrac{(\Gamma)^F, \neg(P^F) \vdash (\Pi)^F \qquad (\Gamma)^F, \neg(Q^F) \vdash (\Pi)^F}{(\Gamma)^F, \neg(P^F) \oplus \neg(Q^F) \vdash (\Pi)^F}\text{Left}\oplus$$

$$\text{since } ((P \,\&\, Q)^\perp)^F = (P^\perp \oplus Q^\perp)^F = \neg(P^F) \oplus \neg(Q^F)$$

$$\dfrac{\dfrac{\pi_1}{\vdash \Gamma, M; \Pi} \qquad \dfrac{\pi_2}{\vdash \Gamma, N; \Pi}}{\vdash \Gamma, M \,\&\, N, X; \Pi}\& \qquad \dfrac{\Gamma^F, (M^\perp)^F \vdash (\Pi)^F \qquad \Gamma^F, (N^\perp)^F \vdash (\Pi)^F}{(\Gamma)^F, (M^\perp)^F \oplus (N^\perp)^F \vdash (\Pi)^F}\text{Left}\oplus$$

$$\text{since } ((M \,\&\, N)^\perp)^F = (M^\perp \oplus N^\perp)^F = (M^\perp)^F \oplus (N^\perp)^F$$

Then the proof of $\pi_1 \,\&\, \pi_2$ is interpreted as the union between both:

$$[\![\pi_1 \,\&\, \pi_2]\!] = \{(x_\Gamma.\mathsf{inl}(x_P).x_\Pi) \mid (x_\Gamma.x_P.x_\Pi) \in [\![\pi_1]\!]\}$$
$$\uplus \{(x_\Gamma.\mathsf{inr}(x_Q).x_\Pi) \mid (x_\Gamma.x_Q.x_\Pi) \in [\![\pi_2]\!]\}$$

Similarly, looking at the strategy σ interpreting the proof π, then $\sigma^\bullet \simeq \mathsf{inl}(\sigma_1) \uplus \mathsf{inr}(\sigma_2)$. Depending on which side of the $\oplus$ the opponent is going to play its first move, the strategy is going to react according to σ_1 or σ_2. Therefore :

$$\begin{aligned}\mathsf{proj}(\sigma^\bullet) =& \{(x_\Gamma.\mathsf{inl}(x_P).x_\Pi) \mid (x_\Gamma.\mathsf{inl}(x_P).x_\Pi) \in \mathsf{proj}(\sigma_1^\bullet)\} \\ &\uplus \{(x_\Gamma.\mathsf{inr}(x_Q).x_\Pi) \mid (x_\Gamma.x_Q.x_\Pi) \in \mathsf{proj}(\sigma_2^\bullet)\}\end{aligned}$$

Finally, we address the $\oplus$-rule. If the last rule of π is an $\oplus$-rule, then it is interpreted in $(\pi)^F$ as an $\oplus$-rule as well. We treat the case where the last rule of π is $\oplus_1$. For instance, we present the translation where the formulas are both positive or negative.

$$\dfrac{\dfrac{\pi'}{\vdash \Gamma; P}}{\vdash \Gamma; P \oplus Q}\oplus_1 \qquad \dfrac{\dfrac{(\pi')^F}{(\Gamma)^F \vdash P^F}}{(\Gamma)^F \vdash P^F \oplus Q^F}\text{Right }\oplus_1$$
$$\text{since } (P \oplus Q)^F = P^F \oplus Q^F$$

$$\dfrac{\dfrac{\pi'}{\vdash \Gamma; M}}{\vdash \Gamma; M \oplus N}\oplus_1 \qquad \dfrac{\dfrac{(\pi')^F}{(\Gamma)^F \vdash \neg(M^\perp)^F}}{(\Gamma)^F \vdash (\neg(M^\perp)^F) \oplus (\neg(N^\perp)^F)}\text{Right }\oplus_1$$
$$\text{since } (M \oplus N)^F = (\neg M^\perp)^F \oplus (\neg(M^\perp)^F)$$

The nominal relation interpreting π will be the left injection of the one interpreting π', that is :

$$[\![\pi]\!]_{\mathsf{NomLinRelPol}} = \{(x_\Gamma.\mathsf{inl}(x_P) \mid (x_\Gamma.x_P) \in [\![\pi']\!]_{\mathsf{NomLinRelPol}}\}$$

Similarly, the strategy σ interpreting $(\pi)^F$ will act as $\sigma' = [\![(\pi')^F]\!]_{\mathsf{TTSFInn}}$, but going on the left branch of the $\oplus$ in its first P-move. Therefore, the following holds:

$$\mathsf{proj}(\sigma) = \{(x_\Gamma, \mathsf{inl}(x_P)) \mid (x_\Gamma, x_P) \in \mathsf{proj}(\sigma')\}$$

This is coherent with the interpretation of π through $[\![.]\!]_{\mathsf{NomLinRelPol}}$, as it satisfies the same equality. □

We remind the $[\![.]\!]_{\mathsf{NomLinRelPol}}$ is a functor, that is, is respects the equivalence of proofs of linear of linear logic ($\pi \sim \pi' \Rightarrow [\![\pi]\!]_{\mathsf{NomLinRelPol}} = [\![\pi']\!]_{\mathsf{NomLinRelPol}}$). Therefore, the interpretation:

$$\pi \xrightarrow{(.)^F} (\pi)^F \xrightarrow{[\![.]\!]_{\mathsf{TTSFInn}}} [\![(\pi)^F]\!]_{\mathsf{TTSFInn}} \xrightarrow{\mathsf{proj}} \mathsf{proj}[\![(\pi)^F]\!]_{\mathsf{TTSFInn}}$$

is a denotation function that respects the equivalence of proofs of linear logic. That is, if $\pi \sim \pi'$ then $\mathsf{proj}[\![(\pi)^F]\!]_{\mathsf{TTSFInn}} = \mathsf{proj}[\![(\pi')^F]\!]_{\mathsf{TTSFInn}}$. Furthermore, one should ensure that it acts as a functor. That is:

$$\mathsf{proj}[\![(\pi;\pi')^F]\!]_{\mathsf{TTSFInn}} = \mathsf{proj}[\![(\pi)^F]\!]_{\mathsf{TTSFInn}} ;_{\mathsf{NomLinRelPol}} \mathsf{proj}[\![(\pi')^F]\!]_{\mathsf{TTSFInn}}$$

If follows from $\mathsf{proj}[\![(\pi;\pi')^F]\!]_{\mathsf{TTSFInn}} = [\![\pi;\pi']\!]_{\mathsf{NomLinRelPol}} = [\![\pi]\!]_{\mathsf{NomLinRelPol}};[\![\pi']\!]_{\mathsf{NomLinRelPol}}$.

6.3.4 Full completeness for linear logic

We are now in position of presenting the full completeness result for linear logic. We work within the category of nominal annotated polarised separated relations. Given some objects A, B, where A, B are seen as formulas of linear logic, we can translate them as formulas of tensorial logic $(A)^F, (B)^F$. Now each strategy in $\mathsf{TTSFInn}((A)^F, (B)^F)$ is a denotation of a proof of tensorial logic $\pi : (A)^F \vdash (B)^F$. To this one corresponds a proof $(\pi)^I : (A)^F \vdash (B)^F$, which has, as denotation $\mathsf{proj}((\pi)^F)$. Therefore, we can select precisely those nominal relations that do correspond to proofs, and obtain a full completeness result.

Definition 6.8. *The category* NomMall *is the star-autonomous category that has same objects as* NomLinRelPol *and morphisms nominal linear polarised relations that arise as projections of strategies of* TTSFInn.

Proposition 6.9. NomMall *is fully complete for multiplicative additive linear logic.*

In NomMall, a map $A \to B$ is a nominal polarised relation $\mathcal{R}$ such that there exists $\sigma \in \mathsf{TTSFInn}(A^F, B^F)$, with $\mathcal{R} = \mathsf{proj}(\sigma)$. We proved in the section above that this forms a category, and, by definition, each morphism in it is the denotation of a proof of linear logic. NomMall is precisely the sub-category of NomLinRelPol that corresponds to the image of the functor $[\![.]\!] : \mathsf{MALL} \to \mathsf{NomLinRelPol}$. As this functor is a star-autonomous one, so is NomMall. More precisely, NomMall is a sound and fully complete model of MALL.

Finally, one might wonder, if, as in the case of TTSFInn, the category obtained is the free star-autonomous category with products. Unfortunately, the answer is negative. A simple counter-example is formed by these two proofs together with their denotations:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash I}\,I}{\vdash \bot_1, I_1}\,\bot \qquad \dfrac{\dfrac{}{\vdash I_2}\,I}{\vdash \bot_2, I_2}\,\bot}{\vdash \bot_1 \otimes \bot_2, I_1, I_2}\,\otimes}{\vdash \bot_1 \otimes \bot_2, I_2, I_1}\,\text{Exchange}}{\vdash \bot_1 \otimes \bot_2, I_2 \parr I_1}\,\parr \qquad\qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\vdash I_1}\,I}{\vdash \bot_1, I_1}\,\bot \qquad \dfrac{\dfrac{}{\vdash I_2}\,I}{\vdash \bot_2, I_2}\,\bot}{\vdash \bot_1 \otimes \bot_2, I_1, I_2}\,\otimes}{\vdash \bot_1 \otimes \bot_2, I_1 \parr I_2}\,\parr$$

Let us name π_1 the left one, π_2 the other. Then $\neg(\pi_1 \sim \pi_2)$. This is notably proved in [47]. On the other hand, they are both denoted by the same relation:

$$[\![\pi_1]\!] = [\![\pi_2]\!] = \{(\bullet, -1).(\bullet, -1).(\bullet, 1).(\bullet.1)\}$$

This mismatch proves that the the nominal relations are too simple, too flat, to fully distinguish between distinct proofs of linear logic. Furthermore, this full completeness is obtained through the medium of tensorial logic: we do not have a direct characterisation of nominal relations that are denotations of proofs of linear logic.

Part III

Static Full Completeness and Conclusion

Chapter 7

Revisiting the Concurrent Model

7.1 Concurrent nominal games

At the end of the second part, we did not succeed to provide a characterisation of nominal relations that arise from proofs of linear logic directly. Therefore, in this section, we explore a different approach, whose starting point relies on concurrent games similar to those desribed in [10]. Those games differ vastly from those presented in the previous sections, since the tensor acts by putting the arenas in parallel, and the negation is modelled by changing an abstract notion of polarity, and not adding any additional moves. Consequently, they offer an ideal candidate for a full completeness result without relying on quotient.

Therefore, we start this section by reformulating the original definitions of concurrent games within the nominal model, modifying them slightly in passing. We present them as a refinement of nominal polarised relations. We then sharpen the model by adding some constraints our strategies must obey. One to ensure they define proof structures properly, the second being the winning condition of the original model, that is, totality. However, we prove that this is not enough for a full completeness result. Indeed, the original model makes full use of di-natural transformations, on which it relies, to reach full completeness. Di-natural transformations enable one to model the atomic types by a variety of different arenas, and, by choosing them appropriately, one can enforce certain properties. On the other hand, our model relies on a fixed arena for each atomic type, and this one is not appropriate for imposing all the needed properties.

We take the point of view that the arena chosen for the atomic type is the right one, as it is seemingly the same as in the previous model of tensorial logic, and therefore attempt to change the model accordingly in order for it to be fully complete. We analyse why the original model fails, and explain how to patch it. That leads to an entirely new model, where polarities are strictly taken into account, but such that not all positions have fully defined polarity. We then characterise those relations that behave well, and prove that they form a fully complete model of MLL^{-}.

This is, to our knowledge, the first full completeness result obtained for MLL^- not based on 2-categorical tools, neither quotient, nor proof structures. The closest result we are aware of is a result of full completeness for $\mathsf{MLL}^- + \mathsf{MIX}$ via experiments on coherence spaces [26] [83]. This result refines it in two ways: experiments are now encoded in a categorical approach, and we dispose of the MIX-rule. The final model could be presented directly without any reference to the original concurrent games model. However, as it resulted from a careful study of the behaviour of concurrent strategies, we believe it is only fair to briefly present it, and show how we reasoned about it.

Finally, we enrich our model with a notion of hypercoherence. This allows us to enrich our model with additive connectives, and model MALL^-. We notably prove that the way the hypercoherence model deals with the additives is reminiscent of the concurrrent games model, by proving each hypercoherence gives rise to a concurrent operator. Finally, we revisit a former result established by Blute in [16], that proved that di-natural transformations over a category of double glued hypercoherence spaces lead to a fully complete model of MALL^-. We translate the work done in our setting, allowing us to prove that the final model we obtain is fully complete for MALL^-. Again, this is the first model we are aware of that is fully complete for MALL^-, and does not rely on 2-categorical tools, proof structures, or quotient.

7.1.1 Polarised nominal qualitative domains

As explained in the end of Section 7.2.3, nominal relations are not fully complete since they are too "flat". Indeed, two non equivalent proofs of $\vdash \bot \otimes \bot, I, I$ were modelled by the same relation $\mathcal{R} = \{(\bullet, -1).(\bullet, -1).(\bullet, 1).(\bullet, 1)\}$, this one not taking into account the dynamics that happens in the proofs: i the first proof the left $\bot$ is linked to the left I, whereas in the second proof the left $\bot$ is linked to the right one. Therefore, we would like to establish a more local control on the relation. That is, we would like to consider only certain parts of the formula and disregard others (for instance, in the above case, the left $\bot$ without the right one). This could solve the problematic case above by discriminating between the two relations, highlighting which $\bot$ is linked to which I. Consequently, we consider concurrent games, where the opponent will bring the negative primes, and the proponent the positive ones. This allows us to consider elements of the relation where only some negative formulas have been provided while others are missing, highlighting the dependencies between negative and positive occurrences. We first focus on MLL^-.

We will be adding a bottom element $\bot$ to the denotation of each atomic formula of Chapter 3, creating a nominal ordered set with a minimal element. This $\bot$ element intuitively corresponds to the "diverging computation" coming from the partiality monad [75]. We remind some basic definitions of nominal domain theory. These were already presented in Section 4.5. Let us remind that we work with orbit-finite nominal sets, and hence every set presented is deemed to have only a finite number of orbits. A nominal poset $(\mathbb{D}, \sqsubseteq)$ is a nominal set whose partial order relation is nominal. A nominal domain is a nominal poset with a minimal element $\bot$. A finitely supported set of elements S is compatible, written $\uparrow S$ if the subset S is bounded. A domain is bounded complete if every compatible S has a least upper bound, denoted $\bigsqcup S$. A bounded complete domain automatically has meets, written $\sqcap$. A prime of a bounded complete domain is an element $\mathsf{p} \in \mathbb{D}$ such that $\mathsf{p} \leq \bigsqcup S \Rightarrow \exists x \in S.\mathsf{p} \sqsubseteq x$. $\mathsf{Pr}(\mathbb{D})$ stands for the nominal subset of primes of $\mathbb{D}$. Furthermore, $\mathbb{D}$ is prime algebraic if it is bounded complete and any element can be written as a finite join of primes.

Definition 7.1. • *A domain $(\mathbb{D}, \sqsubseteq, \bot)$ is* ***qualitative*** *if it is prime algebraic and no distinct primes are related by $\sqsubseteq$.*

- *A qualitative domain is a* ***coherence*** *domain if, given two elements x, y, written as set of primes $x = \sqcup_i \mathsf{p}_i$, $y = \sqcup_j \mathsf{q}_j$, then $\forall i, j.\mathsf{p}_i \uparrow \mathsf{q}_i \Rightarrow x \uparrow y$.*
- *A* ***polarised qualitative domain*** *$(\mathbb{D}, \sqsubseteq, \bot, \lambda)$ is a qualitative domain $(\mathbb{D}, \leq, \bot)$ together with a nominal polarity function $\lambda : \mathrm{Pr}(\mathbb{D}) \to \{-1, 1\}$.*
- *A coherence domain is* ***polarised*** *if it is polarised as a qualitative domain and furthermore $\lambda(\mathsf{p}) \neq \lambda(\mathsf{q}) \Rightarrow \mathsf{p} \uparrow \mathsf{q}$, where $\mathsf{p}, \mathsf{q} \in \mathrm{Pr}(\mathbb{D})$.*

In a qualitative domain, all primes are directly above $\bot$. A qualitative domain that is a coherence domain is perfectly described by its set of primes together with a binary nominal coherence relation between them, that displays which primes are compatible. As each element of a qualitative domain can be seen as a set of primes, we sometimes write $\mathsf{p} \in x$ for $\mathsf{p} \sqsubseteq x$. We work with polarised coherence domains, where each prime p is given a polarity. We write $\mathsf{Pos}(\mathsf{p})$ if the prime p is positive (that is, $\lambda(\mathsf{p}) = 1$), $\mathsf{Neg}(\mathsf{p})$ if it is negative. Similarly, we write $\mathsf{Pos}(x)$ if x is a finite union of P-primes, and $\mathsf{Neg}(x)$ if x is a finite union of negative primes.

Proposition 7.2. *Let S, T bounded set of negative and positive primes respectively in a polarised coherence domain. Then $S \cup T$ is bounded.*

The proof is immediate. We assign to each atomic formula a polarised coherence domain as described in the introduction:

$$\llbracket X \rrbracket_{\mathsf{Qual}} = (\mathbb{A}_X \uplus \{\bot\}, \{\bot\} \sqsubset \mathbb{A}_X, \bot, \lambda(a) = 1)$$
$$\llbracket I \rrbracket_{\mathsf{Qual}} = (\{\bot\}, \{(\bot, \bot)\}, \bot, \lambda = \emptyset)$$

Note that $\mathrm{Pr}(\llbracket X \rrbracket_{\mathsf{Qual}}) = \mathbb{A}_X$. It would have been more consistent to define $\llbracket I \rrbracket_{\mathsf{Qual}} = (\{\bot, \top\}, \{\bot\} \sqsubset \{\top\}, \bot, \lambda(\top) = 1)$, however, as we restrict our interest to MLL^- (that is, without units) for the moment, a "strict" unit as presented above is perfectly adapted. There is a straightforward negation for polarised qualitative domains, consisting in inverting the polarity of primes.

$$(\mathbb{D}, \sqsubseteq, \bot, \lambda)^\perp = (\mathbb{D}, \sqsubseteq, \bot, -\lambda).$$

Notably the negation is involutive. It allows us to define the denotations to $X^\perp$ and $\perp$ as expected.

$$[\![X^\perp]\!]_{\mathsf{Qual}} = [\![X]\!]_{\mathsf{Qual}}^\perp \qquad [\![\perp]\!]_{\mathsf{Qual}} = [\![I]\!]_{\mathsf{Qual}}^\perp = [\![I]\!]_{\mathsf{Qual}}$$

Given an element x of a qualitative polarised domain, there is a unique decomposition of x into primes: $x = \bigsqcup\{\mathsf{p} \in \mathrm{Pr}(\mathbb{D}) \mid \mathsf{p} \sqsubseteq x\}$. We write $\mathsf{pos}(x)$ for $\bigsqcup\{\mathsf{p} \in \mathrm{Pr}(\mathbb{D}) \mid \mathsf{p} \sqsubseteq x, \mathsf{Pos}(\mathsf{p})\}$, and $\mathsf{neg}(x)$ for its negative counterpart. We hence have $\forall x \in \mathbb{D}.x = \mathsf{neg}(x) \sqcup \mathsf{pos}(x)$.

Lemma 7.3.
- *Given $x, y \in \mathbb{D}$ a qualitative polarised domain, $x \uparrow y \Rightarrow \mathsf{pos}(x \sqcup y) = \mathsf{pos}(x) \sqcup \mathsf{pos}(y)$ and similarly $\mathsf{neg}(x \sqcup y) = \mathsf{neg}(x) \sqcup \mathsf{neg}(y)$.*
- *$\mathsf{pos}(x \sqcap y) = \mathsf{pos}(x) \sqcap \mathsf{pos}(y)$ and similarly $\mathsf{neg}(x \sqcap y) = \mathsf{neg}(x) \sqcap \mathsf{neg}(y)$.*

The tensor product of two nominal domains is given by their cartesian product. In particular, note that $\mathrm{Pr}(\mathbb{D}_1 \times \mathbb{D}_2) = (\mathrm{Pr}(\mathbb{D}_1) \times \{\perp_2\}) \uplus (\{\perp_1\} \times \mathrm{Pr}(\mathbb{D}_2)) \simeq \mathrm{Pr}(\mathbb{D}_1) \uplus \mathrm{Pr}(\mathbb{D}_2)$, and that the disjoint union is the coproduct in the category of sets and functions. We can then form the function $\lambda_1 \oplus \lambda_1 : \mathrm{Pr}(\mathbb{D}_1 \times \mathbb{D}_2) \to \{-1, 1\}$. Formally, we reach:

$$(\mathbb{D}_1, \sqsubseteq_1, \perp_1, \lambda_1) \times (\mathbb{D}_2, \sqsubseteq_2, \perp_2, \lambda_2) = (\mathbb{D}_1 \times \mathbb{D}_2, (\sqsubseteq_1 \times \sqsubseteq_2), (\perp_1, \perp_2), \lambda_1 \oplus \lambda_2).$$

In particular, this allows us to assign to each formula of MLL its associated qualitative polarised domain:

$$\begin{aligned} [\![A \otimes B]\!]_{\mathsf{Qual}} &= [\![A]\!]_{\mathsf{Qual}} \times [\![B]\!]_{\mathsf{Qual}} \\ [\![A \parr B]\!]_{\mathsf{Qual}} &= ([\![A]\!]_{\mathsf{Qual}}^\perp \times [\![B]\!]_{\mathsf{Qual}}^\perp)^\perp = [\![A]\!]_{\mathsf{Qual}} \times [\![B]\!]_{\mathsf{Qual}} \end{aligned}$$

7.1.2 Finite supported relations

We introduce here some terminologies that will be useful for the sequel.

Let us suppose that we consider a relation $\mathcal{R} : [\![(X_1 \otimes X_2) \parr X_1^\perp \parr X_2^\perp]\!]_{\mathsf{Qual}}$ and we want to look at how a "strategy" (that is morally a relation) reacts if it is fed an input in $X_1^\perp$ but not in $X_2^\perp$. To model this case, we will consider a "counter strategy" that plays a name in $X_1^\perp$ but not in $X_2^\perp$, corresponding to a relation of the form $\{(\perp, \perp, a, \perp)\}$. However, this one does not have empty support.

To remedy this issue, we will, in the rest of this section, often work within the category FinRel of nominal sets and finitely supported relations between them. If $\mathsf{NomLinRelPol}$ is obviously a subcategory FinRel, they do not share the same monoidal product. Indeed, imagine two relations $I \to A$ with same support. Then we cannot form a product tensor of these in $I \to A \star A$. On the other hand, it certainly works if we take the cartesian product as tensor product.

Definition 7.4. FinRel *is the category which has:*
- *Nominal sets as objects.*

- *Finitely supported relations as morphisms*

$(\mathsf{FinRel}, \times, 1)$ *is a category with products, where* $\times$ *is the cartesian product and* 1 *is a singleton.*

More precisely, FinRel is star autonomous with products, the tensor product being the cartesian product, the product being the disjoint union, and the negation the identity.

In addition, we recall that there is a faithful functor of star-autonomous category from NomLinRelPol to FinRel, namely the $\widehat{\cdot}$ previously defined in section 3, together with forgetting the polarities. $\widehat{\cdot}$ was originally defined as a functor NomLinRelPol $\rightarrow$ LaxNomLinPol, and this one naturally forms a sub star-autonomous category of FinRel. This way the morphisms of NomLinRelPol can be projected into morphisms of FinRel. We know that the morphisms of NomLinRelPol behave well, in the sense that they define sets of axiom links on the formulas they relate. Therefore, in the future, we shall relate our morphisms with those of NomLinRelPol.

7.1.3 Closure operators

Given a nominal qualitative domain $\mathbb{D}$, we call $\mathbb{D}^\top$ its lattice completion. It consists in adding an element $\top$ to $\mathbb{D}$, such that $\forall x \in \mathbb{D}, x \sqsubset \top$. This turns it into a nominal complete lattice.

Definition 7.5. *A* ***nominal complete lattice*** $(L, \sqsubseteq)$ *is a nominal poset* $(L, \sqsubseteq)$ *such that every finitely supported subset of L has a greatest lower bound and a least upper bound.*

In particular, a nominal complete lattice has a minimal element $\bigsqcup \emptyset$ and a maximal element $\bigsqcap \emptyset$.

Definition 7.6. *Given a nominal partially ordered set* $(\mathbb{D}, \sqsubseteq)$*, we call* $\mathbb{D}^\top$ *its completion by maximal element.* $\mathbb{D}^\top = (\mathbb{D} \uplus \{\top\}, \sqsubseteq^\top)$ *where* $\sqsubseteq^\top$ *is defined by* $x \sqsubseteq^\top y$ *if:*

- $y \neq \top$ *and* $x \sqsubseteq y$.
- $y = \top$.

Proposition 7.7. *Given a qualitative domain* $\mathbb{D}$*, then* $\mathbb{D}^\top$ *is a lattice.*

Proof. First we show that every finitely supported subset S has a least upper bound. If $S \subseteq \mathbb{D}$ and $S\uparrow$, then it follows the definition of bounded completeness of $\mathbb{D}$. If not, then $\top$ is an upper bound of S. By definition, it is the only one, and hence, the least one. To show that each finitely supported subset has a greatest lower bound, we consider the set of elements below it. This set of elements is bounded, finitely supported, and has a least upper bound, that provides the greatest lower bound of the original set. □

We write $\mathbb{D} \hookrightarrow \mathbb{D}^\top$ for the natural injection of $\mathbb{D}$ into $\mathbb{D}^\top$. Given two completed partially ordered sets $\mathbb{D}^\top, \mathbb{F}^\top$, we define their product as follows:

$$\mathbb{D}^\top \times \mathbb{F}^\top = (\mathbb{D} \times \mathbb{F})^\top$$

Given $x \in \mathbb{D}^\top, y \in \mathbb{F}^\top$, we will write $(x, y) \in (\mathbb{D}\times\mathbb{F})^\top$ for its natural element if $x \neq \top, y \neq \top$, and for $\top$ if $x = \top$ or $y = \top$. Similarly, we define the projections $\pi_{\mathbb{D}^\top} : (\mathbb{D} \times \mathbb{F})^\top \to \mathbb{D}^\top$ by $\pi_{\mathbb{D}^\top}(x, y) = x$ if $(x, y) \neq \top$ and $\pi_{\mathbb{D}^\top}(\top) = \top$. Therefore, $\pi_{\mathbb{D}^\top}(x, y)$ might be different than x if $y = \top$.

Definition 7.8. *A* ***closure operator*** *on a lattice L, written $\sigma : L$, is a function with finite support $\sigma : L \to L$ that satisfies the following properties:*

- $\sigma^2(x) = \sigma(x)$
- $\sigma(x) \sqsupseteq x$
- $x \geq y \Rightarrow \sigma(x) \sqsupseteq \sigma(y)$

Furthermore, a closure operator on $\mathbb{D}^\top$, where $\mathbb{D}$ is a polarised qualitative domain, is ***positive****, written $\mathsf{Pos}(\sigma)$, or P-closure operator if:*

- $\forall x \in \mathbb{D}.\sigma(\mathsf{neg}(x)) \neq \top$.
- $\forall x \in \mathbb{D}.\sigma(x) \neq \top \Rightarrow (\forall \mathsf{p} \in \mathsf{Pr}(\mathbb{D}).(\mathsf{p} \sqsubseteq \sigma(x) \ \wedge\ \neg(\mathsf{p} \sqsubseteq x)) \ \Rightarrow\ \mathsf{Pos}(\mathsf{p}))$
- $\forall x \in \mathbb{D}$, $\sigma(x) = \sigma(\mathsf{neg}(x)) \sqcup \mathsf{pos}(x)$.

Alternatively, we also speak of P-strategy (or simply strategies) for positive closure operators.

The concept of O-strategy , or counter-strategy , is defined along the same lines, substituting Neg for Pos, and neg for pos. The idea behind the definition is that σ intuitively acts like a strategy in a game. Given a position x, it will play some (maybe none, maybe several) player moves, embodied here by positive primes, until it reaches a position $\sigma(x)$. Then, once it lacks the information (opponent-primes) to move further, it stops. The more information σ is given the more moves it will be able to perform. Note that the second condition could be rewritten by $\sigma(x) \neq \top \Rightarrow \mathsf{neg}(\sigma(x)) = \mathsf{neg}(x)$, and as $\sigma(x) = \sigma(\mathsf{neg}(x)) \sqcup \mathsf{pos}(x)$, and $\sigma(\mathsf{neg}(x)) \neq \top$, this could be further simplified as $\forall x.\mathsf{neg}(\sigma(\mathsf{neg}(x))) = \mathsf{neg}(x)$.

We will now speak about composition of closure operators, and some of their properties. As these have already been studied in [10, 3, 74], most proofs will be skipped. Given a closure operator $\sigma : \mathbb{D}^\top \times \mathbb{F}^\top$ and an element $x \in \mathbb{D}^\top$, we write $\sigma(x, \_)$ for the function $\mathbb{F}^\top \to \mathbb{F}^\top$ defined by $y \mapsto \pi_{\mathbb{F}^\top}(\sigma(x, y))$, and similarly for $\sigma(\_, z) : \mathbb{D}^\top$. These are closure operators [10]. For $\sigma, \tau : \mathbb{D}^\top$, we write $\langle \sigma, \tau \rangle \in \mathbb{D}^\top$ for the element:

$$\langle \sigma, \tau \rangle = \bigsqcup\{(\sigma \circ \tau)^n(\bot) \mid n \in \mathbb{N}\}.$$

As the domain we are working on are orbit-finite, there is no infinite chain. Therefore, the closure operators are automatically continuous, entailing the existence of the limit of the chain defining $\langle \sigma, \tau \rangle$.

Remark 7.9. *$\langle \_, \_ \rangle$ is commutative: $\langle \sigma, \tau \rangle = \langle \tau, \sigma \rangle$.*

Lemma 7.10. *If σ is positive, τ is negative, then $\langle \sigma, \tau \rangle \neq \top$.*

Proof. The proof is done by induction on the growing chain y^n defined as follows:

- $y^0 = \bot$.
- $y^{n+1} = \sigma(y^n)$ if n is even.
- $y^{n+1} = \tau(y^n)$ if n is odd.

We prove by induction that for each n in the chain $\sigma(y^n) = \sigma(\mathsf{neg}(y^n))$ and $\tau(y^n) = \tau(\mathsf{pos}(y^n))$. This is true for the case case $n = 0$. We do the induction case, dealing first with σ. If n is even then $\sigma(y^{n+1}) = \sigma(\sigma(y^n)) = \sigma(y^n)$, making the induction straightforward. In the case where n is odd, $\sigma(y^{n+1}) = (\sigma(\tau(y^n))$. As σ is positive, $\sigma(y^{n+1}) = \mathsf{pos}(y^{n+1}) \sqcup \sigma(\mathsf{neg}(y^n))$. As τ is negative, and $\tau(y^n) \neq \top$ by induction, $\mathsf{pos}(\tau(y^n)) = \mathsf{pos}(y^n)$. Therefore, $\sigma(y^{n+1}) = \sigma(\mathsf{neg}(y^{n+1}))\sqcup\mathsf{pos}(y^n)$. On the other hand, $\sigma(y^n) = \sigma(\mathsf{neg}(y^n))$. Therefore, $\mathsf{pos}(y^n) \subseteq \sigma(y^n) \subseteq \sigma(y^{n+1})$. Consequently, $\sigma(y^{n+1}) = \sigma(\mathsf{neg}(y^{n+1}))$ as required. The proof for τ is done on an equal footing. In particular, this leads to $\sigma(y^n) \neq \top$, $\tau(y^n) \neq \top$, and hence $y^n \neq \top$. Consequently, $\langle\sigma, \tau\rangle \neq \top$. □

Lemma 7.11. *Let $\sigma : \mathbb{D}^\top$ be a positive closure operator, and y such that* $\mathsf{Pos}(y)$*. Then* $\sigma(x)\sqcup y = \sigma(x \sqcup y)$.

Proof. Since $\mathsf{Pos}(y)$, it entails $y = \mathsf{pos}(y)$ and $\mathsf{neg}(y) = \bot$. If $x \uparrow y$ we have $\sigma(x) \sqcup y = \sigma(\mathsf{neg}(x)) \sqcup \mathsf{pos}(x) \sqcup y = \sigma(\mathsf{neg}(x \sqcup y)) \sqcup \mathsf{pos}(x \sqcup y) = \sigma(x \sqcup y)$. In the case where $\neg(x \uparrow y)$ we have $\sigma(x) \sqcup y = \top = \sigma(x \sqcup y)$. □

We will now address composition of closure operators, and display some well-known properties from the literature about them. Given three lattices L, M, N and $\sigma : L \times M$, $\tau : M \times N$, then given $(x, z) \in L \times N$, we define $y \in M$, the witness of interaction, by $y = \langle\sigma(x, \_), \tau(\_, z)\rangle$. It provides us the key to define their composition at x, z.

$$\sigma; \tau(x, z) = (\pi_L(\sigma(x, y)), \pi_N(\tau(y, z)))$$

A quite puzzling fact about closure operators is that they are relational. More specifically, writing $\sigma^\bullet$ for their sets of positions:

$$\begin{aligned}\sigma^\bullet &= \{x \mid \sigma(x) = x\}\\ &= \{x \mid \exists y.\sigma(y) = x\}\end{aligned}$$

then σ is entirely defined by its set $\sigma^\bullet$ through the following equation [64]:

$$\sigma(x) = \bigsqcap\{y \in \sigma^\bullet \mid y \sqsupseteq x\}.$$

Furthermore, their relational composition is equivalent to their dynamic one.

$$(\sigma; \tau)^\bullet = \sigma^\bullet ;_{\mathsf{Rel}} \tau^\bullet$$

Finally, seeing them as sets of positions, we can give an alternative definition of $\langle\sigma, \tau\rangle$.

$$\langle\sigma,\tau\rangle = \mathsf{min}(\sigma^{\bullet} \cap \tau^{\bullet}).$$

More on them can be found in [10, 3, 74, 64].

Definition 7.12. *The category* QualClo *is defined as having :*

- *Polarised coherence domains as objects.*
- *As morphisms:* $\mathbb{D} \to \mathbb{F}$, *equivariant P-closure operators* $\sigma : (\mathbb{D}^{\perp})^{\top} \times \mathbb{F}^{\top}$.
- *As identities* $\mathsf{id} : \begin{cases} (x,y) & \mapsto (x \sqcup \mathsf{neg}(y), y \sqcup \mathsf{neg}(x)) \text{ if } x \neq \top, y \neq \top \\ \top & \mapsto \top\ . \end{cases}$

We add the full proof that this forms a category below. Note that as the morphisms are equivariant, they have empty support and therefore cannot raise a name: $\nu(\sigma(x)) \subseteq \nu(x)$. Furthermore, as $x \sqsubseteq \sigma(x)$, this implies $\nu(x) = \nu(\sigma(x))$.

Proposition 7.13. QualClo *forms a category.*

Proof. We already know from the literature that the closure operators compose, and form a category. We simply need to specialise in the case of P-closure operators. We prove the following:

- The composition of P-closure operators leads to a P-closure operator.
- The morphism id is positive and indeed behaves as an identity.

We consider three domains $\mathbb{D}, \mathbb{F}, \mathbb{G}$, and their associated lattice completions $\mathbb{D}^{\top}, \mathbb{F}^{\top}, \mathbb{G}^{\top}$, and $\sigma : \mathbb{D}^{\top} \times \mathbb{F}^{\top}$, $\tau : (\mathbb{F}^{\top})^{\perp} \times \mathbb{G}^{\top}$ (to be consistent we should have picked $\sigma : (\mathbb{D}^{\perp})^{\top} \times \mathbb{F}^{\top}$, however we find it more convenient to work with the non-negated form). Before tackling the proof, we prove that given $x \in \mathbb{D}$ such that $\mathsf{Neg}(x)$, then $\sigma(x,\_) : \mathbb{F}^{\top}$ is a P-closure operator.

- $\sigma(x,\_)(\mathsf{neg}(y)) = \pi_{\mathbb{F}^{\top}}(\sigma(x,\mathsf{neg}(y)) = \pi_{\mathbb{F}^{\top}}(\sigma(\mathsf{neg}(x),\mathsf{neg}(y))) = \pi_{\mathbb{F}^{\top}}(\sigma(\mathsf{neg}(x,y))) \neq \top_{\mathbb{F}^{\top}}$ as $\sigma(\mathsf{neg}(x,y)) \neq \top_{\mathbb{D}^{\top}\times\mathbb{F}^{\top}}$ by positivity of σ.
- We consider $y \in \mathbb{F}$ and $\mathsf{p} \in \Pr(\mathbb{F})$ such that $\mathsf{p} \sqsubseteq \sigma(x,\_)(y)$ and $\mathsf{p} \not\sqsubseteq y$ and $\sigma(x,y) \neq \top$. Then by positivity of σ, $\mathsf{Pos}(\mathsf{p})$ as expected.
- $\sigma(x,\_)(y) = \pi_{\mathbb{F}^{\top}}(\sigma(\mathsf{neg}(x,y)) \sqcup \mathsf{pos}(x,y)) = \sigma(x,\_)(\mathsf{neg}(y)) \sqcup \mathsf{pos}(y)$.

So $\sigma(x,\_)$ satisfies the three properties of positivity and hence is a positive closure operator. Similarly, $\tau(\_,z)$ is a positive operator of $(\mathbb{F}^{\top})^{\perp}$, that is, a negative closure operator on $\mathbb{F}^{\top}$.

We start with the first point of positivity. Let $(x,z) \in \mathbb{D} \times \mathbb{G}$, such that $\mathsf{Neg}(x)$, $\mathsf{Neg}(z)$. Then $\sigma(x,\_)$ is a positive operator of $\mathbb{F}^{\top}$, and $\tau(\_,z)$ a negative one. Consequently, $y = \langle\sigma(x,\_),\tau(\_,z)\rangle \neq \top_{\mathbb{F}^{\top}}$. Furthermore, as $\sigma(x,y) = \sigma(\mathsf{neg}(x),\mathsf{neg}(y)) \sqcup \mathsf{pos}(x,y) = \sigma(\mathsf{neg}(x),\mathsf{neg}(y)) \sqcup (\perp,\mathsf{pos}(y))$. Therefore, as $\sigma(\mathsf{neg}(x),\mathsf{neg}(y)) = (x',y') \neq \top$, we have $\sigma(x,y) = (x',y') \sqcup (\perp,\mathsf{pos}(y)) = (x',y) \neq \top$ (since $\sigma(x,y) = (x',y)$ by definition). We can similarly prove that $\tau(y,z) \neq \top$, and finally $\sigma;\tau(x,z) \neq \top$.

Let $(x,z) \in \mathbb{D} \times \mathbb{G}$, and y the witness of interaction. Let us suppose that $\sigma;\tau(x,z) \neq \top$. Then for any $\mathsf{p} \in \Pr(\mathbb{D})$ such that $\mathsf{p} \notin x \wedge \mathsf{p} \in \pi_{\mathbb{D}^{\top}}(\sigma(x,y))$, $\mathsf{Pos}(\mathsf{p})$ as σ is positive, and similarly for any $\mathsf{p} \in \Pr(\mathbb{G})$ such that $\mathsf{p} \notin z \wedge \mathsf{p} \in \pi_{\mathbb{G}^{\top}}(\tau(y,z))$ by positivity of τ. So the second property of positivity is preserved. Now, let us consider the element $\sigma;\tau(\mathsf{neg}(x),\mathsf{neg}(z))$. The first thing we prove is that the witness of interaction y is the same for $\mathsf{neg}(x),\mathsf{neg}(z)$ and (x,z). This is proven by induction on the chains of w_i, y_i defined below such as $y = \langle\sigma(x,\_),\tau(\_,z)\rangle = \bigsqcup_i y_i$ and $w = \langle\sigma(\mathsf{neg}(x),\_),\tau(\_,\mathsf{neg}(z))\rangle = \bigsqcup_i w_i$.

- $y_0 = w_0 = \bot_{\mathbb{F}^\top}$.
- $y_{n+1} = \pi_{\mathbb{F}^\top}(\sigma(x,y_n))$, $w_{n+1} = \pi_{\mathbb{F}^\top}(\sigma(\mathsf{neg}(x),w_n))$ if n is even.
- $y_{n+1} = \pi_{\mathbb{F}^\top}(\tau(y_n,z)$, $w_{n+1} = \pi_{\mathbb{F}^\top}(\tau(w_n,\mathsf{neg}(z))$ if n is odd.

We prove $\forall i.w_i = y_i$. The base case holds as $w_0 = y_0 = \bot_M$. Now suppose we proved it up to an even n, and do the induction step to $n+1$ odd, as presented in the following equations:

$$\begin{aligned}
y_{n+1} &= \pi_{\mathbb{F}^\top}(\sigma(x,y_n))\\
&= \pi_{\mathbb{F}^\top}(\sigma(\mathsf{neg}(x),\mathsf{neg}(y_n)) \sqcup \mathsf{pos}(x) \sqcup \mathsf{pos}(y_n))\\
&= \pi_{\mathbb{F}^\top}(\sigma(\mathsf{neg}(x),\mathsf{neg}(y_n))) \sqcup \mathsf{pos}(y_n)\\
&= \pi_{\mathbb{F}^\top}(\sigma(\mathsf{neg}(x),y_n))\\
&= w_{n+1}
\end{aligned}$$

The even $n+1$ case is dealt with similarly. So finally :

$$\begin{aligned}
\pi_{\mathbb{D}^\top}(\sigma;\tau(x,z)) &= \pi_{\mathbb{D}^\top}(\sigma(x,y))\\
&= \pi_{\mathbb{D}^\top}(\sigma(\mathsf{neg}(x),\mathsf{neg}(y)) \sqcup \mathsf{pos}(x) \sqcup \mathsf{pos}(y))\\
&= \pi_{\mathbb{D}^\top}(\sigma(\mathsf{neg}(x),y)) \sqcup \mathsf{pos}(x)\\
&= \pi_{\mathbb{D}^\top}(\sigma(\mathsf{neg}(x),w)) \sqcup \mathsf{pos}(x)\\
&= \pi_{\mathbb{D}^\top}(\sigma;\tau(\mathsf{neg}(x),\mathsf{neg}(z))) \sqcup \mathsf{pos}(x),
\end{aligned}$$

and similarly for $\pi_{\mathbb{G}^\top}(\sigma;\tau(x,z))$. That is, $\sigma;\tau(x,z) = \sigma;\tau(\mathsf{neg}(x),\mathsf{neg}(z)) \sqcup \mathsf{pos}(x) \sqcup \mathsf{pos}(z)$ and therefore $\sigma;\tau$ is positive.

We now focus on the identity. We recall that as id has to be a positive closure operator, it cannot be the identity relation. We consider $\mathsf{id} : (\mathbb{F}_1^\perp)^\top \times \mathbb{F}_2^\top$. $\mathsf{id} : (x,z) \to (x \sqcup \mathsf{neg}(z), z \sqcup \mathsf{neg}(x))$. Let us start by showing that id is positive.

- $\mathsf{id}(\mathsf{neg}(x),\mathsf{neg}(y))) = (\mathsf{neg}(y) \sqcup \mathsf{neg}(x), \mathsf{neg}(x) \sqcup \mathsf{neg}(y))$. Now, one should recall that $x \in A^\perp$, that is, in $A^\perp$ we get $\mathsf{Pos}(\mathsf{neg}(y))$ and in A we get $\mathsf{Pos}(\mathsf{neg}(x))$. Therefore, as the domain is polarised coherent, and $\mathsf{neg}(x),\mathsf{neg}(y)$ are of opposite polarities, we get $\mathsf{neg}(x) \sqcup \mathsf{neg}(y) \neq \top$. Therefore, $\mathsf{id}(\mathsf{neg}(x),\mathsf{neg}(y)) \neq \top$.
- The second point is straightforward.
- $\mathsf{id}(x,y) = \mathsf{id}(\mathsf{neg}(x),\mathsf{neg}(y)) \sqcup \mathsf{pos}(x) \sqcup \mathsf{pos}(y)$ as expected.

Now we consider $\tau : (\mathbb{D}^\perp)^\top \times \mathbb{F}_1^\top$, our goal being to prove that $\tau;\mathsf{id} = \tau$. Let $(x,z) \in (\mathbb{D}^\perp)^\top \times \mathbb{F}_2^\top$. Let us compute y the witness of interaction, assuming we start the chain with id. In the case where $\tau(x,\bot) = \top$ then $\tau;\mathsf{id}(x,z) = \top$ as expected. So we deal with the case where $\tau(x,\bot) \neq \top$.

1. $y_0 = \bot$
2. $y_1 = \mathsf{neg}(z)$.
3. $y_2 = \pi_{\mathbb{F}'_1{}^{op}}(\tau(x, \mathsf{neg}(z))) = \mathsf{neg}(z) \sqcup w$, where $\mathsf{Pos}(w)$ in $\mathbb{F}_1^\top$, and therefore $\mathsf{Neg}(w)$ in $(\mathbb{F}_1^\bot)^\top$.
4. $y_3 = (\mathsf{neg}(z) \sqcup w) \sqcup \mathsf{neg}(z) = y_2$.

Therefore, the witness of interaction is $y = y_2$. We now have the following sequence of equations.

$$\begin{aligned}
\pi_{(\mathbb{D}^\top)^\bot}(\tau(x, y)) &= \pi_{\mathbb{D}^\top}(\tau(x, \mathsf{neg}(z) \sqcup w)) \\
&= \pi_{(\mathbb{D}^\top)^\bot}(\tau(x, \mathsf{neg}(z)) \sqcup w) \\
&= \pi_{(\mathbb{D}^\top)^\bot}(\tau(x, \mathsf{neg}(z))) \\
&= \pi_{(\mathbb{D}^\top)^\bot}(\tau(x, \mathsf{neg}(z)) \sqcup (\bot, \mathsf{pos}(z))) \\
&= \pi_{(\mathbb{D}^\top)^\bot}(\tau(x, z))
\end{aligned}$$

Similarly, on the right hand side we get, noticing that $w = \mathsf{neg}(y)$ in $\mathbb{F}_1^\top$.

$$\begin{aligned}
\pi_{\mathbb{F}_2^\top}(\mathsf{id}(y, z)) &= (z \sqcup \mathsf{neg}(y)) \\
&= (z \sqcup \pi_{\mathbb{F}_1^\top}(\tau(x, \mathsf{neg}(z)))) \\
&= (z \sqcup \pi_{\mathbb{F}_q^\top}(\tau(x, z))) \\
&= \pi_{\mathbb{F}_1^\top}(\tau(x, z))
\end{aligned}$$

Similarly, given $\mathsf{id} : \mathbb{D}_1 \to \mathbb{D}_2$ and $\tau : \mathbb{D}_2 \to \mathbb{F}^\top$, then, given $(x, z) \in (\mathbb{D}_1^\bot)^\top \times \mathbb{F}^\top$, we compute y the witness of interaction: $y = \pi_{\mathbb{D}_2^\top}(\tau(\mathsf{neg}(x), z))$. The proof that $\mathsf{id}; \tau(x, z) = \tau(x, z)$ follows the same lines as above. □

QualClo is almost a subcategory of the category of nominal relations, the only problem being that they do not share the same identities. Just as its parent category, it is star autonomous. More precisely, the category is compact closed.

Proposition 7.14. QualClo *is a compact closed category, with monoidal product* $\otimes$ *the cartesian product, and negation the negation of* Qual *on objects and simple symmetry on morphisms.*

That is, given $\sigma : \mathbb{D} \to \mathbb{F}$ (that is $\sigma : (\mathbb{D}^\bot)^\top \times \mathbb{F}^\top$) then $\sigma^\bot : (\mathbb{F}^\bot)^\top \times (\mathbb{D}^{\bot\bot})^\top = (\mathbb{F}^\bot)^\top \times \mathbb{D}^\top$, is defined to be $\sigma^\bot = s_{\mathbb{F}^\top, \mathbb{D}^\top} \circ \sigma \circ s_{\mathbb{D}^\top, \mathbb{F}^\top}$ (where we confound $(\mathbb{F}^\top)^\bot$ and $\mathbb{F}^\top$, and similarly for $\mathbb{D}^\top$).

Proof. We start by showing the monoidal closure. A morphism $\tau : \mathbb{D} \otimes \mathbb{F} \to \mathbb{G}$ is an equivariant P-closure operator $((\mathbb{D} \times \mathbb{F})^\bot)^\top \times \mathbb{G}^\top$. Equivalently, it is a positive closure operator $\tau : (\mathbb{D}^\bot)^\top \times ((\mathbb{F}^\bot)^\top \times \mathbb{G}^\top$ or $(\mathbb{D}^\bot)^\top \times (\mathbb{F}^\bot \times \mathbb{G})^\top$. Hence the category is monoidal closed, with $\mathbb{F} \multimap \mathbb{G} = \mathbb{F}^\bot \times \mathbb{G}$. The unit of the $\multimap$ functor is the domain $\bot_{\mathsf{Qual}} = I_{\mathsf{Qual}}$. Now, given a coherence polarised domain

$\mathbb{D}$, $\mathbb{D} \multimap \bot_{\mathsf{Qual}} \multimap \bot_{\mathsf{Qual}} = \mathbb{D} \times I_{\mathsf{Qual}} \times I_{\mathsf{Qual}} \simeq \mathbb{D}$, therefore the category is star-autonomous. Furthermore $(\mathbb{D} \otimes \mathbb{F})^{\bot} = \mathbb{D}^{\bot} \otimes \mathbb{F}^{\bot}$, therefore the star-autonomy of this category is degenerated. Thus, QualClo is compact closed. □

As closure operators compose relationally, given $\tau : \mathbb{D}^{\top} \times \mathbb{F}^{\top}$, and $\sigma : \mathbb{D}^{\top}$, we can talk about $\sigma;_{\mathbb{D}^{\top}} \tau : \mathbb{F}^{\top}$ defined by $(\sigma;\tau)^{\bullet} = \{y \in \mathbb{F}^{\top} \mid \exists(x,y) \in \tau^{\bullet}, x \in \sigma^{\bullet}\}$. Furthermore, if τ is a P-closure operator, σ is an O-closure operator, then $\sigma;_{\mathbb{D}^{\top}} \tau$ is a P-closure operator. Similarly, if $\tau : \mathbb{D}^{\top} \times \mathbb{F}^{\top}$ is a P-closure operator, $\sigma : \mathbb{F}^{\top}$ is an O-closure operator, then $\tau;_{\mathbb{D}^{\top}} \sigma$ is a P-closure operator. The proof is done seeing σ as a P-closure operator $\mathbb{F} \to I$, and $\tau : \mathbb{D} \to \mathbb{F}$.

When restricting the category QualCLo to its full sub-category that consists of objects that are denotations of formulas of MLL^{-} (that we still denotes QualClo for simplicity), there is a forgetful functor from the category QualClo to the category of lax nominal relations NomReL. We recall that NomReL has objects nominal sets and morphisms nominal relations. We deemed them *lax* since we do not impose the "separated" condition. This one consists in projecting on maximal elements. We name it Max. We say that an element of a domain $d \in \mathbb{D}$ is **maximal** if $\neg(\exists e \in \mathbb{D}.d < e)$. Given an element d of $\mathbb{D}^{\top}$, we write $\mathsf{max}(d)$ if $d \neq \top$ and d is maximal in $\mathbb{D}$.

- $\mathsf{Max}(\mathbb{D}) = \{d \in \mathbb{D} \mid \mathsf{max}(d)\}$
- Given $\sigma : (\mathbb{D} \times \mathbb{F})^{\top}$, $\mathsf{Max}(\sigma) = \{(x,y) \in \sigma^{\bullet} \mid \mathsf{max}(x,y)\}$

Furthermore, this functor can be refined by taking into account the polarity. Its image category becomes the category of polarised lists and lax nominal relations[1]. Indeed, as the monoidal product is cartesian the separated condition is dropped. Similarly, given σ a P-closure operator, and $\mathcal{R} = \mathsf{Max}(\sigma)$, then nothing forces the relation $\mathcal{R}$ to be linear, that is, given $x \in \mathcal{R}$ that $\nu(\mathsf{neg}(x)) = \nu(\mathsf{pos}(x))$, neither even $\nu(\mathsf{pos}(x)) \subseteq \nu(\mathsf{neg}(x))$. However, if the x comes from an interaction with a counter strategy, that is, there exists a O-closure operator τ such that $x = \langle\sigma,\tau\rangle$, then $\nu(\mathsf{pos}(x)) \subseteq \nu(\mathsf{neg}(x))$ since σ is equivariant, and hence, unable to raise a name by itself ($\nu(\sigma(x)) = \nu(x)$) and τ can only raise names whose corresponding primes are of negative polarity (since it is a O-strategy). Note that Max is a functor of compact closed categories, since $\mathsf{Max}(\sigma \times \tau) = \mathsf{Max}(\sigma) \times \mathsf{Max}(\tau)$, and $\mathsf{Max}(\sigma^{\bot}) = \mathsf{Max}(\sigma)^{\bot}$.

The concurrent games used in [10] were based on a double glueing construction. The objects of the category were triples $(\mathbb{D}, S_{\mathbb{D}}, S_{\mathbb{D}}^{\bot})$, where $\mathbb{D}$ was a prime algebraic domain, S a set of strategies on $\mathbb{D}^{\top}$, and $S^{\bot}$ a set of counter strategies on it. Furthermore the morphisms were total, in the sense that given a morphism $\sigma : I \to \mathbb{D}$ then $\forall\tau \in S_{\mathbb{D}}^{\bot}$, $\langle\sigma,\tau\rangle$ reaches a maximal position. Finally to achieve full completeness, one needed to consider dinatural transformations over this category. Here, we keep working with the morphisms of the first-order category and this condition is replaced by a "linearity" condition on names used by the morphisms.

Definition 7.15. *The category* GQualClo *has as objects triples* $(\mathbb{D}, S_{\mathbb{D}}, S_{\mathbb{D}}^{\bot})$ *where* $\mathbb{D}$ *is a nominal polarised coherence domain,* $S_{\mathbb{D}}$ *(respectively* $S_{\mathbb{D}}^{\bot}$*) nominal sets of P-closure (respectively O-closure) operators with finite supports on* $\mathbb{D}$*. A morphism* $(\mathbb{D}, S_{\mathbb{D}}, S_{\mathbb{D}}^{\bot}) \to (\mathbb{F}, U_{\mathbb{F}}, U_{\mathbb{F}}^{\bot})$ *is a mor-*

[1]This category is just like the category of lax nominal polarised relations LaxNomLinPol defined in 3.4.5, except that the linearity condition is dropped.

phism of QualClo*,* $\sigma : (\mathbb{D}^{\perp})^{\top} \times \mathbb{F}^{\top}$ *such that* $\forall \tau \in S_{\mathbb{D}}.\tau;_{(\mathbb{D}^{\perp})^{\top}} \sigma \in U_{\mathbb{F}}$ *and* $\forall \tau \in U_{\mathbb{F}}^{\perp}.\sigma;_{\mathbb{F}^{\top}} \tau \in S_{\mathbb{D}}^{\perp}$*.* *Furthermore, it must obey the following conditions:*

- *(totality): For each* $\tau \in S_{\mathbb{D}} \times U_{\mathbb{F}}^{\perp}$*,* $\langle \sigma, \tau \rangle \neq \top$ *is a maximal element of* $\mathbb{D}^{\perp} \times \mathbb{F}$*.*
- *(linearity): There exists a relation* $\mathcal{R} \in$ NomLinRelPol *such that* $\mathsf{Max}(\sigma) = \widehat{\mathcal{R}}$ *.*

Notably, the empty support condition prevents σ from introducing names: $\forall x \in \mathbb{D}.\nu(\sigma(x)) = \nu(x)$. The linearity condition prevents the strategy to react differently if opponent happens to bring two equal names.

We will prove that GQualClo is star autonomous, with monoidal tensor:

$$(\mathbb{D}, S_{\mathbb{D}}, S_{\mathbb{D}}^{\perp}) \otimes (\mathbb{F}, U_{\mathbb{F}}, U_{\mathbb{F}}^{\perp}) = (\mathbb{D} \times \mathbb{F}, S_{\mathbb{D}} \times U_{\mathbb{F}}, Z)$$
$$Z = \{\tau : (\mathbb{D} \times \mathbb{F})^{\top} \mid \tau \text{ is an O-closure operator and}$$
$$(\forall \sigma \in S_{\mathbb{D}}, \sigma;_{\mathbb{D}} \tau \in U_{\mathbb{F}}^{\perp} \wedge \forall \sigma \in U_{\mathbb{F}}.\tau;_{\mathbb{F}} \sigma \in S_{\mathbb{D}}^{\perp})\}$$

and negation:

$$(\mathbb{D}, S_{\mathbb{D}}, S_{\mathbb{D}}^{\perp})^{\perp} = (\mathbb{D}^{\perp}, S_{\mathbb{D}}^{\perp}, S_{\mathbb{D}})$$

where in the right hand side of the equation, the closure operators of $S_{\mathbb{D}}^{\perp}, S_{\mathbb{D}}$ are seen as closure operators of $\mathbb{D}^{\perp}$.

Proposition 7.16. GQualClo *is a star-autonomous category.*

Proof. We start by briefly showing that it indeed forms a category. We focus on totality and linearity, since the double-glueing conditions are coming from the literature, where they have already been studied. Identity is obviously total, and $\mathsf{Max}(\mathsf{id}_{\mathsf{GQualClo}}) = \widehat{\mathsf{id}_{\mathsf{NomLinRelPol}}}$. Given two morphisms $\sigma : \mathbb{D} \to \mathbb{F}$ and $\tau : \mathbb{F} \to \mathcal{G}$ then $\sigma;\tau$ remains total since τ acts like a counter-strategy to σ on $\mathbb{F}$, and respectively for σ and τ. Furthermore, the composite remains linear since Max acts as a functor.

Secondly, since Max acts as a functor of compact-closed category, the linearity conditions remains stable under tensor and negation. It is quite straightforward to prove that this is equally the case for the totality one. Therefore, GQualClo forms a star-autonomous category. □

Finally, we describe the denotation function from formulas of MLL to the category GQualClo.

- $[\![X]\!] = ([\![X]\!]_{\mathsf{Qual}}, S_X = \{\sigma \mid \sigma(\perp) = a, a \in \mathbb{A}_X\}, S_X^{\perp} = \{\mathsf{id}\})$
- $[\![I]\!] = ([\![I]\!]_{\mathsf{Qual}}, S_I = S_I^{\perp} = \{\mathsf{id}\})$

And, as usual, we restrict the objects of our category to those that are freely generated from $[\![X]\!]$, $[\![I]\!]$ and the operations $(.)^{\perp}, \otimes$. GQualClo is star-autonomous. The double glueing condition prevents the category from being degenerated, that is, it is not a compact closed category.

Unfortunately, these morphisms are too lax to be potential representations of MLL proofs: two morphisms can be different though establishing the same axiom links, preventing the category for being a candidate for full completeness. Therefore, we establish an equivalence relation between strategies. Two strategies $\sigma, \tau : \mathbb{D} \to \mathbb{F}$ are equivalent, written $\tau \sim \sigma$ if $\mathsf{Max}(\sigma) = \mathsf{Max}(\tau)$. This equivalence is compatible with the star-autonomy structure. That is, if $\sigma_1 \sim \sigma_2$ and $\tau_1 \sim \tau_2$ then:

- $\sigma_1; \tau_1 \sim \tau_1; \tau_2$ (if the types of σ, τ are compatible).
- $\sigma_1 \otimes \sigma_2 \sim \tau_1 \otimes \tau_2$.
- $\sigma_1^\perp \sim \tau_1^\perp$.

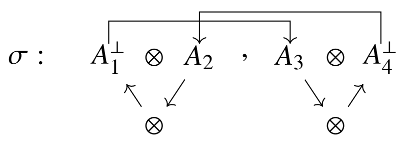
All these equivalences relies on Max being a functor of star-autonomous categories.

Therefore, we can refine the category so that morphisms are equivalence classes of morphisms. This remains a star-autonomous category, and this way, we can define the denotation of a proof π as being the equivalence class of P-closure operators such that $\mathsf{Max}(\sigma) = \widehat{[\![\pi]\!]_{\mathsf{NomLinRel}}}$. We denote by ExtGQualClo this category. Of course, we get the following proposition whose proof remains solely on the preservation of equivalences displayed above.

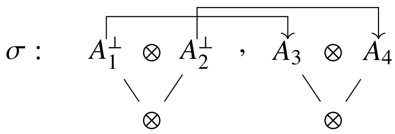
Proposition 7.17. ExtGQualClo *is a star-autonomous category.*

7.1.4 Failure of full completeness

The morphisms of ExtGQualClo lead to properly defined MLL^- proof structures. That is, they define a unique set of axiom links that encompasses all literals, by definition of the linearity condition we impose on them. Furthermore, if we orient the axiom links from $A^\perp$ to A, then the conditions imposed by ExtGQualClo exclude proof structures with directed cycles. More precisely, each morphism of ExtGQualClo is the denotation of a proof structure without directed cycles. This has been studied in [10, 74, 70]. For instance, consider the following proof structure, together with its unique oriented cycle:



We furthermore consider a counter-strategy τ that acts as follows. τ waits for A_2 to be maximal before playing in A_1. Similarly, it will wait for A_3 to be maximal before playing in A_4. On the other hand, a strategy σ that implements this proof structure needs the name in A_1 or A_4 before being able to move himself. Hence, we reach a deadlock $\langle \sigma, \tau \rangle = \perp$, and σ is not total. Therefore, there are no strategies σ of GQualClo such that $\mathsf{Max}(\sigma)$ is the nominal relation corresponding to this proof structure. This is linked to the fact that the cycle is oriented. On the other hand, if we consider this new proof structure, whose cycle cannot be given an orientation:



then a counter strategy will always start by bringing the names in A_1, A_2, since it is of the form $\tau_{A_1^\perp \otimes A_2^\perp} \times \tau_{A_3 \otimes A_4}$. Therefore, a P-closure operator can simply wait for a counter strategy to bring those names, and answer in A_3, A_4. Hence there is a P-closure operator associated with this proof structure that is total, linear, and therefore a morphism of GQualClo.

Somehow, the opponent "is forced" to move simultaneously on A_1 and A_2, whereas they are, from the opponent point of view, two different threads. Therefore, we would like to attach some kind of information to the positions, that would make it clear that a position (a_1, a_2) in $A_1^\perp \otimes A_2^\perp$ corresponds to two threads, whereas its player answer (a_3, a_4) in $A_3 \otimes A_4$ corresponds to a unique one, leading to a mismatch.

7.2 Partial nominal relations and Chu-conditions

7.2.1 Partial nominal relations

To start, we have to gain finer control on the way the strategy acts. This will also allow us to forget about the quotient.

To achieve that, we focus on a structure possibly underlying σ, an **output function** f, that is a monotone nominal function $f : \mathbb{D} \to \mathbb{D}$ such that $\sigma(x) = x \sqcup f(x)$. Working with output functions allows us to achieve finer control on our strategies. Therefore, from now on, we forget about σ, the closure operator, and focus on f, while keeping in mind that the function $x \sqcup f(x)$ should form a morphism of GQualClo.

First, we expect this function to be equivariant. Just as in [10], we require that this output function is **stable**, that is, such that $\forall x, y \in \mathbb{D}, x \uparrow y \Rightarrow f(x \sqcap y) = f(x) \sqcap f(y)$. We want f to satisfy this condition, and we furthermore require f to be **additive**. That is :

$$x \uparrow y \Rightarrow f(x \sqcup y) = f(x) \sqcup f(y)$$

A monotone function that is both additive and stable is said to be **linear**. Since f is stable, it is entirely defined by its trace:

$$\mathsf{tr}(f) = \{(x, \mathsf{p}) \mid \mathsf{p} \in \mathrm{Pr}(\mathbb{D}), x \in \mathbb{D} \text{ minimal s.t } \mathsf{p} \sqsubseteq f(x)\}.$$

The reverse operation being:

$$f(y) = \bigsqcup \{\mathsf{p} \mid (y, \mathsf{p}) \in \mathsf{tr}(f), y \leq x\}$$

Furthermore, since it is linear, $\forall(x, \mathsf{p}) \in \mathsf{tr}(f)$, x is a prime. We enforce f to be **polarised**, that is, for each element of its trace, its left hand side is an O-prime, and its right hand side a P-prime. This notably entails this simple lemma.

Lemma 7.18.
- $\forall x \in \mathbb{D}. f(x) = f(\mathsf{neg}(x))$.
- $\mathsf{Pos}(f(x))$
- $\mathsf{Neg}(x) \Rightarrow x \sqcup f(x) \neq \top$

Proof. We start with the first property.

$$\begin{aligned} f(x) &= \bigsqcup\{\mathsf{p} \mid (\mathsf{p}', \mathsf{p}) \in \mathsf{tr}(f), \mathsf{p}' \leq x\} \\ &= \bigsqcup\{\mathsf{p} \mid (\mathsf{p}', \mathsf{p}) \in \mathsf{tr}(f), \mathsf{p}' \leq \mathsf{neg}(x)\} \\ &= f(\mathsf{neg}(x)) \end{aligned}$$

Where we used on the following property: $\mathsf{p}' \leq x \wedge \mathsf{neg}(\mathsf{p}') \Leftrightarrow \mathsf{p}' \leq \mathsf{neg}(x)$, on which we can rely since, as, since f is polarised, $\mathsf{neg}(\mathsf{p}')$.

The second property is straightforward and the third follows from our domains being polarised coherent, and, therefore, negative and positive elements are always compatible with one another. □

Furthermore, since we require that the maximal positions project linearly to denotations of proofs, we also impose the two following properties, called **separation conditions**.
- $\forall(\mathsf{p}_1, \mathsf{p}_1'), (\mathsf{p}_2, \mathsf{p}_2') \in \mathsf{tr}(f). \mathsf{p}_1' \neq \mathsf{p}_2' \Rightarrow \mathsf{p}_1 \neq \mathsf{p}_2$ that is, $\mathsf{tr}(f)$ is an injective partial function from negative to positive primes.
- $\forall \mathsf{p} \in \Pr(\mathbb{D}). \mathsf{Neg}(\mathsf{p}) \Rightarrow \exists \mathsf{p}'.(\mathsf{p}, \mathsf{p}') \in \mathsf{tr}(f)$, that is, this function is total.
- $\forall \mathsf{p} \in \Pr(\mathbb{D}). \mathsf{Pos}(\mathsf{p}) \Rightarrow \exists \mathsf{p}'.(\mathsf{p}', \mathsf{p}) \in \mathsf{tr}(f)$, that is, this function is surjective.

This basically states that the trace, and therefore f establishes a nominal bijection between the negative and positive primes.

Proposition 7.19. *Let π be a nominal bijection between negative and positive primes of a nominal coherent domain. Then f, defined as follows:*

$$f(x) = \bigsqcup\{\mathsf{p} \mid \pi^{-1}(\mathsf{p}) \sqsubseteq x\}$$

is a linear monotone function, whose trace is indeed $\{(\mathsf{p}, \pi(\mathsf{p})\}$.

The proof is immediate. It entails that the separation conditions are indeed compatible with the definition of trace.

We call **positions of interactions** the positions that can be reached through an interaction against a supposedly given opponent. That is, given an output function f, we write $f^\bullet$ for **the set of positions of interactions**, that is defined to be the closure under finite compatible union of the following set:

$$\{a \sqcup f(a) \mid (a, f(a)) \in \mathsf{tr}(f)\}$$

As we restrict to finite unions, each element of $f^\bullet$ has finite support, and $f^\bullet$ is a nominal set. Furthermore, as we only consider compatible union $\top \notin f^\bullet$ and $\bot = \bigsqcup \emptyset \in f^\bullet$. Finally, $\forall x \in f^\bullet . x \sqcup f(x) = x$. Now given a pair (σ, f) such that $\sigma(x) = x \sqcup f(x)$, we straightforwardly have $f^\bullet \subseteq \sigma^\bullet$. That is, $f^\bullet$ selects those positions of $\sigma^\bullet$ that are relevant.

The output functions compose through the Kahn semantics of dataflow [55, 6]. Given $f : A \times B$ and $g : B \times C$, then $f; g(a, c) = (a', c') \in A \times C$ where (a', c') is defined to be the least solution of the following equations:

- $f(a, b) = (a', b')$
- $g(b', c) = (b, c')$

Note that b' can be computed as the limit of the growing chain $\bigsqcup_{i \in \mathbb{N}} ((\pi_B; f(a, \_)) \circ (\pi_B; g(\_, c)))^i(\bot)$. As the domain only admits finite chains, f, g are automatically continuous, and hence b, b' always exists and are well defined. Just as their closure operator counterparts, we will prove $(f; g)^\bullet = f^\bullet ;_{\mathsf{Rel}} g^\bullet$. These functions form a category, called category of partial nominal relations.

Definition 7.20. *A **Partial Nominal Relation** on $\mathbb{D}$, where $\mathbb{D}$ is a polarised coherent domain, is a nominal relation $\mathcal{R} : \mathbb{D}$ such that there exists $f : \mathbb{D} \to \mathbb{D}$ nominal, linear, polarised function subject to the separation conditions (f establishes a bijection between negative and positive primes), satisfying $\mathcal{R} = f^\bullet$.*
The category ParNomRel *has objects nominal polarised qualitative domains and morphisms $A \to B$ partial nominal relations in $A^\perp \times B$. The identity is the identity relation and the morphisms compose as relations.*

Proposition 7.21.
- *$f^\bullet$ forms a lattice.*
- $\forall x \in f^\bullet, \nu(\mathsf{neg}(x)) = \nu(\mathsf{pos}(x))$.
- *Working with polarised coherent domains $\mathbb{D}$ that arise as denotations of formulas of* MLL$^-$*, f preserves the compatibility: $\forall x, y \in \mathbb{D}. x \uparrow y \Rightarrow f(x) \uparrow f(y)$.*
- ParNomRel *forms a category, and $(f; g)^\bullet = f^\bullet ;_{\mathsf{Rel}} g^\bullet$.*

Proof. To show that $f^\bullet$ forms a lattice, we have to prove it is stable under finite union and intersection. The stability under finite union follows from its definition. Now, given a finite family of x_i in $f^\bullet$, $x_i = \bigsqcup_j (\mathsf{p}_j \sqcup f(\mathsf{p}_j))$, then $\bigsqcap x_i = \bigsqcup \{\mathsf{p}_k \sqcup f(\mathsf{p}_k) \mid \forall i. \mathsf{p}_k \leq x_i\}$, (this follows from a simple computation using that $\mathsf{tr}(f)$ is a bijection between negative and positive primes), and hence $\bigsqcap x_i \in f^\bullet$.

The second point is immediate, just as the third, noticing that in domains two elements are conflicting if they have two equivalent primes in it.

We now start to prove that linear functions compose, and that the additional criteria are respected.

Let us consider two functions $f : A \times B$ and $g : B \times C$, together with $(a, c) \in A \times C$. We assume that $f; g(a, c) = (a', c')$, that is (a', c') is the minimal pair solution of the equations:

- $f(a, b) = (a', b')$
- $g(b', c) = (b, c')$.

In that case, we say that $(a, b), (b', c)$ satisfy the equational system for the composition.

To tackle the proof, we will use an additional property of our functions, namely their stability under the complement operator , denoted \. Given x, y two elements of the qualitative domain, seen as sets of primes, we write $x \setminus y$ for the element $\bigsqcup\{\mathsf{p} \mid \mathsf{p} \in \mathrm{Pr}(\mathbb{D}), \mathsf{p} \sqsubseteq x \wedge \mathsf{p} \not\sqsubseteq y\}$. Then our functions satisfy the following property:

$$f(x \setminus y) = f(x) \setminus f(y).$$

This is straightforward to prove, seeing $\mathsf{tr}(f)$ as a bijection between negative and positive primes.

The composite is stable: Let us compare $f; g(x \sqcap y)$ versus $f; g(x) \sqcap f; g(y)$. Let $x = (x_A, x_C)$, and $y = (y_A, y_C)$ be compatible. Let x_B^1, x_B^2 the minimal elements of interaction, that is, such that $f(x_A, x_B^1) = (x'_A, x_B^2)$ and $g(x_B^2, x_C) = (x_B^1, x'_C)$. Similarly, we introduce y_B^1, y_B^2. As f, g preserve the compatibility, seeing x_B^1, y_B^1 as upper lowest bound, we got $x_B^1 \uparrow y_B^1$, and respectively for x_B^2, y_B^2. As f, g are stable, we have:

$$f(x_A \sqcap y_A, x_B^1 \sqcap y_B^1) = (x'_A \sqcap y'_A, x_B^2 \sqcap y_B^2)$$
$$g(x_B^2 \sqcap y_B^2, x_C \sqcap y_C) = (x_B^1 \sqcap x_B^2, x'_C \sqcap y'_C).$$

Therefore, $(x_A \sqcap y_A, x_B^1 \sqcap y_B^1)$, $(x_B^2 \sqcap y_B^2, x_C \sqcap y_C)$ satisfy the equational system. However, they might not be minimal. Therefore, we already know that:

$$f; g(x_A \sqcap y_A, x_C \sqcap y_C) \sqsubseteq f; g(x_A, x_C) \sqcap f; g(y_A, y_C).$$

Let suppose there is some lesser v^1, v^2 such that $f(x_A \sqcap y_A, v_B) = (w_A, v^2)$ and $g(v^2, x_C \sqcap y_C) = (v^1, w_C)$ and $v^1 \sqsubseteq x_B^1 \sqcap y_B^1$, $v^2 \sqsubseteq x_B^2 \sqcap y_B^2$ (where at least one of the inequalities is strict). We write z^1 for $x_B^1 \sqcap y_B^1$, and z^2 for $x_B^2 \sqcap y_B^2$. The following equations hold:

$$f(\bot, z^1 \setminus v^1) = f(x_A \sqcap y_A, z_1) \setminus f(x_A \sqcap y_A, v_1) = (z'_A, z^2 \setminus v^2)$$

and

$$f(z^2 \setminus v^2, \bot) = f(z^2, x_C \sqcap y_C) \setminus f(v^2, x_C \sqcap y_C) = (z^1 \setminus v^1, z''_C).$$

Therefore, $(\bot, z^1 \setminus v^1)$ and $(z^2 \setminus v^2, \bot)$ satisfy the equational system. But therefore, so does $(x_A, x_B^1 \setminus (z^1 \setminus v^1))$ and $(x_B^2 \setminus (z^2 \setminus v^2), x_C)$. Or, x_B^1 and x_B^2 are minimal. This implies $x_1^B \sqcap (z^1 \setminus v^1) = \emptyset$ (and similarly for x_B^2, z^2, v^2). Or, as $z^1 = x_B^1 \sqcap y_B^1$, this entails that $z^1 \sqsubseteq v^1$. Or v^1 was chosen to be minimal so $z^1 = v^1$. So $x_B^1 \setminus (z^1 \setminus v^1) = x_B^1 \Rightarrow z^1 = v^1$. A similar pattern leads us to conclude that $z^2 = v^2$. So, overall, z^1, z^2 are minimal. As a result, this entails:

$$f; g(x_A^1 \sqcap x_B^2, x_C^1 \sqcap x_C^2) = f; g(x_A^1, x_C^1) \sqcap f; g(x_A^2, x_C^2),$$

that is, the composite function is stable.

The composite is additive: Composition of additivity is proven as follows. Given (x_A, x_B^1), (x_B^2, x_C) and (y_A, y_B^1), (x_B^2, y_C) satisfying the equational system, we get $(x_A \sqcup y_A, x_B^1 \sqcup y_B^1)$, $(x_B^2 \sqcup y_B^2, x_C \sqcup y_C)$ satisfying it as well, following the additivity of f. This proves that $f; g(x_A \sqcup y_A, x_C \sqcup y_C) \sqsubseteq f; g(x_A, x_C) \sqcup f; g(y_A, y_C)$. Now as $f; g$ is monotone, then $f; g(x \sqcup y) \sqsupseteq f; g(x) \sqcup f; g(y)$ and by combining the two inequalities we get the result.

The separation conditions compose: We prove the proposition by analysing how $f; g$ acts on a negative prime. Suppose $\mathsf{p} \in A$ and $\mathsf{Neg}(\mathsf{p})$. Then $f(\mathsf{p}, \bot)$ will trigger exactly one positive prime p'. Either this prime is in A, in which case $f; g(\mathsf{p}) = \mathsf{p}'$, or this prime is in B. Now, this prime p', that is negative from the g point of view, will trigger, through g, a unique positive prime p''. This chain of p continues through f, g, playing new primes on B. As there is only a finite number of primes (compatible with each other) in B the process ends, that is, either f finishes the chain by playing in A, or g finishes it by playing in C. Therefore, each negative prime triggers one and only one positive prime through $f; g$. Furthermore, this one is unique. Finally, by backtracking, one can see that every positive prime is triggered through $f; g$ by a single negative prime.

Polarised function do compose: We need to prove that for each couple $(\mathsf{p}, \mathsf{p}') \in \mathsf{tr}(f; g)$ then $\mathsf{Neg}(\mathsf{p}), \mathsf{Pos}(\mathsf{p}')$. This is straightforward.

Equivalence sequential -relational composition: Let $(x_A, x_C) \in (f; g)^\bullet$. This implies that there exist y_B^1, y_B^2, such that $f(\mathsf{neg}(x_A), y_B^1) = (\mathsf{pos}(x_A), y_B^2)$, $g(y_B^2, \mathsf{neg}(x_C)) = (y_B^1, \mathsf{pos}(x_C))$. Furthermore, as f, g is polarised then $\mathsf{Neg}(y_B^1)$ and $\mathsf{Pos}(y_B^2)$. Therefore, $(x_A, y_B^1 \sqcup y_B^2)$ in $f^\bullet$, $(y_B^2 \sqcup y_B^1, x_C) \in g^\bullet$. Thus, $(x_A, x_C) \in f^\bullet ;_{\mathsf{Rel}} g^\bullet$, and $(f; g)^\bullet \subseteq f^\bullet ;_{\mathsf{Rel}} g^\bullet$. Now, let us consider $(x_A, x_B) \in f^\bullet$, $(x_B, x_C) \in g^\bullet$. Then, let us consider the set of those x_i such that $(x_A, x_i) \in f^\bullet$, $(x_i, x_C) \in g^\bullet$. This set is closed under compatible intersection. Therefore, we can take a least element. We rename this element x_B. Now, $f(\mathsf{neg}(x_A), \mathsf{neg}(x_B)) = (\mathsf{pos}(x_A), \mathsf{pos}(x_B))$ and $g(\mathsf{pos}(x_B), \mathsf{neg}(x_C)) = (\mathsf{neg}(x_B), \mathsf{pos}(x_C))$. Furthermore, $\mathsf{neg}(x_B), \mathsf{pos}(x_B)$ are minimal satisfying this set of equations. Hence $(x_A, x_C) \in (f; g)^\bullet$, and $(f; g)^\bullet = f^\bullet; g^\bullet$. □

Note that, as our the traces of our functions act as bijections, the composition through de Kahn semantics of data-flow is equivalent to the composition of permutations trough tracing, as defined in 3.3.

Proposition 7.22. ParNomRel *is star-autonomous, the monoidal product being the cartesian product, and the negation consists in reversing polarities on objects, together with taking $f^\perp$ as* *being the monotone linear function whose trace arise from the bijection* $\mathsf{tr}(f)^{-1}$ *on morphisms. Finally, the unit of the monoidal product is the qualitative domain with no primes I.*

There is a functor $F : \mathsf{ParNomRel} \to \mathsf{QualClo}$, and this functor is a functor of star-autonomous categories. It lets the objects invariant, and maps a function f to $\sigma = \mathsf{id} \sqcup f$.

7.2.2 Chu conditions

Working with partial nominal relations allows us to have a greater control on the relations. Now, we wish to gain control of a notion of O and P positions. In order to do that, we took inspiration from the simplest instance of the Chu-construction, that we remind below.

Proposition 7.23. *Let C be a symmetric monoidal closed category with products (written $*$ here). Then $C^{\mathsf{d}} = C \times C^{\mathsf{op}}$ is a star-autonomous category. The tensor product is :*

$$(U, X) \otimes (V, Y) = (U \otimes V, U \multimap Y * V \multimap X)$$

The unit for the tensor is $(I, 1)$, and the negation $(U, X)^{\perp} = (X, U)$.

Hyland and Schalk [51] used this construction in order to establish a functor from the category of tree-games into the star-autonomous category $\mathsf{Rel}^{\mathsf{d}}$. The functor sent an arena to its sets of P and O-positions respectively. That way, one saw an object of $\mathsf{Rel}^{\mathsf{d}}$ as a pair (P_A, O_A), composed of sets of P and O positions respectively. Finally, a strategy $\sigma : A \to B$ of the standard category of tree-games was sent to a pair of relations $(\sigma^+ : P_A \to P_B, \sigma^- : O_B \to O_A)$. This functor effectively detemporised the games.

We will apply the same idea here for our domains. The starting point is to consider that the objects of our category are now 3-tuples $A = (\mathbb{D}_A, P_A, O_A)$, where $\mathbb{D}$ is an object of $\mathsf{ParNomRel}$, or, equivalently $\mathsf{QualClo}$, and $P_A, O_A \subseteq \mathbb{D}_A$ are the sets of P and O-positions. Note that we expect $P_A \cap O_A = \emptyset$. However, it might not be the case that $P_A \uplus O_A = \mathbb{D}_A$ (where we see $\mathbb{D}_A$ as its underlying set). Some positions of the domain might belong to neither player nor opponent. The tensor and negation are defined according to the Chu-construction.

$$A \otimes B = (\mathbb{D}_A \otimes \mathbb{D}_B, P_{A \otimes B} = P_A \times P_B, O_{A \otimes B} = O_A \times P_B \uplus P_A \times O_B)$$
$$A^{\perp} = (\mathbb{D}_A^{\perp}, O_A, P_A).$$

where we remind that the tensor product of objects of Qual is defined as the cartesian product of the underlying domain, together with the coproduct of their polarity functions.

Finally, we require that our morphisms somehow respect the Chu-condition as well. Given a morphism f of $\mathsf{ParNomRel}$, $f : A \to B$, we expect that $f^{\bullet}$ establishes two relations :

$$f^\bullet : P_A \to P_B$$
$$f^\bullet : O_B \to O_A.$$

That is, written $Q_A = P_A \uplus O_A$ and similarly for Q_B, we expect that $(P_A; f^\bullet) \cap Q_B \subseteq P_B$, and $(f^\bullet; O_B) \cap Q_A \subseteq O_A$, where we remind that given a relation $\mathcal{R} : A \times B$ and a relation $\mathcal{Q} : A$, we write $\mathcal{Q} ;_A \mathcal{R}$ for the set $\{y \in B \mid \exists (x, y) \in \mathcal{R}, x \in \mathcal{Q}\}$. Furthermore, we want to make sure that these two relations are not empty. To do that, we define this simple variant of the double-glueing construction.

We introduce two new sets $T_{\mathbb{D}} \subseteq \mathcal{P}_{\mathsf{finsup}}(P_{\mathbb{D}})$ and $T_{\mathbb{D}}^{\perp} \subseteq \mathcal{P}_{\mathsf{finsup}}(O_{\mathbb{D}})$, where $\mathcal{P}_{\mathsf{finsup}}$ denotes the operator that to a set gives its subsets of finite support. The objects of our new category $\mathsf{ChuLinNom}$ will be 5-tuples $(\mathbb{D}, P_{\mathbb{D}}, O_{\mathbb{D}}, T_{\mathbb{D}}, T_{\mathbb{D}}^{\perp})$. $O_{\mathbb{D}}$ and $P_{\mathbb{D}}$ will be the sets of O and P positions respectively, and $T_{\mathbb{D}}$ and $T_{\mathbb{D}}^{\perp}$ the sets of P and O relations respecting the Chu and double glueing conditions.

There is a monoidal tensor on $\mathsf{ChuLinNom}$ objects defined through the usual double glueing structure. Given a relation $\mathcal{R} : A \times B$, we denote $\mathcal{R} \upharpoonright B$ the set $\{y \in B \mid \exists (x, y) \in \mathcal{R}\}$. The monoidal product is defined as follows:

$$\begin{aligned}
A \otimes B &= \{\mathbb{D}_A \times \mathbb{D}_B, P_A \times P_B, O_A \times P_B \uplus P_A \times O_B, T_{A\otimes B}, T_{A\otimes B}^{\perp}\} \\
T_{A\otimes B} &= \{\mathcal{R} \subseteq \mathcal{P}_{\mathsf{finsup}}(P_{A\otimes B}) \mid \mathcal{R} = \mathcal{R}_A \times \mathcal{R}_B \wedge \mathcal{R}_A \in T_A \wedge \mathcal{R}_B \in T_B\} = T_A \times T_B \\
T_{A\otimes B}^{\perp} &= \{\mathcal{R} \in \mathcal{P}_{\mathsf{finsup}}(O_{A\otimes B}) \mid \\
&\qquad \forall \mathcal{Q} \in T_A . \mathcal{Q} ;_A \mathcal{R} \in T_B^{\perp} \wedge \forall \mathcal{Q} \in T_B . \mathcal{R} ;_B \mathcal{Q} \in T_A^{\perp}\}
\end{aligned}$$

The unit I is defined by:

$$I_{\mathsf{ChuLinNom}} = (\llbracket I \rrbracket_{\mathsf{Qual}}, P_I = \{\perp\}, O_I = \emptyset, T_I = \{\{\perp\}\}, T_I^{\perp} = \emptyset)$$

Similarly, there is a negation defined by:

$$(\mathbb{D}, P_{\mathbb{D}}, O_{\mathbb{D}}, T_{\mathbb{D}}, T_{\mathbb{D}}^{\perp})^{\perp} = (\mathbb{D}^{\perp}, O_{\mathbb{D}}, P_{\mathbb{D}}, T_{\mathbb{D}}^{\perp}, T_{\mathbb{D}})$$

where we see $O_{\mathbb{D}}$ as a subset of $\mathbb{D}^{\perp}$, and similarly for the other elements of the negated 5-tuple. The objects of $\mathsf{ChuLinNom}$ are built freely out of the two following objects by tensor and negation:

$$\begin{aligned}
\llbracket X \rrbracket &= (\llbracket X \rrbracket_{\mathsf{Qual}}, P_X = \mathbb{A}_X \subseteq \llbracket X \rrbracket_{\mathsf{Qual}}, O_X = \{\perp\}, \\
&\qquad T_X = \{x \in \mathcal{P}_{\mathsf{finsup}}(P_X), x \neq \emptyset\} \quad T_X^{\perp} = \{\{\perp\}\}) \\
\llbracket I \rrbracket &= I_{\mathsf{ChuLinNom}}
\end{aligned}$$

It is important to note that the concept of O-positions is unrelated to the one of O-primes, or O-strategies (as defined in the section 7.1.3 about closure operators) For instance, $(a, \perp)$ is an O-position of $A \otimes A$, as a is a P-position and $\perp$ an O one. Hence, by raising a P-prime (the name a), the strategy would have actually created an O-position. This might be explained by the fact that a strategy is supposed to play simultaneously in a tensor. Hence, the real position a P-strategy should reach is (a, a'), which is indeed a P-position. Note that there are functions f of ParNomRel that are denotations of valid proof structures, and such that there exists $x \in f^\bullet$ with $x \notin P_{\mathbb{D}}$. Indeed, some positions of $f^\bullet$ might lie outside the set of positions that are given a polarity. For instance, consider a maximal position (a, a, a, a) of $(A^\perp \otimes A^\perp) \parr A \parr A$, that is a valid formula of MLL.

We would like our morphisms $A \to B$ to be maps $f \in \mathsf{ParNomRel}(A, B)$ such that $f^\bullet \cap Q_{A \multimap B} \in T_{A \multimap B}$, where we remind that $Q_{A \multimap B} = P_{A \multimap B} \uplus O_{A \multimap B}$, and that $A \multimap B$ is the monoidal closure, defined as $A \multimap B = (A \otimes B^\perp)^\perp$. However, at this stage, it turns out we have not been able to prove composition with such morphisms. We will explain why in the next paragraph, and expose a slightly weaker criterion.

Let us consider two morphisms of ParNomRel as follows $f : A \multimap B$ and $g : B \multimap C$ that satisfy the desired criterions, that is, $f^\bullet \cap Q_{A \multimap B} \in T_{A \multimap B}$ and similarly for g. Then, in order to establish composition, we must prove that $(f^\bullet; g^\bullet) \cap Q_{A \multimap C} \in T_{A \multimap C}$. By definition of the double glueing construction, $(f^\bullet \cap Q_{A \multimap B}) ;_{\mathsf{Rel}} (g^\bullet \cap Q_{B \multimap C}) \in T_{A \multimap C}$. The difficulty comes from the fact that there might be some b such that $(a, c) \in Q_{A \multimap C}$, $(a, b) \in f^\bullet$, $(b, c) \in g^\bullet$ but $b \notin Q_B$. We have not found an example where such a b exists, however we have not been able to rule it out either. In other terms, if such a b exists this would entail $(f^\bullet \cap Q_{A \multimap B}); (g^\bullet \cap Q_{B \multimap C}) \neq (f^\bullet; g^\bullet) \cap Q_{A \multimap C}$. On the other hand, $(f^\bullet \cap Q_{A \multimap B}); (g^\bullet \cap Q_{B \multimap C}) \subseteq (f^\bullet; g^\bullet) \cap Q_{A \multimap C}$. Therefore, we settle for a slightly lesser property, that we expose in the below definition.

Definition 7.24. ChuLinNom *is the category with objects the 5-tuples* $A = (\mathbb{D}_A, P_A, O_A, T_A, T_A^\perp)$, *where* $\mathbb{D}_A$ *is an object of* Qual, $P_A \subseteq \mathbb{D}_A$, $O_A \subseteq \mathbb{D}_A$, $T_A \subseteq \mathcal{P}_{\mathsf{finsup}}(P_A)$, *and* $T_A^\perp \subseteq \mathcal{P}_{\mathsf{finsup}}(O_A)$ *that are freely generated from* $[\![X]\!]$ *by tensor, and negation. The morphisms* $A \to B$ *are maps* $f : \mathsf{ParNomRel}(\mathbb{D}_A, \mathbb{D}_B)$ *such that:*

- $\exists \mathcal{R} \in T_{A \multimap B}$ *satisfying* $\mathcal{R} \subseteq f^\bullet \cap Q_{A \multimap B}$
- $f^\bullet \cap Q_{A \multimap B} \subseteq P_{A \multimap B}$

This way, we will be able to prove that composition is well defined. Each object of ChuLinNom is the denotation of a formula of MLL^-, that is, we forget about the multiplicative units $\perp, I$. Indeed, the proof of composition we have only works while restricting to this case. Furthermore, this subcategory is enough as we aim for a full completeness result for MLL^-, the case with units is left open.

In order to tackle the proof of this property we introduce the function $\mathcal{I}$ that generalises the concept of O and P-position. The function $\mathcal{I} : \mathbb{D} \to \mathbb{Z} \setminus \{0\}$ (set that we refer to as $\mathbb{Z}^*$ in the future) sends any position to a number, that is supposed to represent a generalised notion of polarity. It it sound in the sense that it sends P-positions to 1 and O-positions to -1. We will refer to $\mathcal{I}(x)$ as the payoff of x.

The function $\mathcal{I}$ is defined by induction as follows:

- For $[\![X]\!]$, $\mathcal{I}(\bot) = -1$ and $\mathcal{I}(a) = 1$.
- Given any formula F, $\mathcal{I}_{F^\bot}(x) = -\mathcal{I}_F(x)$.
- Given two formulas A and B, $\mathcal{I}$ is computed according to the following table , where given an element $(x, y) \in [\![A \otimes B]\!]$, we write p, q for $\mathcal{I}_A(x)$ and $\mathcal{I}_B(y)$ respectively, and distinguish between the cases where p (respectively q) are positive and negative:

$\otimes$	$p > 0$	$p < 0$
$q > 0$	$p + q - 1$	if $q > \|p\|$ then $p + q$ if $q \leq \|p\|$ then $p + q - 1$
$q < 0$	if $p > \|q\|$ then $p + q$ if $p \leq \|q\|$ then $p + q - 1$	$p + q$

Another way of presenting this table is by introducing two new functions:

- $\eta : \mathbb{Z} \to \mathbb{Z}^* : \begin{cases} n \mapsto n & \text{if } n > 0 \\ n \mapsto n - 1 & \text{if } n \leq 0 \end{cases}$
- $\iota : \mathbb{Z}^* \to \mathbb{Z} : \begin{cases} n \mapsto n & \text{if } n > 0 \\ n \mapsto n + 1 & \text{if } n < 0 \end{cases}$

This is not hard to see that η, ι are inverse to one another. Then the above table could be synthesised into $p \otimes q = \eta(\iota(p) + \iota(q) - 1)$. This allows us to conclude about the associativity of the $\otimes$:

$$p \otimes (q \otimes r) = \eta(\iota(p) + \iota(\eta(\iota(q) + \iota(r) - 1)) - 1) = \eta(\iota(p) + \iota(q) + \iota(r) - 2) = (p \otimes q) \otimes r.$$

By duality, we obtain the associativity for $\invamp$ as well. We will also make use of the following property in a future proof.

Lemma 7.25.
- $\forall m \in \mathbb{Z}^*. m \invamp 1 = m \otimes 2$.
- $\forall m \in \mathbb{Z}^*. m \otimes -1 = m \invamp -2$.

Proof. We only need to prove the first point by duality. That is, $-(-m \otimes -1) = (m \otimes 2)$.

- if $m > 0$ then $m \otimes 2 = m + 2 - 1 = m + 1$. On the other and $(-m \otimes -1) = (-m - 1)$ as expected.
- if $m = -1$ then $m \otimes 2 = 1$, and $(-m \otimes -1) = 1 - 1 - 1 = -1$ as expected.
- if $m < -1$ then $m \otimes 2 = 2 + m - 1 = m + 1$, and $(-m \otimes -1) = -m - 1$.

□

Lemma 7.26. *Let* $A \in \mathsf{Obj}(\mathsf{ChuLinNom})$:

- *Let* $x \in P_A$. *Then* $\mathcal{I}(x) = 1$.
- *Let* $x \in O_A$. *Then* $\mathcal{I}(x) = -1$.
- *Let* $x \in Q_A = P_A \uplus O_A$. *Then* $\mathcal{I}(x) = 1 \Rightarrow x \in P_A$, $\mathcal{I}(x) = -1 \Rightarrow x \in O_A$.

Proof. The third point is a direct consequence of the two first ones. We prove the two first points together. The proof is by induction on the structure of the formula. If it is atomic, then it is by definition. In the case of $A = A_1 \otimes A_2$, then $\mathcal{I}(x_1 \otimes x_2) = \mathcal{I}(x_1) \otimes \mathcal{I}(x_2) = 1 \otimes 1 = 1$ since $x_1 \in P_{A_1}, x_2 \in P_{A_2}$ by definition of $P_{A_1 \otimes A_2}$. Finally, if $A = A_1 \parr A_2$, then $\mathcal{I}(x) = \mathcal{I}(x_1) \parr \mathcal{I}(x_2)$, and $\mathcal{I}(x_1) = -1, \mathcal{I}(x_2) = 1$, or the other way around $\mathcal{I}(x_1) = 1, \mathcal{I}(x_2) = -1$, by definition of $P_{A \parr B}$. Then $-1 \parr 1 = 1$, and we conclude that $\mathcal{I}(x) = 1$. The proof for the $O_{A \parr B}$ and $O_{A \otimes B}$ is dealt with similarly. □

We will also need the following lemma.

Lemma 7.27.
- *$\forall F$ formula of* MLL^-*.$T_F \neq \emptyset \wedge T_F^\perp \neq \emptyset$,*
- *$\forall F$ formula of* MLL^-*, $\forall \mathcal{R} \in T_F . \mathcal{R} \neq \emptyset$ and $\forall \mathcal{R}' \in \mathcal{T}_F^\perp . \mathcal{R}' \neq \emptyset$.*

Proof. This is proven by mutual induction along the structure of the formula F. For the first point, the atomic case is by definition. In the case $F = A \otimes B$, then we simply consider a relation $\mathcal{R}_A \times \mathcal{R}_B \in T_{A \otimes B}$ and a counter-strategy $P_A \times \tau_B \uplus \tau_A \times P_B$, where $\tau_A \in T_{A^\perp}$, and $\tau_B \in T_{B^\perp}$. The case $A \parr B$ is dealt with similarly.

For the second point the atomic case is by definition. For the elements of T, the inductive case $\otimes$ is automatic. For $T_{A \otimes B}^\perp$, given a relation $\mathcal{R} \in T_{A \otimes B}^\perp$, then given $Q \in T_A$, $Q ;_A \mathcal{R} \in T_B^\perp$. Hence $Q ;_A \mathcal{R} \neq \emptyset$ by inductive hypothesis and hence $\mathcal{R} \neq \emptyset$. The case for relations in $A \parr B$ is proven along the same lines. □

Proposition 7.28. *Let $f \in \mathsf{ParNomRel}(A, B)$ such that $\exists \mathcal{R} \in T_{A \multimap B} . \mathcal{R} \subseteq f^\bullet$. Then for any $x \in f^\bullet, \mathcal{I}(x) = 1$.*

Proof. Instead of working with $A^\perp \parr B$, we work with a general formula of MLL^-. So let $f^\bullet : \mathbb{D}$ where $\mathbb{D}$ is the qualitative polarised domain $[\![F]\!]_{\mathsf{Qual}}$ for a formula F of MLL^-. As there exists $\mathcal{R}$ in T_F such that $\mathcal{R} \subseteq f^\bullet \cap Q_F$, this implies, by the lemma above 7.27 that there exists some $x \in f^\bullet$ such that $x \in P_F$. Consequently $\mathcal{I}(x) = 1$.

Now we prove the following property: let us assume x such that $\mathcal{I}(x) = 1$, and primes p, p' such that p O-prime, p' P-prime, $\mathsf{p}, \mathsf{p}' \uparrow x$, and both $\mathsf{p}, \mathsf{p} \in x$ or $\mathsf{p}, \mathsf{p}' \notin x$. Then $\mathcal{I}(x \uplus \{\mathsf{p}, \mathsf{p}'\}) = 1$ or $\mathcal{I}(x \setminus \{\mathsf{p}, \mathsf{p}'\}) = 1$. That is, if we start from a position x of payoff 1, and do, or undo, a pair of OP primes, then we reach a new position of payoff 1. As every position of $f^\bullet$ can be obtained from one another by adding or removing pairs of OP-primes (this follows from the definition of $f^\bullet$) this will allow us to conclude that every position y of $f^\bullet$ satisfies $\mathcal{I}(y) = 1$.

In order to establish that, we simply need to prove that given a position x of payoff n, then if we add an opponent prime to it, or remove a player prime from it, we reach a position $n \otimes -1$. By duality, if we add a player prime, or remove an opponent prime to a position of payoff n, then we reach a position of payoff $n \parr 1$. Therefore, if we start from a position of payoff 1, and add, or remove, a couple of OP primes then we will reach a position of payoff $1 \otimes -1 \parr 1 = 1$.

The proof is done by induction on the structure of the formula F. If x is a position of an atomic formula X, then if x is a prime a, x has payoff 1. Then if we remove a prime from x we end up in the position $\perp$, that has payoff $-1 = 1 \otimes -1$. Similarly, if x has payoff -1 then if we add a P-prime to it, we reach a position of payoff $1 = -1 \parr 1$. The same reasoning works for $X^{\perp}$. We now focus on the induction case. Imagine that x is a position of a qualitative polarised domain denotation of a formula $A = A_1 \otimes A_2$, and $\mathcal{I}(x) = 1$. Then let us imagine we add a O-prime to x, and we consider without loss of generality that this O prime is in A_1. Then setting $n_1 = \mathcal{I}(x \upharpoonright A_1), n_2 = \mathcal{I}(x \upharpoonright A_2)$, we reach a position of payoff $(n_1 \otimes -1) \parr n_2 = (n_1 \parr -2) \parr n_2 = (n_1 \parr n_2) \parr -2 = (n_1 \parr n_2) \otimes -1$. The other cases are dealt similarly.

So we can conclude that $\forall x \in f^{\bullet}.\mathcal{I}(x) = 1$. □

This was the last missing element needed to prove composition. As we proved composition only for morphisms of MLL^{-}, we have to work in the category without units. It was studied in [25] how to characterise unitless star-autonomous categories. It resulted in the following definition.

Definition 7.29. *A* ***symmetric semi-monoidal closed category*** *is described by the following data:*

- *A category $\mathcal{C}$.*
- *Two functors $\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ and $\multimap : \mathcal{C}^{\mathsf{op}} \times \mathcal{C} \to \mathcal{C}$.*
- *Three natural isomorphisms corresponding to the following properties:*
  1. *(symmetry) $A \otimes B \simeq B \otimes A$.*
  2. *(associativity) $A \otimes (B \otimes C) \simeq (A \otimes B) \otimes C$.*
  3. *(closure) $\mathcal{C}(A \otimes B, C) \simeq \mathcal{C}(A, B \multimap C)$.*
- *A functor $J : \mathcal{C} \to \mathsf{Set}$ together with a natural isomorphisms $\mathcal{C}(A, B) \simeq J(A \multimap B)$.*

A ***semi star-autonomous category*** *is a symmetric semi-monoidal closed category with a full and faithful functor $(.)^{\perp} : \mathcal{C} \to \mathcal{C}^{\mathsf{op}}$ and a natural isomorphism:*

- *$\mathcal{C}(A \otimes B, C^{\perp}) \simeq \mathcal{C}(A \otimes B^{\perp}, C)$.*

Proposition 7.30. ChuLinNom *is a semi star-autonomous category.*

Actually, it is a sub semi star-autonomous category of ParNomRel. That is, the tensor, and negation operation lift from ParNomRel to ChuLinNom.

Proof. We start by proving that the morphisms compose. We consider two morphisms $f : A \to B$ and $g : B \to C$ of ChuLinNom. Given $\mathcal{R} \in T_{A \multimap B}$ such that $\mathcal{R} \subseteq f^{\bullet} \cap Q_{A \multimap B}$, and $Q \in T_{B \multimap C}$ such that $Q \subseteq g^{\bullet} \cap Q_{B \multimap C}$, then $\mathcal{R}; Q \subseteq (f; g)^{\bullet} \cap Q_{A \multimap C}$, and, by definition, $\mathcal{R}; Q \in T_{A \multimap C}$. To finish, we simply need to prove that $(f^{\bullet}; g^{\bullet}) \cap Q_{A \multimap C} \subseteq P_{A \multimap C}$. As $\exists \mathcal{R} \in T_{A \multimap C}$ such that $\mathcal{R} \subseteq f^{\bullet}$, we know by proposition 7.28 that $\forall x \in f^{\bullet}.\mathcal{I}(x) = 1$. In particular, by lemma 7.26, we can conclude that $\forall x \in f^{\bullet} \cap Q_{A \multimap B}.x \in P_{A \multimap B}$ as expected.

We tackle monoidality, that is, that $\otimes$ indeed acts as a functor. The definition of $T_{A \otimes B}$ as $T_A \times T_B$ is coherent with $P_{A \otimes B} = P_A \times P_B$. That is, given two relations $\mathcal{R}_1, \mathcal{R}_2 \in T_A, T_B$, then by definition $\mathcal{R}_A \times \mathcal{R}_B \subseteq P_{A\otimes B}$. Finally, given two morphisms $f : A, g : B$, and given $\mathcal{R}_1 \subseteq f^\bullet \cap Q_A$, $\mathcal{R}_2 \subseteq g^\bullet \cap Q_B$, then $\mathcal{R}_1 \times \mathcal{R}_2 \subseteq f^\bullet \times g^\bullet \cap Q_{A\otimes B}$. Indeed, as $f^\bullet \cap Q_A \subseteq P_A$, and respectively $g^\bullet \cap Q_B \subseteq P_B$, it automatically entails that $f^\bullet \times g^\bullet \cap Q_{A\otimes B} = f^\bullet \cap Q_A \times g^\bullet \cap Q_B \subseteq P_A \times P_B = P_{A\otimes B}$ as expected. The symmetry and associativity natural isomorphisms are the ones of ParNomRel.

We now prove monoidal closure. Let $f \in$ ChuNomLin$(A \otimes B, C)$, that is $f : (A \otimes B)^\perp \parr C = A^\perp \parr (B^\perp \parr C)$. Then there exists $\mathcal{R} : T_{(A\otimes B)\multimap C}$ such that $\mathcal{R} \subseteq f^\bullet \cap Q_{(A\otimes B)\multimap C}$. However, as $(A \otimes B)^\perp \parr C = A \multimap (B \multimap C)$, then one can see $f^\bullet$ as a morphism $A \to B \multimap C$, and similarly, one can see $\mathcal{R}$ as belonging in $T_{A\multimap(B\multimap C)}$, with $\mathcal{R} \in f^\bullet \cap Q_{A\multimap(B\multimap C)}$.

Moreover, the definition of J is obvious setting $J(A)$ being the set of morphisms of ParNomRel(I, A) that satisfy the desired conditions. Finally, the negation provides the desired functor required for characterising semi star-autonomous categories. □

7.2.3 Full completeness

MDNF stands for multiplicative disjunctive normal form. It is the form for formulas of MLL$^-$ that results from applying the transformations $(A \parr B) \otimes C \to A \parr (B \otimes C)$, and its right variant, in order to dispose of the $\parr$ that are not at the bottom level in a formula.

Definition 7.31. *A sequent Γ of* MLL *is **MDNF** if Γ is a multiset of formulas F_i and each $F_i = \otimes_j X_{i,j}$ where $X_{i,j}$ is a literal.*

A model is MDNF-fully complete if it is fully complete for every MDNF-sequent. That is, given a MDNF sequent Γ, then every morphism $\sigma : I \to [\![\Gamma]\!]$ is the denotation of a proof. Proving MLL$^-$ full-completeness relying on MDNF formulas, and sequents, is an argument drawn from [84], itself taking inspiration from [5]. The proof structures, and proof nets, of MLL$^-$-MDNF formulas are much simplified compared to the ones of MLL$^-$. Indeed, as there is no $\parr$, there is no need to consider switchings anymore. We use the term "block" to refer to a formula F_i of $\Gamma = F_1, F_2, ..., F_n$.

Proposition 7.32. ChuLinNom *is MDNF fully-complete. That is, given a MDNF sequent Γ of* MLL$^-$ *, then any morphism $I \to [\![\Gamma]\!]$ is the denotation of a proof $\pi :\vdash \Gamma$.*

Given a morphism f of ParNomRel, there is a canonical proof structure of MLL$^-$ associated to it, since a proof structure of MLL consists of a bijection between negative and positive litterals. We recall that the argument for full completeness is in two steps. Given the proof structure canonically associated with f, we first have to prove that the proof structure is acyclic, and then connected.

Proof. We start by showing that there is no cycle inside a block. Let us assume that there is one, and we name the block F_1. We call $\mathcal{R}$ the relation of T_Γ such that $\mathcal{R} \subseteq f^\bullet \cap Q_\Gamma$. Then let us pick

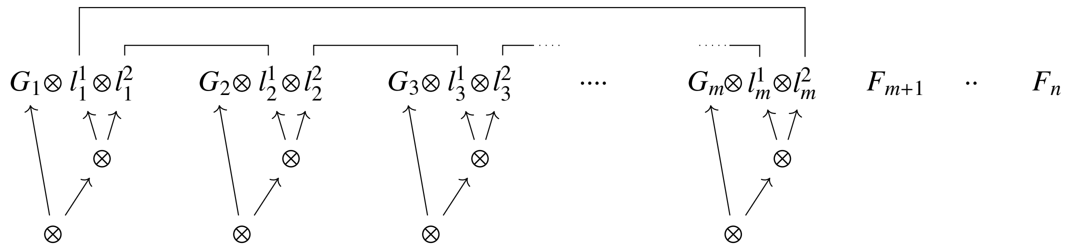


Figure 7.1: A cycle inside a proof structure corresponding to a MDNF sequent

a counter strategy $\mathcal{Q} \in T^{\perp}_{F_2 \parr .... \parr F_n}$, and, making it interact with $\mathcal{R}$ through $\mathcal{R}\,;_{F_2 \parr ... \parr F_m}\, \mathcal{Q} = \mathcal{S}$, we get that $\mathcal{S}$ is a strategy of F_1: $\mathcal{S} \in T_{F_1}$. Let us assume that the block is of the shape $F_1 = Y_1 \otimes Y_2 \otimes ... \otimes Y_l \otimes X \otimes X^{\perp}$, and the strategy establishes an axiom link between X and $X^{\perp}$. Then let us pick a position x of $\mathcal{S}$. As $x \upharpoonright X^{\perp} = x \upharpoonright X$, this entails that $x \upharpoonright X^{\perp} \in P_{X^{\perp}} \Leftrightarrow x \upharpoonright X \in O_X$ and $x \upharpoonright X^{\perp} \in O_{X^{\perp}} \Leftrightarrow x \upharpoonright X \in P_X$. Therefore, $x \upharpoonright X \otimes X^{\perp} \in O_{X \otimes X^{\perp}}$, and $x \notin P_{F_1}$. This is a contradiction. So there is no cycle inside a block.

So let us assume there is a cycle that goes through several blocks, and let us pick one of minimal length. As the cycle is of minimal length, it only goes at most once through each block, passing through two literals. Therefore, given a block F_i on the cycle, we write $F_i = G_i \otimes l_i^1 \otimes l_i^2$, where l_i^1, l_i^2 are the literals that belong in the cycle. We then consider that the F_i have been rearranged in the order of the cycle, starting from l_1^1. This is drawn in the figure 7.1.

Let us pick a counter-strategy of $\mathcal{Q}$ of $F_2 \parr ... \parr F_m \parr F_{m+1} \parr ... \parr F_n$. That is, $\mathcal{Q} \in T^{\perp}_{F_2 \parr .. \parr F_n}$. Then, writing $\mathcal{S} = \mathcal{R}\,;_{F_2 \parr ... \parr F_n}\, \mathcal{Q}$, we get $\mathcal{S} \in T_{F_1}$, In particular, $\mathcal{S}$ is non-empty and therefore $\exists x \in \mathcal{S}$, $x \in P_{F_1}$, $\exists y \in O_{F_2 \parr ... \parr F_m}$ such that $(x, y) \in \mathcal{R}$ and in particular $(x, y) \in f^{\bullet} \cap Q_{\Gamma}$. This implies $y \upharpoonright F_i \in O_{F_i}$ for all i greater than 1. As x is a P-position, this implies that $x \upharpoonright l_1^1 \in P_{l_1^1}$ and $x \upharpoonright l_1^2 \in P_{l_1^2}$. As $y \upharpoonright F_2$ is an O-position, it implies that there is exactly one literal l of F_2 in it, such that $y \upharpoonright l \in O_l$. This has to be l_2^1, as $x \upharpoonright l_1^2$ is a P-position, and the polarity of y in l_1^2 must be the opposite. This implies that $y \upharpoonright l_2^2$ is positive. Following the same reasoning, it implies that $y \upharpoonright l_3^1$ is negative, and $y \upharpoonright l_3^2$ positive. Repeating this, we finally obtain that $y \upharpoonright l_m^1$ is negative and $y \upharpoonright l_m^2$ is positive. But the polarity of $y \upharpoonright l_n^2$ is the opposite of the one of $x \upharpoonright l_1^1$ that is positive. Hence $y \upharpoonright l_m^2$ is positive and negative, bringing a contradiction. We deduct the acyclicity.

Proving the connectedness is just as simple. Note that every literal of a block belongs to the same connected component. Suppose that there are two connected components (the result generalises straightforwardly for more), that we split into two sets of blocks $F_1, ..., F_k$ and $F_{k+1}, ..., F_m$. We pick a random block in the first connected component, and name it F_1. We consider a counter-strategy $\mathcal{Q} \in T^{\perp}_{F_2 \parr F_3 \parr .... \parr F_m}$. Let us consider a position (x, y) such that $y \in \mathcal{Q}$ and $x \in \mathcal{R}\,;_{F_2 \otimes ... \otimes F_m}\, y$ as above. Then by definition, x is a player position, and all positions $y \upharpoonright F_i$ are O-positions. In particular, $y \upharpoonright F_{k+1} \parr ... \parr F_m \in O_{F_{k+1} \parr ... \parr F_m}$. On the other hand, we already know that the part of the relation in $F_{k+1} \parr ... \parr F_m$ corresponds to an acyclic proof structure. Furthermore, as it is connected, it corresponds to a proof. Therefore it is a denotation of a proof and belongs in $P_{F_{k+1}\parr\ldots\parr F_m}$ by soundness. Consequently, $y \in P_{F_{k+1}\parr\ldots\parr F_m} \cap O_{F_{k+1}\parr\ldots\parr F_m} = \emptyset$. This is a contradiction, and the structure is connected. □

Finally, we simply note that MDNF full completeness entails MLL^- full completeness in any semi star-autonomous category. This has already been devised in [5, 84]. We repeat the argument here.

Suppose that F is a formula such that there is a morphism $f : I \to F$ whose corresponding proof structure is cyclic (and, or, disconnected) for a given switching S. First, we handle acyclicity. Let us select an occurrence $\parr$ such that the cycle passes through it. Then either the $\parr$ is at a lowest level, in which case we do nothing. Or there is a $\otimes$ that appears before in the parse-tree. That is, it occurs in a context of the form $(A \otimes (B \parr C))$. Then if the switching selects the formula B, we post-compose f with the morphism that is identity almost everywhere but transforms $(A \otimes (B \parr C))$ into $((A \otimes B) \parr C)$. Then there is still a cycle in the newly obtained proof structure corresponding to the new morphism. By doing so repeatedly, we eventually obtain a MDNF formula that has a cycle in it. However, such a morphism is rejected by the model. Therefore, there is no cycle in the original formula. The connectedness proof works on a similar basis.

This allows us to conclude this section with the expected theorem.

Theorem 7.33. ChuLinNom *is fully-complete for* MLL^-.

7.2.4 ChuLinNom, a new definition

This fully complete result allows us to redefine slightly ChuLinNom, by replacing the condition:

$$\exists \mathcal{R} \in T_{A\multimap B}. \mathcal{R} \in f^\bullet \cap Q_{A\multimap B}$$

in the definition, with:

$$f^\bullet \cap Q_{A\multimap B} \in T_{A\multimap B}.$$

We recall that the problem with the second condition was that we were not able to prove the composition. However, now, we can rely on the full-completeness result in order to establish that the second condition composes. Indeed, two morphisms being denotation of proofs of MLL^-, they will compose just as their proof compose.

Proposition 7.34. ChuLinNom *can be presented as the category having:*

- *Polarised coherence objects equipped with Chu-structure as objects:* $A = (\mathbb{D}_A, P_A, O_A, T_A, T_A^\perp)$*, where* $O_A, P_A \subseteq \mathbb{D}_A$*,* $T_A \subseteq \mathcal{P}_{\mathsf{finsup}}(P_A)$*,* $T_A^\perp \subseteq \mathcal{P}_{\mathsf{finsup}}(O_A)$*, such that each object is the denotation of a* MLL^- *formula.*
- *As morphisms* $A \to B$ *morphisms* $f : \mathsf{ParNomRel}(\mathbb{D}_A, \mathbb{D}_B)$ *such that* $f^\bullet \cap Q_{A\multimap B} \in T_{A\multimap B}$.

Proof. To differentiate, we name $\mathsf{ChuLinNom}_2$ the latest construction, and $\mathsf{ChuLinNom}_1$ the former. The identities are morphisms of $\mathsf{ChuLinNom}_2$. Furthermore, given morphisms f, g of $\mathsf{ChuLinNom}_2$, $f \otimes g$ is a morphism of $\mathsf{ChuLinNom}_2$, just as is $f^{\perp}$. Therefore, any cut-free morphism of MLL^- can be modelled within $\mathsf{ChuLinNom}_2$. Furthermore, (forgetting that the composition is not yet defined), $\mathsf{ChuLinNom}_2$ is a sub-semi-star-autonomous category of $\mathsf{ChuLinNom}_1$ (in the sense that every morphism of $\mathsf{ChuLinNom}_2$ is a morphism of $\mathsf{ChuLinNom}_1$ and they share the same tensor and negation). As $\mathsf{ChuLinNom}_1$ is fully-complete, so is $\mathsf{ChuLinNom}_2$, that is, every morphism of $\mathsf{ChuLinNom}_2$ is the denotation of a proof. Therefore, the composition of two morphisms of $\mathsf{ChuLinNom}_2$ results in a morphism that is the denotation of a proof. This one is then a morphism of $\mathsf{ChuLinNom}_2$. That is, $\mathsf{ChuLinNom}_2$ is a category.

Therefore, $\mathsf{ChuLinNom}_2$ is a sub-semi-star-autonomous category of $\mathsf{ChuLinNom}_1$ able to faithfully model axioms, and, as $\mathsf{ChuLinNom}_1$ is fully-complete for MLL^-, $\mathsf{ChuLinNom}_2 = \mathsf{ChuLinNom}_1$. □

Let us note that as this stage, we **do not** make use of names. That is, we could present an even simpler category, if our goal was limited to MLL^-. This one consists in an almost similar definition as ChuLinNom, but relying on a simplified ParNomRel category.

Definition 7.35. *We define* ParRel *as the category:*

- *whose objects are coherence polarised domains obtained by tensor and negation from the labelled Sierpinski domain, that is the 2-elements lattice domain $\mathbf{O}_X$ with $\bot \sqsubseteq \top_X$, whose only prime is positive. Of course, we set $[\![X]\!] = \mathbf{O}_X$. We set $\mathsf{label}(\top_X) = X$. Each object is then a denotation of a* MLL^- *formula.*
- *whose morphisms of* $\mathsf{ParRel}(A, B)$ *are linear monotone functions of $A \multimap B$ subject to the separation conditions and such that* $(\mathsf{p}, \mathsf{p}') \in \mathsf{tr}(f) \Rightarrow \mathsf{label}(\mathsf{p}) = \mathsf{label}(\mathsf{p}')$.

Relying on this simpler category, we can define ChuLin. The objects of ChuLin are construct inductively from $[\![X]\!]$, defined as follows:

$$[\![X]\!]_{\mathsf{ChuLin}} = (\mathbf{O}_X, P_X = \{\top_X\}, O_X = \{\bot\}, T_X = \{\{\top\}\}, T_X^{\perp} = \{\{\bot\}\})$$

We now have the necessary ingredients to define the category.

Definition 7.36. ChuLin *is the category having:*

- *Polarised coherence objects equipped with Chu-structure as objects: $A = (\mathbb{D}_A, P_A, O_A, T_A, T_A^{\perp})$, where $O_A, P_A \subseteq \mathbb{D}_A$, $T_A \subseteq \mathcal{P}_{\mathsf{finsup}}(P_A)$, $T_A^{\perp} \subseteq \mathcal{P}_{\mathsf{finsup}}(O_A)$, such that each object is the denotation of a* MLL^- *formula.*
- *As morphisms $A \to B$ morphisms $f : \mathsf{ParRel}(\mathbb{D}_A, \mathbb{D}_B)$ such that $f^{\bullet} \cap Q_{A \multimap B} \in T_{A \multimap B}$.*

Proposition 7.37. ChuLin *is a star-autonomous category and is fully complete for* MLL^-.

The proof follows the exact same argument as presented before for ChuLinNom.

7.2.5 A connection to graph games

There is a way of seeing $f^\bullet$ as a strategy of $Q_{\mathbb{D}}$ seen as a graph. This bears a central connection to the graph games developed by Hyland and Schalk [51, 52], even though our games and strategies are fundamentally different.

Given a qualitative polarised domain $\mathbb{D}$, we look at $Q_{\mathbb{D}}$ as the graph defined as follows:

- Its set of vertices are the elements of $Q_{\mathbb{D}}$, split into its set of player positions $P_{\mathbb{D}}$, and its set of opponent positions $O_{\mathbb{D}}$.
- An edge e between two positions $x \xrightarrow{\mathsf{e}} y$, $x \neq y$, corresponds to a prime p such that $x \sqcup \mathsf{p} = y$ or $x \backslash \mathsf{p} = y$. In that case we can see that $y \in P_{\mathbb{D}} \Leftrightarrow x \in O_{\mathbb{D}}$ and $x \in O_{\mathbb{D}} \Leftrightarrow y \in P_{\mathbb{D}}$. We write $E_{\mathbb{D}}$ for its set of edges.

We call such a graph a **qualitative polarised graph**.

To make clear the direction in which we travel a prime, we write $x \xrightarrow{\mathsf{p}}$ if we add it, and $x \xleftarrow{\mathsf{p}}$ if we remove it. Therefore, we refer to **moves** for the data of a prime, together with a direction (adding, removing). We usually use the lowercase m to refer to them. A move is an O-move if the edges corresponding to it start from P-positions and target O-positions, and a P-move if they start from O-positions to target P-positions. Equivalently, a move is an O-move if it is an O-prime that is added, or a P-prime that is removed. Similarly, it is a P-move if it is a P-prime that is added, or an O-prime that is removed. We define O-edges and P-edges accordingly. Let us note that when restricted to one direction (that is, either add or remove), then the graph $Q_{\mathbb{D}}$ is acyclic (that is, a dag), but might contain several roots (that are, minimal elements). So if it certainly might be seen as an asynchronous graph, this one does not come from an event structure.

In the work of Hyland and Schalk the strategies for graphs were defined as partial functions α from O-positions to P-positions, such that, for all O-positions x in the domain of α, there is a move m satisfying $x \xrightarrow{\mathsf{m}} \alpha(x)$. In our case, this definition does not work, since we take into account the dynamics. That is, given $x \xrightarrow{\mathsf{p}} y$, such that $y \in O_{\mathbb{D}}$, then this corresponds to adding an O-prime, and the strategy should answer by adding the associated P-prime. On the other hand, considering $z \xleftarrow{\mathsf{p}} y$ (and $y \in O_{\mathbb{D}}$), then this corresponds to removing a P-prime, and the strategy should remove the associated O-prime. Note that given a position x and a prime p (such that $\mathsf{p} \uparrow x$), then there is a unique direction p can be travelled starting from x. Therefore, we might refer to this edge (respectively move) as $\mathsf{e}(x, \mathsf{p})$ (respectively $\mathsf{m}(x, \mathsf{p})$).

Definition 7.38. *A **strategy** on a qualitative polarised graph $Q_{\mathbb{D}}$ is defined as follows:*

- *A set of $\Upsilon \subseteq P_{\mathbb{D}}$ of P-positions.*
- *A partial function $\alpha : E_{\mathbb{D}} \to E_{\mathbb{D}}$ defined as follows. Given an edge $\mathsf{e}(x, \mathsf{p})$ such that $x \in \Upsilon$, and $x \xrightarrow{\mathsf{p}} y$ (respectively $x \xleftarrow{\mathsf{p}} y$), then, if there exists p' such that $y \xrightarrow{\mathsf{p}'} z \in P_{\mathbb{D}} \cap \Upsilon$ (respectively $y \xleftarrow{\mathsf{p}'} z \in P_{\mathbb{D}} \cap \Upsilon$), then this pair (z, p') satisfying the property is unique and we write $\alpha(\mathsf{e}(x, \mathsf{p})) = \mathsf{e}(y, \mathsf{p}')$.*

The data of α is redundant: α is perfectly defined from Υ. In our case, the strategy corresponds to a morphism of f of ChuLinNom, and f is a bijection between primes. So what matters is not the position, but only the prime. That is, the strategy is **history-free** (in the sense of [7]).

Definition 7.39. *A strategy on a qualitative polarised graph is* ***history-free*** *if there exists a function* $\beta : \mathrm{Pr}(\mathbb{D}) \to \mathrm{Pr}(\mathbb{D})$ *such that* $\forall (x, y, \mathsf{p}) \in P_{\mathbb{D}} \times O_{\mathbb{D}} \times \mathrm{Pr}(\mathbb{D})$ *with* $x \xrightarrow{\mathsf{e}(x,\mathsf{p})} y$*, then* $\alpha(\mathsf{e}(x, \mathsf{p})) = \mathsf{e}(y, \beta(\mathsf{p}))$*.*

Finally, we want the strategy to be total.

Definition 7.40. *A strategy on a qualitative polarised graph is* ***total*** *if it is not empty and* $\forall x \in \Upsilon$*,* $\forall \mathsf{p}$ *such that* $x \xrightarrow{\mathsf{e}(x,\mathsf{p})} y$*,* $\mathsf{e}(x, \mathsf{p})$ *being an O-edge,* β *is defined on* p *and* $x \xrightarrow{\mathsf{e}(x,\mathsf{p})} y \xrightarrow{\mathsf{e}(y,\beta(\mathsf{p}))} z \in P_{\mathbb{D}}$*.*

The fact that our morphisms behave that way is demonstrated below.

Proposition 7.41. *Let us consider a function* $f : \mathbb{D}$ *of* ChuLinNom*, where* $\mathbb{D}$ *is the qualitative polarised domain denotation of a* MLL^- *formula. Then* $f^\bullet$ *leads to a strategy that is history-free and total.*

The fact that it leads to a history-free strategy is straightforward. However, the totality is a real property, and the proof is not elementary. Let $x \in f^\bullet \cap Q_{\mathbb{D}}$. By definition, $x \in P_{\mathbb{D}}$. Let p be an O-prime, seen as an O-move, $x \xrightarrow{\mathsf{p}} y$, $y \in O_{\mathbb{D}}$. Then, by definition of f, this prime is mapped to a P-prime p'. Hence the strategy associated with f reacts in $\mathbb{D}$, by playing a P-move $y \xrightarrow{\mathsf{p}'} z$. What we need to prove is that $z \in P_{\mathbb{D}}$. Indeed, at this stage we have no proof that z lies in $Q_{\mathbb{D}}$.

Proof. We rely on full completeness. We prove by induction on the rules of MLL^- that the strategy denoting the proof is total. First, we introduce some terminology. Let $\mathbb{D}$ be a polarised coherent domain denotation of a formula F of MLL^-, and we consider a position $x \in P_{\mathbb{D}}$. We say that x is ready to move in l, where l is a literal of F, if $x \upharpoonright l \in P_l$ and if an element $y \in \mathbb{D}$, which is equal to x on all literals except l, where $y \upharpoonright l \in O_l$, is in $Q_{\mathbb{D}}$ (and hence $y \in O_{\mathbb{D}}$). We say that y is the result of switching x in l. Similarly, we say that $y \in O_{\mathbb{D}}$ is ready to move in l, if the element x, which is equal to y on all literals except l, where $x \upharpoonright l \in P_l$, is in $Q_{\mathbb{D}}$ (and hence $x \in P_{\mathbb{D}}$).

We prove the following intermediate property, which straightforwardly entails that $f^\bullet$ is total. Let $x \in f^\bullet \cap Q_{\mathbb{D}}$. Let $l, l^\perp$ linked by an axiom-link in the proof whose denotation is f. Then if x is ready to move in l, then given y as above, y is ready to move in $l^\perp$. The proof is done by induction on the structure of the proof. For the axiomatic case, this is straightforward. So let us suppose that we have the following rule:

$$\frac{\vdash \Gamma, A \qquad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B} \otimes$$

Let us note that if we pick a literal l of a formula F, such that $x \upharpoonright l \in P_l$ and x is ready to move in l, then it automatically entails that $x \upharpoonright F \in P_F$. Indeed, swapping the polarity of a literal from P to O entails a downward change of the global polarity function $\mathcal{I}(x \upharpoonright F)$. Hence, the only possible case is $x \upharpoonright l \in P_l$, and once the swap happens, leading to y, one has $y \upharpoonright F \in O_l$.

We deal with the different cases, though restricting our study to the cases where the axiom link happens in the Γ, A part of the sequent. The cases where it belongs to the Δ, B are symmetric. We always assume $x \upharpoonright l^{\perp} \in P_{l^{\perp}}$. The case $x \upharpoonright l \in P_l$ is dealt with similarly.

1. $l, l^{\perp} \in \Gamma$. Then let x as above, and $x \upharpoonright l^{\perp} \in P_{l^{\perp}}$ ready to move in Γ. This means $x \upharpoonright \Gamma \in P_{\Gamma}$, and, consequently, $x \upharpoonright A \in O_A$. This entails $x \upharpoonright B \in P_B$ and thus $x \upharpoonright \Delta \in O_{\Delta}$. So suppose we switch $x \upharpoonright l$ from O to P, getting y. Then projecting on Γ, A, we see that $y \upharpoonright \Gamma \in O_{\Gamma}$, and y is ready to switch in $l^{\perp}$. Hence we can switch from O to P in $l^{\perp}$, and reach a position z where $z \upharpoonright \Gamma \in P_{\Gamma}$, $z \upharpoonright \Delta \in O_{\Delta}$, $z \upharpoonright A \in O_A$, $z \upharpoonright B \in P_B$, which is a proponent position as expected.
2. We tackle the case $l \in \Gamma, l^{\perp} \in A$. As x is ready to switch in $l^{\perp}$, and $x \upharpoonright l^{\perp} \in P_{l^{\perp}}$, we get that $x \upharpoonright A \in P_A$, and, similarly $x \upharpoonright A \otimes B \in P_{A \otimes B}$ so $x \upharpoonright B \in P_B$. Hence, we get $x \upharpoonright \Gamma \in O_{\Gamma}$ and $x \upharpoonright \Delta \in O_{\Delta}$. Therefore, doing the two switchings in $l^{\perp}$ in $A \otimes B$, and in l in Γ, will get us a final position z having local polarities $z \upharpoonright A \in O_A$, $z \upharpoonright B \in P_B$, $z \upharpoonright \Gamma \in P_{\Gamma}$ and $y \upharpoonright \Delta \in O_{\Delta}$. Therefore $z \in P_{\Gamma}$.
3. The third case is $l^{\perp} \in \Gamma$, and $l \in A$. This time, we get $x \upharpoonright \Gamma \in P_{\Gamma}$, $x \upharpoonright \Delta \in O_{\Delta}$, $x \upharpoonright A \in O_A$, $x \upharpoonright \Delta \in P_{\Delta}$. After swapping x in both l, and $l^{\perp}$, we get $z \upharpoonright \Gamma \in O_{\Gamma}$, $z \upharpoonright \Delta \in O_{\Delta}$, $z \upharpoonright A \in P_A, z \upharpoonright B \in P_B$, hence this results in a successful switch.
4. The last case is $l^{\perp}$ in A and $l \in A$. With this we have $x \upharpoonright A \in P_A$, $x \upharpoonright B \in P_B$, $x \upharpoonright \Gamma \in O_{\Gamma}, x \upharpoonright \Delta \in O_{\Delta}$. Once we switch x in $l, l^{\perp}$ resulting in a position z, we have $z \upharpoonright A \in P_A$, and the others local polarities remained unchanged. So the final element remains in P_{Γ}.

The case of an $\parr$ rule is straightforward as it is interpreted as identity, hence it does not change the local, and hence global, polarities.

Therefore, the property holds for any f denotation of a proof. By full completeness, this is the case for any f satisfying the Chu-conditions. □

Within the work of graph-games, a property often expected from strategies is conflict-freeness. However, in our case, one has to fix an orientation to define it properly. For this paragraph, we suppose we have fixed an orientation. We write $x \leq z$ if there is an oriented path $x \twoheadrightarrow z$. Suppose $x, z \in X$ such that $x \leq z$. Then the strategy is **conflict-free**, if given an O-move $x \xrightarrow{\mathsf{m}} y$, such that $y \leq z$, then the strategy answers with a move $y \xrightarrow{\mathsf{n}} w$, and $w \leq z$. This holds straightforwardly for strategies coming from morphisms of $\mathsf{ChuLinNom}$.

We believe that this topic is worth being a subject for further investigation. We leave open the questions of characterising precisely the strategies that arise from morphisms of $\mathsf{ChuLinNom}$, or, equivalently, from proofs.

7.3 Hypercoherences and MALL full completeness

The goal of this section is to enrich the former model with an additive structure and prove a MALL full completeness result. The former model has a natural additive structure, that consists in:

$$A \oplus B = \{\mathsf{inl}(\mathbb{D}_A) \uplus \mathsf{inr}(\mathbb{D}_B), \mathsf{inl}(P_A) \uplus \mathsf{inr}(P_B), \mathsf{inl}(O_A) \uplus \mathsf{inl}(O_B), \mathsf{inl}(T_A) \uplus \mathsf{inr}(T_B), \mathsf{inl}(T_A^\perp) \times \mathsf{inr}(T_B^\perp)\}$$
$$A \,\&\, B = \{\mathsf{inl}(\mathbb{D}_A) \uplus \mathsf{inr}(\mathbb{D}_B), \mathsf{inl}(P_A) \uplus \mathsf{inr}(P_B), \mathsf{inl}(O_A) \uplus \mathsf{inl}(O_B), \mathsf{inl}(T_A) \times \mathsf{inr}(T_B), \mathsf{inl}(T_A^\perp) \uplus \mathsf{inr}(T_B^\perp)\}$$

Unfortunately, we have not been able to prove full completeness with this definition of additives, and hence relied on an alternative one. We will first decorate the morphisms of ParNomRel with a notion of hypercoherence, and prove some properties about the thus obtained category. Then, we will add the Chu-conditions, refining the previously obtained category. This detour through ParNomRel is needed in order to establish some properties, and to precisely describe what properties the hypercoherence enforces.

7.3.1 Polarised coherence hypercoherence spaces

Originally, concurrent games were enriched with a a weak coproduct, where the proponent could choose with two additional moves whether he would like to pick the left or right component. These additional moves were part of the reason why one needed to use a quotient eventually. Indeed, for instance, in that case $(A \oplus B) \oplus C$ and $A \oplus (B \oplus C)$ are not denoted by the same arena, allowing morphisms to act differently on these two. Here, we choose a different route. What used to be dynamic is now presented in a static way thanks to hypercoherences.

Therefore, the objects we will now work with are pairs $(A, \Gamma(A))$ where A is a sum of polarised coherence domains, and $\Gamma(A) \subseteq \mathbb{D}_A$ its set of coherence. Formally, given two polarised coherence domains A, B we define their sum as follows:

$$A \oplus B = (\mathbb{D}_A \uplus \mathbb{D}_B, \sqsubseteq_A \uplus \sqsubseteq_B, \lambda_{A \oplus B} = \lambda_A \uplus \lambda_B : \mathrm{Pr}(A) \uplus \mathrm{Pr}(B)).$$

$A \oplus B$ is not a polarised coherence domain, as it does not have a unique minimal element. We call the resulting element a polarised coherence object.

Definition 7.42. *A* ***polarised coherence object*** *is an object of the form $\uplus_i \mathbb{D}_i$, where each $\mathbb{D}_i$ is a polarised coherence domain.*

The tensor product lifts straightforwardly to polarised coherence objects: given two polarised objects $\uplus \mathbb{D}_i$ and $\uplus \mathbb{D}_j$, then $(\uplus_i \mathbb{D}_i) \otimes (\uplus \mathbb{D}_j) = (\uplus_i \mathbb{D}_i) \times (\uplus \mathbb{D}_j) \simeq \uplus_{i,j} \mathbb{D}_i \times \mathbb{D}_j$. Similarly, the negation is defined by negating all objects $(\uplus_i \mathbb{D}_i)^\perp = \uplus_i \mathbb{D}_i^\perp$. This lifting from domains to objects via sums is analogous to the family construction [8].

We remind that, given a formula F of MALL, we call a &-resolution (respectively $\oplus$-

resolution, additive-resolution) the choice, for each & (respectively for each $\oplus$, for each & and $\oplus$) of one of its premises. Given Ψ a &−, or a $\oplus$−, or an additive resolution of F, F being a MALL formula, we write $F \upharpoonright \Psi$ for the formula arising by selecting the sub-formulas of F according to Ψ. We enlarge our definition of resolutions to polarised coherence objects, by seeing them as denotations of formulas of MALL. We define $\mathbb{D} \upharpoonright \Psi$ similarly as for formulas of MALL. That is, if $\mathbb{D}$ is a denotation of a formula F of MALL, Ψ a resolution of F, then $\mathbb{D} \upharpoonright \Psi$ is the denotation of $F \upharpoonright \Psi$. Given a polarised coherence object $\mathbb{D}$, an element $x \in \mathbb{D}$, and Ψ a &, or $\oplus$, or additive-resolution of $\mathbb{D}$, we say that x is on Ψ if x is coming from the canonical injection from $\mathbb{D} \upharpoonright \Psi$ into $\mathbb{D}$. Similarly, given $\mathcal{R}$ a relation of $\mathbb{D}$, (that is, a subset of $\mathbb{D}$), we write $\mathcal{R} \upharpoonright \Psi$ for the restriction of $\mathcal{R}$ to $\mathbb{D} \upharpoonright \Psi$. When working with elements, or relations, we may slightly abuse notation in the following way: given an element x of $\mathbb{D}$ (respectively $\mathcal{R} \subseteq \mathbb{D}$), and Ψ a resolution of $\mathbb{D}$ such that x (respectively $\mathcal{R}$) is coming from the injection $\mathbb{D} \upharpoonright \Psi \hookrightarrow \mathbb{D}$, then we might consider that $x = x \upharpoonright \Psi$ (or $R \subseteq \mathbb{D} \upharpoonright \Psi$).

We define a new category by describing morphisms between polarised coherence objects enriching with a notion of hypercoherence. We write $\mathcal{P}_{\mathsf{fin}}$ for the finite subsets operator, and $\mathcal{P}^*_{\mathsf{fin}}$ for the non-empty finite subsets operator.

Definition 7.43. *A* ***polarised coherence hypercoherence space*** *is a pair* $(A, \Gamma(A))$ *where* A *is a polarised coherence object and* $\Gamma(A) \subseteq \mathcal{P}^*_{\mathsf{fin}}(\mathbb{D}_A)$ *is its set of hypercoherence.*

We furthermore write $\Gamma^*(X)$ for the subset of $\Gamma(X)$ of elements that are not singletons. We define the category HypGraph as having (some, not all) qualitative coherence hypercoherence spaces as objects.

$$\begin{aligned} [\![X]\!]_{\mathsf{HypGraph}} &= \{[\![X]\!]_{\mathsf{Qual}}, \Gamma^*(X) = \{x \uplus \bot \mid x \subseteq \mathcal{P}_{\mathsf{fin},>1}(\mathbb{A}_X)\}\} \\ [\![I]\!]_{\mathsf{HypGraph}} &= \{[\![I]\!]_{\mathsf{Qual}}, \Gamma(1) = \{\top\}\}, \end{aligned}$$

where we denote $\mathcal{P}_{\mathsf{fin},>1}(\mathbb{A}_X)$ the set of finite subsets of $\mathbb{A}_X$ of cardinal more than 1. That is, $\Gamma(X)$ is the set of singletons, together with the sets $\{a_1, ..., a_n, \bot\}$. The way to think about it is that the $\bot$ switches the hypercoherence. That is, for instance, $\{a\} \in \Gamma(X)$ and $\{a, \bot\} \in \Gamma^\perp(X)$. Similarly, $\{a_1, ..., a_n\} \in \Gamma^\perp(X)$, and $\{a_1, ..., a_n, \bot\} \in \Gamma(X)$. For the remaining objects, it is defined by repeated applications of the required rules associated with the connectives. These were already defined in 3.5, and we briefly remind them below.

$$\begin{aligned} \Gamma(A \otimes B) &= \{w \in \mathcal{P}^*_{\mathsf{fin}}(\mathbb{D}_A \times \mathbb{D}_B) \mid w \upharpoonright \mathbb{D}_A \in \Gamma(A) \wedge w \upharpoonright \mathbb{D}_B \in \Gamma(B)\} \\ \Gamma(A \oplus B) &= \{w \in \mathcal{P}_{\mathsf{fin}}(\mathbb{D}_A \uplus \mathbb{D}_B) \mid w \in \Gamma(A) \vee w \in \Gamma(B)\} \\ \Gamma(A^\perp) &= \mathcal{P}^*_{\mathsf{fin}}(\mathbb{D}_A) \setminus \Gamma^*(A) \end{aligned}$$

Definition 7.44. *The category* HypGraph *is the category that has:*

- *as objects polarised coherence hypercoherence spaces* $(A, \Gamma(A))$ *generated by induction from* $[\![X]\!]_{\mathsf{HypGraph}}, [\![I]\!]_{\mathsf{HypGraph}}$ *using* $\otimes, \oplus$ *and* $(.)^\perp$.
- *as morphisms* $A \to B$ *the nominal relations* $\mathcal{R} : \mathbb{D}_{A \multimap B}$ *such that:*

1. *(P1) For each &-resolution Ω of $A \multimap B$, there is an additive resolution Ψ on Ω such that $\mathcal{R} \upharpoonright \Psi \in \mathsf{ParNomRel}(A \multimap B \upharpoonright \Omega)$*
2. $\forall s \in \mathcal{P}^*_{\mathsf{fin}}(\mathcal{R})$, $s \in \Gamma(A \multimap B)$.

For a morphism $\mathcal{R} : A \to B$, the condition (P1) translates as: for each $\oplus$-resolution of Ψ_A of A, for each &-resolution Ψ_B of B there exists a &-resolution Φ_A on $A \upharpoonright \Psi_A$ and a $\oplus$-resolution Φ_B on $B \upharpoonright \Psi_B$ such that $\Psi_A\Phi_A \upharpoonright \mathcal{R} \upharpoonright \Psi_B\Phi_B$ is a partial nominal relation $(A \upharpoonright \Psi_A\Phi_A) \to (B \upharpoonright \Psi_B\Phi_B)$.

In order to prove that HypGraph forms a category, the main difficulty is to establish that the condition $(P1)$ composes. To our knowledge, it has never been proven before. An attempt was made through characterising the proof-structures that are hypercoherent in [87], but the author noticed the difficulty in proving that these compose with this method. The property could have followed a good correspondence between game-semantics and hypercoherence, as attempted in [28, 18]. However, as the games they were working with were sequential, this conflicted with the non-sequential, and non-polarised aspect of linear logic.

We present below a proof that this condition composes. This is done by establishing that the hypercoherences give rise to concurrent operators, as originally defined in [10]. As we have not proven that HypGraph forms a category at this point, we will speak about the pre-category HypGraph.

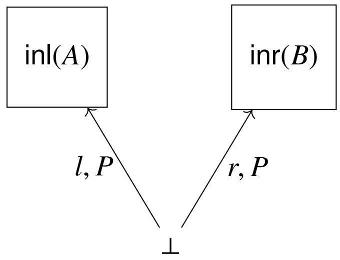
7.3.2 Projecting hypercoherences on concurrent games

In this section we establish that each (P1) hypercoherence, that is, a morphism of HypGraph, produces a concurrent operator on the arena of additive-resolutions (that we define later); the exact same arena that was used by Abramsky and Melliès in their seminal paper on concurrent games [10]. More precisely, we establish that each hypercoherence gives rise to a concurrent operator that is civil, and total. These are the same properties that were used in [10] to provide the proof of full completeness. Establishing this allows us to prove that morphism of HypGraph composes. Given two morphisms $\mathcal{R}$: HypGraph(A, B), and Q : HypGraph(B, C), a $\oplus$-resolution Ψ_A of A, a &-resolution Ψ_C of C, the goal is to prove that there is an additive resolution Φ_B of B, that can also be seen as an additive resolution of $B^\perp$, such that $\Psi_A \upharpoonright \mathcal{R} \upharpoonright \Phi_B \neq \emptyset$, and $\Phi_B \upharpoonright Q \upharpoonright \Psi_C \neq \emptyset$. That is, there is an additive resolution on which $\mathcal{R}$ and Q meets. As a result, there exists a &-resolution Φ_A on $A \upharpoonright \Psi_A$ such that $\Psi_A.\Phi_A \upharpoonright \mathcal{R} \upharpoonright \Phi_B \in \mathsf{ParNomRel}(A \upharpoonright \Psi_A.\Phi_A, B \upharpoonright \Phi_B)$. Furthermore, we prove that this additive resolution is unique.

We start by defining an arena for each formula of linear logic. Each arena is a polarised di-domain, where each prime corresponds to the left or right branch of an additive resolution. We furthermore transform it into a polarised di-domain, by giving a polarity to each prime.

- $[\![X]\!] = (\{\perp\}, \sqsubseteq = \{(\perp, \perp)\}, \perp)$ for each literal X.
- $[\![A \otimes B]\!] = [\![A \parr B]\!] = [\![A]\!] \times [\![B]\!]$

- $[\![A \oplus B]\!]$ is defined as follows:



  Formally, $\mathbb{D}_{A\oplus B} = (\mathbb{D}_A \uplus \mathbb{D}_B \uplus \{\bot\}, \sqsubseteq = \sqsubseteq_A \uplus \sqsubseteq_B \uplus \bot \times \mathbb{D}_A \uplus \bot \times \mathbb{D}_B, \bot)$, and the two new primes hence created have polarity $+1$.
- $[\![A^\bot]\!] = [\![A]\!]^\bot$, where the $(.)^\bot$ operation consists in swapping the polarity of the primes. Consequently, $[\![A \& B]\!] = [\![A^\bot \oplus B^\bot]\!]^\bot$.

Note that these arenas are not alternated. There might be two moves of the same polarity right after another. For instance, in the arena $(A \oplus (B \oplus C))$, there are two succesive moves of polarity P (for instance $(r, P).(l, P)$, picking up the formula B). This interpretation is the same as the one of [10], with instantiation $[\![X]\!] = \{\bot\}$ for each literal. This domain was also studied in more depth in [2], in an attempt to redefine Girard and Hugues and Van Glabbeek's proof structures and their correctness criteria in the context of di-domains. Let us note that each position of the domain corresponds to a partial additive-resolution of the formula, and the maximal positions correspond precisely to total additive resolutions.

First of all, we define a polarity for each position of our domain. This time, the polarity is going to range over a three-elements set: $\{O, P, N\}$. We define the polarity of the unique element $\bot$ of $[\![X]\!], [\![X^\bot]\!]$ to be N. The polarity of the the root of $[\![A \oplus B]\!]$ is O (seen as opponent winning), and P for the root element of $[\![A \& B]\!]$. For the multiplicative case, we devise the polarity as follows:

$\otimes$	O	P	N
O	O	O	O
P	O	P	P
N	O	P	N

For the $⅋$, we have, as expected $K ⅋ L = (K^\bot ⅋ L^\bot)^\bot$ where $O^\bot = P$, $P^\bot = O$, and $N^\bot = N$. That is:

$⅋$	O	P	N
O	O	P	O
P	P	P	P
N	O	P	N

At the level of each prime, seen as a position, the polarity is going to behave as follows: a prime-position is P if all the primes starting from the position are O-primes. It is N if neither opponent nor proponent can play. Respectively, it will be O if all the primes from it are P.

Given $\mathcal{R}$ a relation of $\mathsf{HypGraph}(A, B)$, we strip it by forgetting all the names that happen in it. That is, we focus on the elements of the relations that consist solely of sequences of $\bot$. To put it formally, we define:

$$\tilde{\mathcal{R}} = \{x \in \mathcal{R} \mid \nu(x) = \emptyset\}.$$

Of course, $\tilde{\mathcal{R}} \subseteq \mathcal{R}$, and furthermore $\tilde{\mathcal{R}}$ is finite so $\tilde{\mathcal{R}} \in \Gamma(A \multimap B)$: it is hypercoherent, and a clique (every subset of $\tilde{\mathcal{R}}$ is hypercoherent). Note that for any relation $\mathcal{R}$ in $\mathsf{ParNomRel}$, the bottom element $\bot$ belongs to $\mathcal{R}$. Therefore, $\tilde{\mathcal{R}}$ will map each subset of $\mathcal{R}$ that is a relation of $\mathsf{ParNomRel}$ to a single element. Respectively, each element in $\tilde{\mathcal{R}}$ corresponds to a $\mathsf{ParNomRel}$ relation.

Definition 7.45. *We define* $\mathsf{SimpleHypGraph}$*, as the full sub-pre-category of* $\mathsf{HypGraph}$ *that consists of the freely generated class of objects from* $I = (\{\bot\}, \Gamma(I) = \{\{\bot\}\})$ *under the operations* $\otimes, \oplus, \neg$*. The morphisms of* $\mathsf{SimpleHypGraph}$ *are those of* $\mathsf{HypGraph}$ *between the relevant objects.*

As there are no primes in the domains of this category, we forget about polarities. Given $\mathcal{R} \in \mathsf{HypGraph}(A, B)$, $\tilde{\mathcal{R}} \in \mathsf{SimpleHypGraph}(\tilde{A}, \tilde{B})$, where the $\tilde{(.)}$-operation sends a formula A to the same formula where each occurrence of atomic variable has been replaced by an occurrence of I. That is, $\tilde{(.)}$ acts as a functor $\mathsf{HypGraph} \to \mathsf{SimpleHypGraph}$. The goal of this section is to prove that each morphism of $\mathsf{SimpleHypGraph}(\tilde{A}, \tilde{B})$ defines a civil, concurrent, total, P-operator on its associated arena $[\![A^\bot \parr B]\!]$. Given $\mathcal{R} \in \mathsf{HypGraph}(A)$, (or more generally, any morphism $\mathcal{Q} \in \mathsf{SimpleHypGraph}(\tilde{A})$), then $\tilde{\mathcal{R}}$ (resp $\mathcal{Q}$) projects naturally into a set of maximal positions of the domain $[\![A]\!]$. We define the corresponding closure operator σ_R on $[\![A]\!]^\top$, the lattice completion of $[\![A]\!]$, as follows:

$$\sigma_{\mathcal{R}}(x) = \bigsqcap\{y \in \tilde{\mathcal{R}} \mid y \sqsupseteq x\}$$

Similarly, given a &-resolution Ψ on A, one can associate a closure operator to it. It can be described syntactically, as being the O-operator that chooses the left or right branch of each occurrence of & following Ψ. Similarly, given the set of maximal positions y of $[\![A]\!]^\top$ that correspond to those additive resolutions that are on Ψ (simply written y on Ψ), it can be given the following description:

$$\tau_\Psi(x) = \bigsqcap\{y \mid y \sqsupseteq x, y \text{ on } \Psi\}$$

Then let us consider a relation $\mathcal{R} \in \mathsf{SimpleHypGraph}(A, B)$ and $\mathcal{Q} \in \mathsf{SimpleHypGraph}(B, C)$, and a pair of &-resolutions (Ψ_A, Ψ_C) of $(A^\bot, C)$. Then in order to prove composition, one must prove that the following element:

$$\langle \tau_{\Psi_A} ;_A \sigma_{\mathcal{R}}, \sigma_{\mathcal{Q}} ;_C \tau_{\Psi_C} \rangle$$

is maximal. This element then corresponds to an additive resolution such that both relations have elements in this additive resolution. First, we prove the following property.

Proposition 7.46. *Let $\mathcal{R} \in \mathsf{SimpleHypGraph}(A, B)$, and Ψ_A a $\oplus$-resolution of A. Then there exists $Q \in \mathsf{SimpleHypGraph}(I, B)$ such that $\Psi_A \upharpoonright \mathcal{R} \upharpoonright B = Q$. Furthermore, given $\sigma_{\mathcal{R}}, \tau_{\Psi_A}$, and σ_Q their closure-operator counterparts, $\tau_{\Psi_A};_A \sigma_{\mathcal{R}} = \sigma_Q$*

Proof. Let $\mathcal{R} \in \mathsf{SimpleHypGraph}(A, B)$, Ψ_A a $\oplus$-resolution of A. Then for all &-resolutions Φ_B of B, there is a unique element $x \in \mathcal{R}$ such that x on $\Psi_A\Phi_B$. So for every &-resolution of Φ_B, there is an element $x \in (\Psi_A \upharpoonright \mathcal{R} \upharpoonright \Phi_B)$, and this element is unique. So $\mathcal{R} \upharpoonright \Psi_A \upharpoonright B$ is $(P1)$. Furthermore, one needs to prove that for every subset $Q \subseteq (\Psi_A \upharpoonright \mathcal{R}) \upharpoonright B$ then $Q \in \Gamma(B)$. Either there is only one element in $\Psi_A \upharpoonright Q \upharpoonright A$, and therefore this element is both incoherent and coherent. Then as $\Psi_A \upharpoonright Q \subseteq \Gamma(A^\perp \parr B)$, this entails $(\Psi_A \upharpoonright Q) \upharpoonright B \in \Gamma(B)$. Either there are two or more elements. In this case, they are in the same &-resolution of $A^\perp$, but on different $\oplus$-ones. This would entail $(\Psi_A \upharpoonright Q) \upharpoonright A^\perp \in \Gamma^{\perp *}(A^\perp)$, and therefore $(\Psi_A \upharpoonright Q) \upharpoonright B \in \Gamma^*(B)$.

One can note that the only possible elements of $(\sigma^\bullet_{\mathcal{R}} \upharpoonright A) \cap \tau^\bullet_{\Psi_A}$ are the maximal elements that correspond to the $\Psi_A \upharpoonright \mathcal{R}$. Therefore, $\tau_{\Psi_A};_A \sigma_{\mathcal{R}} = \sigma_{(\Psi_A \upharpoonright \mathcal{R}) \upharpoonright B} = \sigma_Q$, where $Q = \Psi_A \upharpoonright \mathcal{R} \upharpoonright B \in \mathsf{SimpleHypGraph}(I, B)$. □

And finally, we prove the required property.

Proposition 7.47. *Let $\mathcal{R} \in \mathsf{SimpleHypGraph}(A)$, $Q \in \mathsf{SimpleHypGraph}(A^\perp)$. Then $\langle \sigma_{\mathcal{R}}, \tau_Q \rangle$ is maximal, that is $\mathcal{R} \cap Q \neq \emptyset$*

To establish this proposition given two simple hypercoherences as above, we prove that the interaction between their associated concurrent operators never reach a deadlock. This follows from the following lemma.

Lemma 7.48. *Given $\mathcal{R} \in \mathsf{SimpleHypGraph}(A)$, and $\sigma_{\mathcal{R}}$ the associated closure operator, then $\sigma_{\mathcal{R}}$ satisfies the following:*

- *$\sigma_{\mathcal{R}}$ is positive.*
- *$\sigma_{\mathcal{R}}$ is total: $\forall x \in \sigma^\bullet$, either x has polarity P, or N.*

In particular, being positive entails that σ is *civil*, meaning that for every counter-strategy τ, $\langle \sigma; \tau \rangle \neq \top$. This was established in lemma 7.10. We call the second condition totality as it will , as proven below, entails that the element reached through a interaction with a counter-opponent is maximal, and hence implies totality as defined in the original paper on concurrent games [10]. Assuming the lemma, we prove the proposition.

Proof. Let us note that if we reach a N-position, then this position is maximal and the proposition is proven. As σ and τ are O and P-closure operators respectively, $\langle \sigma, \tau \rangle \neq \top$. Furthermore, as σ is total, $\langle \sigma, \tau \rangle$ is either a P or N position. Similarly, as τ is total (but negative), $\langle \sigma, \tau \rangle$ is either a O or N-position. Hence $\langle \sigma, \tau \rangle$ is a N-position, and hence it is maximal. □

We now endeavour to prove the lemma.

Proof. We start by proving positivity. Given a partial additive resolution x seen as an element of the domain, and considering its sub-partial &-resolution $\mathsf{neg}(x)$, as $\mathcal{R}$ is $(P1)$, there are some elements of $\mathcal{R}$ above every &-resolution and in particular above $\mathsf{neg}(x)$. Hence $\sigma_{\mathcal{R}}(\mathsf{neg}(x)) \neq \top$. Given any element $x \in [\![A]\!]$, either x is incompatible with $\mathcal{R}$, that is, $\sigma_{\mathcal{R}}(x) = \top$, or there are some positions of $\mathcal{R}$ above it. We write $A \upharpoonright x$ for $A \upharpoonright \Psi$ where Ψ is the partial additive resolution corresponding to x. Let us consider the set of &-occurrences of $A \upharpoonright x$. As $\mathcal{R}$ is $(P1)$, for each partial &-resolution of it, there is an element on it. In particular, each O-prime p starting from x correspond to selecting the direct left or right sub-formula of a direct sub-formula $C \,\&\, D$ of $A \upharpoonright x$. Consequently, there is a position $y_1 \in \mathcal{R}$ such that $y_1 \upharpoonright C \neq \emptyset$ and a position $y_2 \in \mathcal{R}$ such that $y_2 \upharpoonright D \neq \emptyset$. Hence $\mathsf{p} \not\sqsubseteq y_1 \sqcap y_2$ and $\sigma_{\mathcal{R}}$ will not bring it. So, as a result, $x \overset{P}{\twoheadrightarrow} \sigma(x)$, establishing the second point of positivity. The proof that $\sigma_{\mathcal{R}}(\mathsf{neg}(x)) \sqcup \mathsf{pos}(x) = \sigma_{\mathcal{R}}(x)$ follows the same lines, establishing that the P-primes from x brought by $\sigma_{\mathcal{R}}$ depend solely on $\mathsf{neg}(x)$.

Now, let us focus on totality. Totality is immediate once we notice the following property: O corresponds to strict incoherence, P to strict coherence, and N to a position that is both coherent and incoherent. More precisely, given a set of maximal positions $\{x_i\}$ such that $\bigsqcap\{x_i\} : O$, (respectively P, N), then $\{x_i\}$ is strictly hyper-incoherent (respectively strictly hypercoherent, respectively hyper-coherent and hyper-incoherent, that is, a singleton). That is, there is an equivalence between the polarity of the intersection and the hyper-coherence.

This is proven by induction on the formula A. If $A = X$, then there is only a N-position, that is both coherent and incoherent. If $A = B \,\&\, C$, then if $\bigsqcap\{x_i\}$ is in B, or C, the property follows by induction. Otherwise it is $\perp_A$, and hence it implies that there are some x_i in B and some in C. So by definition of hypercoherence $\{x_i\} \in \Gamma^*(A)$, and $\perp_A : P$ as expected. A similar reasoning works for the $\oplus$-case. For the multiplicative cases, we first deal with $\otimes$. Let us consider a set $\{p_i\}$ such that the intersection in $A \otimes B$ is P. It means that $\bigcap\{x_i\} \upharpoonright A = \bigcap\{x_i \upharpoonright A\}$ is P or N, and similarly for B, with at least one of the both being P. By induction hypothesis, $\{p_i \upharpoonright A\} \in \Gamma(A)$, and $\{p_i \upharpoonright B\} \in \Gamma(B)$, with at least one of them in Γ^*. Henceforth, $\{x_i\} \in \Gamma^*(A \otimes B)$. The other cases are proven similarly.

Therefore, given a partial additive resolution x, either $\sigma(x)$ is incompatible with x, in which case $\sigma(x) = \top$. Or $\sigma(x)$ corresponds to the intersection of the set of elements of $\mathcal{R}$ that are above x. And these ones are, by definition, hypercoherent. Hence $\sigma(x) : P$ if there is more than one element above it, and $\sigma(x) : N$ is there is exactly one. □

Finally, we conclude the section with the desired property to prove that $\mathsf{HypGraph}$ forms a category.

Proposition 7.49. *Let $\mathcal{R} \in \mathsf{HypGraph}(A, B)$, $\mathcal{Q} \in \mathsf{HypGraph}(B, C)$, then given Ψ_A a $\oplus$-resolution of A, Ψ_C a &-resolution of C, there exists a unique additive resolution Φ_B such that $\mathcal{R} \upharpoonright \Psi_A\Phi_B \neq \emptyset$, and $\mathcal{Q} \upharpoonright \Psi_C\Phi_B \neq \emptyset$.*

That is, $\mathcal{R} \upharpoonright \Psi_A\Phi_B$ is a partial nominal relation, and so is $Q \upharpoonright \Phi_B\Psi_C$. The proof of this property is a simple combination of all the results of this section. The unicity simply comes from noticing that if there were two, then there would be two maximal elements in the set $\sigma^\bullet_{\tilde{\mathcal{R}}} \cap \tau^\bullet_{\tilde{Q}}$, which would contradict $\langle \sigma_{\tilde{\mathcal{R}}}, \tau_{\tilde{Q}} \rangle$ being minimal.

7.3.3 The category HypGraph

The goal of this section is to prove that every morphism of MALL can be soundly interpreted in HypGraph. We prove that HypGraph is a sound model of MALL by establishing that it forms a star-autonomous category with products.

Lemma 7.50. HypGraph *forms a category.*

Proof. First, we shall explicit what is the identity. A &-resolution $A \multimap A = A^\perp \parr A$, is the data of a &-resolution Φ on A, and a &-resolution Ψ on $A^\perp$, that is a $\oplus$-resolution on A. Let us note that a &-resolution together with a $\oplus$-resolution give rise to a unique additive-resolution, and this additive resolution is on these both partial resolutions. The identity morphism on this &-resolution is the identity partial nominal relation on the unique additive resolution coming from Φ, Ψ on A.

We prove that this morphism indeed acts as the identity. We do the proof only for the right-action, the left-one being dealt with symmetrically. We consider a morphism $\mathcal{R} : A \to B_1$, and the identity relation $\mathsf{id}_B : B_1 \to B_2$. We shall prove that $\mathcal{R}; \mathsf{id}_B = \mathcal{R}$. Let us pick a &-resolution Ψ on B_2, and a $\oplus$ resolution Υ of A. The set of $\oplus$-resolutions $\{\Phi_i\}$ on $B_1^\perp$ such that $\mathsf{id}_B : B_1 \upharpoonright \Phi_i \to B_2 \upharpoonright \Psi \neq \emptyset$ is exactly $\{\Psi\}$ by definition. Therefore, the composition $\Upsilon \upharpoonright \mathcal{R}; \mathsf{id}_B \upharpoonright \Psi$ is equal to $\Upsilon \upharpoonright \mathcal{R} \upharpoonright \Psi; \Psi \upharpoonright \mathsf{id}_B \upharpoonright \Psi$. There is a unique $\oplus$-resolution Λ of $A \upharpoonright \Upsilon \multimap B_1 \upharpoonright \Psi$ such that $\Upsilon \upharpoonright \mathcal{R} \upharpoonright \Psi$ is on $A \upharpoonright \Upsilon\Lambda_A \multimap B \upharpoonright \Psi\Lambda_B$. Completing arbitrarily Λ_B as a $\oplus$-resolution of B, then $\Lambda_B \upharpoonright \mathsf{id}_B \upharpoonright \Psi : B_1 \upharpoonright \Lambda_B \to B_2 \upharpoonright \Psi$ is the identity partial nominal relation $\Lambda_B\Psi \upharpoonright \mathsf{id} \upharpoonright \Lambda_B\Psi : B_1 \upharpoonright \Lambda_{B_1}\Psi \to B_2 \upharpoonright \Lambda_B\Psi$. Therefore, we have the following sequence of equations:

$$
\begin{aligned}
\Upsilon \upharpoonright \mathcal{R}; \mathsf{id}_B \upharpoonright \Psi &= \Upsilon \upharpoonright \mathcal{R} \upharpoonright \Psi; \Psi \upharpoonright \mathcal{R} \upharpoonright \Psi \\
&= \Upsilon\Lambda_A \upharpoonright \mathcal{R} \upharpoonright \Psi\Lambda_B; \Lambda_B\Psi \upharpoonright \mathsf{id} \upharpoonright \Lambda_B\Psi \\
&= \Upsilon\Lambda_A \upharpoonright \mathcal{R} \upharpoonright \Psi\Lambda_B \\
&= \Upsilon \upharpoonright \mathcal{R} \upharpoonright \Psi
\end{aligned}
$$

Therefore, on each &-resolution $\mathcal{R}$ and $\mathcal{R}; \mathsf{id}_B$ agree. As $\mathcal{R}$ is perfectly determined by its nominal partial relation on each &-resolution, following the (P1) condition, we conclude that $\mathcal{R} = \mathcal{R}; \mathsf{id}$.

We now deal with composition more generally. We need to prove that given two partial nominal relations $\mathcal{R} : A \to B$ and $Q : B \to C$ satisfying (P1), namely, that on each &-

resolution they give rise to a non-empty partial nominal relation, then so does their composite. This was proven in the previous section. Furthermore, we need to prove that $\mathcal{R}\,;\,Q$ is a clique (that is, each subset of it is hypercoherent). This follows from the definition of $\Gamma(A \multimap B)$.

□

Lemma 7.51. HypGraph *is star-autonomous with products.*

The star-autonomy follows from the one of hypercoherences and the one of ParNomRel. The product follows from $A \,\&\, B$ being a product of A, B in the category of hypercoherences.

Therefore, HypGraph forms a sound model of MALL. We will establish some properties of the morphisms of HypGraph in the next sections. First, we prove strong softness. This allows us to prove that HypGraph is fully compete for a fragment of MALL, and allows us to map a proof-structure to each morphism of HypGraph.

7.3.4 Strong softness

In the next two sections, we would like to characterise precisely how the hypercoherence condition shapes partial nominal relations. Our first step is to prove that the morphisms of HypGraph are strongly soft. This is what we target in this section. We will see in the next section that this will allow us to prove a full-completeness result for HypGraph for a lax fragment of MALL.

Definition 7.52. *Suppose $l_1, ..., l_n$ are literals. We say that a morphism $\mathcal{R}$,*

$$\mathcal{R} : I \to l_1 \parr ... \parr l_m \parr (A_{1,1} \oplus A_{1,2}) \parr ... \parr (A_{n,1} \oplus A_{n,2})$$

*is **strongly soft** if it exists k that is such that $1 \leq k \leq n$ such that $\mathcal{R}$ factors through one of the injection $\mathsf{in}_{k,j} : A_{k,j} \to A_{k,1} \oplus A_{k,2}$, $j = 1, 2$. That is:*

$$\mathcal{R} = \mathcal{R}'\,;\,\mathsf{id}_{l_1} \parr ... \parr \mathsf{id}_{l_n} \parr \mathsf{id}_{A_{1,1}\oplus A_{1,2}} \parr ... \parr \mathsf{in}_{k,j} \parr ... \parr \mathsf{id}_{A_{n,1}\oplus A_{n,2}}.$$

where $\mathcal{R}' : I \to l_1 \parr ... \parr l_m \parr (A_{1,1} \oplus A_{1,2}) \parr ... \parr A_{k,j} \parr ... \parr (A_{n,1} \oplus A_{n,2})$.

Proposition 7.53. *Every morphism $\mathcal{R} : 1 \to l_1 \parr ... \parr l_m \parr (A_{1,1} \oplus A_{1,2}) \parr ... \parr (A_{n,1} \oplus A_{n,2})$ in* HypGraph *is strongly soft.*

Proof. We assume that $\mathcal{R}$ is not strongly soft and prove a contradiction. This means that for every $A_{i,1}$, and $A_{i,2}$, there are two elements $x_{i,1}, x_{i,2} \in \mathcal{R}$ such that $x_{i,1} \upharpoonright A_{i,1} \neq \emptyset$, and $x_{i,2} \upharpoonright A_{i,2} \neq \emptyset$. Furthermore, we pick $x_{i,1}, x_{i,2}$ such that $\forall j$, if $1 \leq j \leq m$ then we have $x_{i,1} \upharpoonright l_j = x_{i,2} \upharpoonright l_j = \bot$. Let us consider the set $\mathcal{R} \supseteq s = \cup_i\{x_{i,1}, x_{i,2}\}$, then for every $\parr$ block $\Delta_i = A_{i,1} \oplus A_{i+2}$, $s \upharpoonright \Delta_i \supseteq \{x_{i,1}, x_{i,2}\} \upharpoonright \Delta_i$ and therefore is strictly incoherent. Now among the literals, we have $s \upharpoonright l_1, ..., l_m \in \Gamma(l_1, ..., l_m) \cap \Gamma^{\perp}(l_1, ..., l_m)$, that is, they project onto a singleton. Therefore, $\forall l_i . s \upharpoonright l_i \in \Gamma^{\perp}(l_i)$, and $\forall \Delta_i . s \upharpoonright \Delta_i \in \Gamma^{\perp *}(\Delta_i)$. Hence $s \in \Gamma^{\perp,*}(l_1 \parr ... \parr l_m \parr (A_{1,1} \oplus A_{1,2}) \parr ... \parr (A_{n,1} \oplus A_{n,2}))$.

Hence $\exists s \supseteq \mathcal{R}$ such that s is strictly incoherent. This is a contradiction, and $\mathcal{R}$ factors through one of the $\oplus$. □

7.3.5 Full completeness for PALL$^-$

Hypercoherences obey a rule that is rejected by linear logic, namely the mix rule. Although it does not hold in linear logic, for this section it is helpful to use it. We say that a categorical model accepts the mix rule if for all objects A, B there is a morphism $\mathsf{mix}_{A,B} : A \otimes B \to A ⅋ B$. Logically, the mix rule behaves as follows :

$$\frac{\vdash \Gamma \qquad \vdash \Delta}{\vdash \Gamma, \Delta}\ \mathsf{MIX}$$

We write PALL$^-$ for the fragment of MALL$^-$ without the $\otimes$-rule and PALL$^-$ + MIX for this fragment extended with the MIX-rule. The first full completeness theorem we establish is for the PALL$^-$ + MIX fragment.

Theorem 7.54. HypGraph *is fully complete for* PALL$^-$ + MIX.

The key of the proof is the softness of hypercoherences, that we proved in the above section 7.3.4. With this in mind, we prove the theorem.

Proof. We do the proof for HypGraph$(I, [\![\Gamma]\!])$, where Γ is a sequent, by induction on the number of additives. If there is none, then as it is a partial nominal relation, it is of the form:

$$\vdash X_1, X_1^{\perp}, X_2, X_2^{\perp}, ..., X_n, X_n^{\perp}$$

And this can be obtained as the denotation of a sequence of axioms and MIX-rules.

If there is a conclusion C that is of the form $C = C_1 \,\&\, C_2$. That is, the sequent can be written:

$$\vdash \Gamma, C_1 \,\&\, C_2$$

Then the let us write $\mathcal{R}_1$ for the elements $x \in \mathcal{R}, x \upharpoonright C_1 \neq \emptyset$ and $\mathcal{R}_2$ for the elements $x \in \mathcal{R}, x \upharpoonright C_2 \neq \emptyset$. Then $\mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset$ and $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$. Furthermore, $\mathcal{R}_1, \mathcal{R}_2$ are morphisms made out of partial nominal relations, are cliques (any finite subset of them is hypercoherent), and satisfy the (P1) property on &-resolutions. Therefore, $\mathcal{R}_1 \in$ HypGraph(Γ, C_1), and similarly for $\mathcal{R}_2$. Therefore, $\mathcal{R}$ is the denotation of the following proof, where $[\![\pi_1]\!] = \mathcal{R}_1$, $[\![\pi_2]\!] = \mathcal{R}_2$. :

$$\frac{\dfrac{\pi_1}{\vdash \Gamma, C_1} \qquad \dfrac{\pi_2}{\vdash, \Gamma, C_2}}{\vdash \Gamma, C_1 \,\&\, C_2}\ \&$$

Similarly, if $\mathcal{R}$ is a morphisms of $\mathsf{HypGraph}(1,\Gamma)$, and there is a $\parr$ at the bottom-level in the sequent, we just can forget about it, since it will not change the denotation of Γ.

$$\frac{\vdash \Gamma, A, B}{\vdash, \Gamma, A \parr B}\ \parr$$

By applying the two above rules, we can progressively decompose the relation $\mathcal{R}$ by disposing of the &-connectives and the $\parr$-connectives at the bottom level of the sequent. We then obtain a relation $\mathcal{R}$ that is in the denotation of a sequent of the form:

$$\vdash l_1, ..., l_m, A_{1,1} \oplus A_{1,2}, ..., A_{n,1} \otimes A_{n,2}$$

As $\mathsf{HypGraph}$ is strongly soft, there is a i such that $\mathcal{R}$ factors through one of the injection, that is, $\mathcal{R}$ is derived from the following $\oplus$-rule, where the $\oplus$ is either a $\oplus_1$ or a $\oplus_2$. We present the $\oplus_1$ case above:

$$\frac{\vdash l_1, ..., l_m, A_{1,1} \oplus A_{1,2}, ..., A_{i,1}, .., A_{n,1} \oplus A_{n,2}}{\vdash l_1, ..., l_m, A_{1,1} \oplus A_{1,2}, ..., A_{i,1} \oplus A_{i,2}, .., A_{n,1} \oplus A_{n,2}}\ \oplus_1$$

We now have dealt with all the cases, and are able to interpret $\mathcal{R}$ as the denotation of a proof of PALL^-. This concludes the full completeness proof. □

7.3.6 The category ChuHypGraph

It has been studied and proven that hypercoherences themselves are not strong enough to provide a fully complete of MALL [87]. Their main flaw is that they allow the MIX-rule, just as the coherence spaces. This leads to unexpected problems when studying their proof structures. To solve that, we will refine HypGraph by adding the Chu-conditions. This way, the top level morphisms are fully complete for MLL, so they reject the mix rule.

ChuHypGraph is defined the same way as the category HypGraph, except that the Chu-conditions have been added. We define the $\oplus$ of two objects of ChuLinNom to be:

$$A \oplus B = \{\mathsf{inl}(\mathbb{D}_A) \uplus \mathsf{inr}(\mathbb{D}_B), \mathsf{inl}(P_A) \uplus \mathsf{inr}(P_B), \mathsf{inl}(O_A) \uplus \mathsf{inl}(O_B), \mathsf{inl}(T_A) \uplus \mathsf{inr}(T_B), \mathsf{inl}(T_A^\perp) \cup \mathsf{inr}(T_B^\perp)\}.$$

where $\mathbb{D}_A$, $\mathbb{D}_B$ are their underlying polarised coherence domains. A **Chu-coherence polarised object** is a finite sum of polarised coherence objects equipped with the Chu-structure. We call ChuLinNomA the class of Chu qualitative polarised objects obtained from $[\![X]\!]_{\mathsf{ChuLinNom}}$ and the operations $\otimes, \oplus$ and $(.)^\perp$.

Definition 7.55. *A **hypercoherence** on an object of* ChuLinNomA *(called Chu-hypercoherence space) is a couple* $(A, \Gamma(A))$ *where* $A \in$ ChuLinNomA *and* $\Gamma(A) \subseteq \mathcal{P}^*_{\mathsf{fin}}(\mathbb{D}_A)$ *is its set of hypercoherence.*

We furthermore write $\Gamma^*(X)$ for the subset of $\Gamma(X)$ of elements that are not singletons.

$$\begin{aligned}[\![X]\!]_{\mathsf{ChuHypGraph}} &= ([\![X]\!]_{\mathsf{ChuLinNom}}, \Gamma^*(X) = \{x \uplus \bot \mid x \subseteq \mathcal{P}_{\mathsf{fin},>1}(\mathbb{A}_X)\}) \\ &= ([\![X]\!]_{\mathsf{HypGraph}} + \text{Chu-conditions})\end{aligned}$$

where we remind that we denote $\mathcal{P}_{\mathsf{fin},>1}$ the set of finite sets of cardinal more than 1. To each formula of MALL^- a Chu-hypercoherence space can be assigned by repeated applications of the $\otimes, (.)^\perp$ and $\oplus$ rules.

Definition 7.56. *The category* ChuHypGraph *is defined as follows:*

- *The objects* $(A, \Gamma(A))$ *are Chu-hypercoherence spaces built by induction from* $[\![X]\!]_{\mathsf{ChuHypGraph}}$ *using* $\otimes, \oplus$ *and* $(.)^\perp$.
- *The morphisms* $A \to B$ *are those relations* $\mathcal{R} : \mathbb{D}_{A\multimap B}$ *such that:*
  1. *(P1) For each* &*-resolution* Ω *of* $A \multimap B$*, there is an additive resolution* Ψ *on* Ω *such that* $\mathcal{R} \upharpoonright \Psi \in \mathsf{ChuLinNom}(A \multimap B \upharpoonright \Omega)$
  2. $\forall s \in \mathcal{P}^*_{\mathsf{fin}}(\mathcal{R})$, $s \in \Gamma(A \multimap B)$.

Proposition 7.57. ChuHypGraph *is a sub-semi-star-autonomous category of* HypGraph.

Proof. The composition of morphisms follows from the ones of HypGraph and the ones of ChuLinNom. Furthermore, the tensor product, the $\oplus$, and the $(.)^\perp$ are similar to those of HypGraph. □

There is an obvious injection ChuHypGraph → HypGraph, therefore the properties we established about morphisms of HypGraph lift straightforwardly to ChuHypGraph. For instance, the morphisms of ChuHypGraph are fully complete for PALL^-, since they originate from morphisms of HypGraph but reject the MIX-rule.

7.3.7 Full completeness

The goal of this section is to established that ChuHypGraph is fully complete for MALL^-. The proof we develop here is a copycat of the proof provided in [16] by Blute, Hamano and Scott for their category of double glued hypercoherences. The whole idea is that hypercoherence encapsulates well the additive conditions, but fails at the multiplicative level by allowing the MIX-rule. Therefore, we shall rely on some additive constructions to make them well-behaved at the multiplicative level. Just as we use ChuLinNom morphisms in order to be fully complete at the multiplicative level, their use of the double glueing allows them to reject the MIX-rule.

One of the key elements of the proof of [16] is the full-completeness arguement for PALL^-+ MIX established for HypGraph . Indeed, this enables us, as we will explain below, to assign to each morphism of HypGraph (and therefore, each morphism of ChuHypGraph) a Girard MALL^- proof structure. Another direction was taken in [10] with concurrent games, where a proof-structure was built on top of the domain on which the strategies were acting. It is however a bit stretchy to adapt the proof of [10] in our case (even if the hypercoherence condition gives rise to concurrent games), since their strategies encapsulate two kinds of dependencies within one unifying framework: the ones between the $\oplus$ and the &, and the ones between the litterals of oppositve polarities arising from axiom-links. However, in our case, the hypercoherence condition allows us to capture only the dependencies between $\oplus$ and & , and the dependencies arising fom axiom-links are forced upon by an additive condition (that the above morphisms belong in ParNomRel). On the other hand, the proof of [16] fits well our morphisms.

We expose in broad terms the plan for the full completeness proof. Let us consider a morphism $\mathcal{R} : I \to A$ in ChuHypGraph, seen as a morphism of HypGraph. Then we replace it, using numerous time the mix-rule, by a morphism $I \to A \xrightarrow{\text{mix}} .. \xrightarrow{\text{mix}} B$, where B is a formula of PALL^-. Then as HypGraph is fully complete, this give rise to a proof of $\mathsf{PALL}^- + \mathsf{MIX}$. This one generates one (or several) Girard proof structure(s) (whose definition is given in the appendix 9.1.1.1). Now, we notice that in a Girard proof structure, occurrences of $\otimes/\parr$ can be switched (more on this on appendix 9.6). That is, given a Girard proof structure with some occurrences of $\otimes$, if we replace them by $\parr$, it remains a Girard proof structure, and reversely. Therefore, by backtracking the mix, we can establish that there is one (or several) proof-structure(s) corresponding to the original morphism $\mathcal{R} : I \to A$.

Now, the goal is to prove that these proof structures satisfy the criteria for being proof nets. In order to do that, we establish that we can focus on certain proof structures that are canonical (whose definitions are provided in 9.1.1.4), and such that manipulations on morphisms translate well to manipulations on the proof structures (more on that is developed in appendix Section 9.1.1.5). Using them, we can transform the morphism, and the proof structure, to an appropriate form (presented in Section 9.1.4 of the appendix). We give all the necessary details in the Appendix 9.1. Although the necessary work done in the Appendix relies on the morphisms belonging in ChuHypGraph (notably, as ChuHypGraph is fully-complete for MLL, the proof structures are connected, and hence we can conclude that the possible cycles have a particular form), for the final argument we forget their Chu-structure, looking at them as morphisms of HypGraph, in order to be able to apply to them a series of mix-rule.

Assuming that there is a proof structure coming from a morphism of ChuHypGraph with a cycle, we get, after many steps, to the final argument (presented in 9.1.4) where we have obtained a morphism of HypGraph encoding a proof structure of the type :

$$\vdash F_1, ..., F_n$$
$$\text{and } F_i = \alpha_{i,m} \otimes (B_1^i \,\&\, B_2^i), N_i[\alpha_{i+1,1}^\perp, \alpha_{i+1,1}^\perp], \alpha_{i+1,1} \otimes \alpha_{i+1,2}^\perp, ..., \alpha_{i+1,m-1} \otimes \alpha_{i+1,m}^\perp, \Xi_i$$

Where $\Xi_i = E_{i11} \oplus E_{i12}, ..., E_{im-1} \oplus E_{im}, l_{i1}, ..., l_{ir}$ and N_i is a context made out exclusively of literals, $\oplus$ and $\parr$, and such that the two $\alpha_i^\perp$ do not appear simultaneously in an $\oplus$- resolution. That is, $N_i = M_i[C_{i,1}[\alpha_i^\perp] \oplus C_{i,2}[\alpha_i^\perp]]$: the first $\alpha_i^\perp$ appears in C_1, and the second in C_2. Furthermore, the linking is such that the choice of B_1 by the opponent entails the choice for proponent of an additive resolution that selects the left $\alpha_i^\perp$, and respectively for B_2 and the right one, while otherwise letting the the other links that appear in the cycle (defined later) intact. Locally, the linking is as follows:

$$\mathsf{ax}_{i,1}$$

$$\alpha_{i,m-1} \otimes \alpha_{i,m}^{\perp} \quad \alpha_{i,m} \otimes (B_{i1} \&_i B_{i2}) \quad M_i[C_{i,1}[\alpha_{i,1}^{\perp}] \oplus C_{i,2}[\alpha_{i,1}^{\perp}]] \quad \alpha_{i,2} \otimes \alpha_{i,2}^{\perp} \quad \cdots \quad \alpha_{i,m-1} \otimes \alpha_{i,m}^{\perp} \quad \Xi \quad F_{i+1}$$

$$\mathsf{ax}_{i,2}$$

and therefore leads to a cycle as displayed in Figure 7.2, where the cycle is displayed in red, and the axiom-links that are not part of the cycle in black. So one can see that there are 2^n additive resolutions on which the cycle belongs (by switching the n $\&_i$s).

We define G_i to be the part of the sequent:

$$G_i = (B_1^i \&_i B_2^i), \alpha_{i+1,1} \otimes \alpha_{i+1,2}^{\perp}, \alpha_{i+1,2} \otimes \alpha_{i+1,3}^{\perp}, \alpha_{i+1,n-1} \otimes \alpha_{i+1,n}^{\perp}, \alpha_{i_1,n}, \Xi_i$$

We assume for contradiction that there is a nominal partial hypercoherent relation $\mathcal{R}$ that encodes this proof-structure. We consider a set S of n^3 elements $S = \{(x)_l\}$ of $\mathcal{R}$, indexed by lists $l = (l_1, ..., l_n)$ such that $l_i \in \{1, 2, 3\}$. Furthermore, these elements are such that:

- $x_{(....,l_i,...)}$ is on the &-additive resolution that picks $B_{i,1}$ (and hence picks the $\mathsf{ax}_{i,1}$) if $l_i \in \{1, 3\}$
- $x_{(....,l_i,...)}$ is on the &-additive resolution that picks $B_{i,2}$ (and hence picks the $\mathsf{ax}_{i,2}$) if $l_i = 2$

All these $x's$ will be $\perp$ on all literals, except precisely those that appear in the cycle, that are those written $\alpha_{i,j}$. We refer to Figure 7.2 for a better understanding. We write $x_{...,i:k,...}$ to indicate that we speak about an element whose index-list is such that its i^{th} position is k. The elements x's are chosen such that for every two lists l, l' that agree on the i^{th} element, $l_i = l'_i$ then $x_l \upharpoonright \alpha_{i,j} = x_{l'} \upharpoonright \alpha_{i,j}$. Therefore $x_{....,i:k,....} \upharpoonright \alpha_{i,j}$ is a singleton and we write $x_{i:k} \upharpoonright \alpha_{i,j}$.

In this paragraph, we will be working within the context G_i, and will simply write $x_{i:k}$, to speak broadly about all those elements $x_l \upharpoonright G_i$ such that $l_i = k$. Formally, $x_{i:k} = \{x_l \mid l_i = k\} \upharpoonright G_i$, and we can write $x_{i,k} \upharpoonright A$ for any sub-formula A of G to design the projection of this family to A. The family of sets $x_{i:1}, x_{i:2}, x_{i:3}$ will be designed such that $x_{i:1} \cup x_{i:2} \cup x_{i:3} \upharpoonright \alpha_{i,n} \in \Gamma^{\perp,*}(\alpha_{i,n})$. Furthermore, we remind that $x_{i:1}, x_{i:3}$ choose the left &-resolution on $\&_i$, (and hence, the left $\oplus$ one as well on M_i) and $x_{i:2}$ the right one. Furthermore, the only literals l of G_i such that $x_{i:1}, x_{i:2}, x_{i:3} \upharpoonright l \neq \perp$ are related by axiom-links to other literals appearing in G_i. So the elements of $x_{i:1}$ will be of the following form:

$$\perp_{B_{i,1}}, \perp_{M_i}[\perp_{C_{i,1}}[a_{i,1}]], a_{i,1} \otimes a_{i,2}, ..., a_{i,m-2} \otimes a_{i,m-1}, a_{i,m-1} \otimes a_{i,m}, a_{i,m}, \perp_{\Xi_i} .$$

where we write $\perp_{B_{11}}$ for a minimal $\perp$ element of $[\![B_{11}]\!]$, and $\perp_M[\perp_{C_1}[a_{11}]]$ for an element of $[\![M]\!]$ that is $\perp$ everywhere except in $\alpha_{i,m}$ where it is a_{11}, and picks the $\oplus$-resolution that chooses C_1. We recall that $x_{i:2}$ is picking the right resolution, and furthermore we settle to make them select different names than $x_{i:1}$. That is, the elements of $x_{i:2}$ are of the following form, where

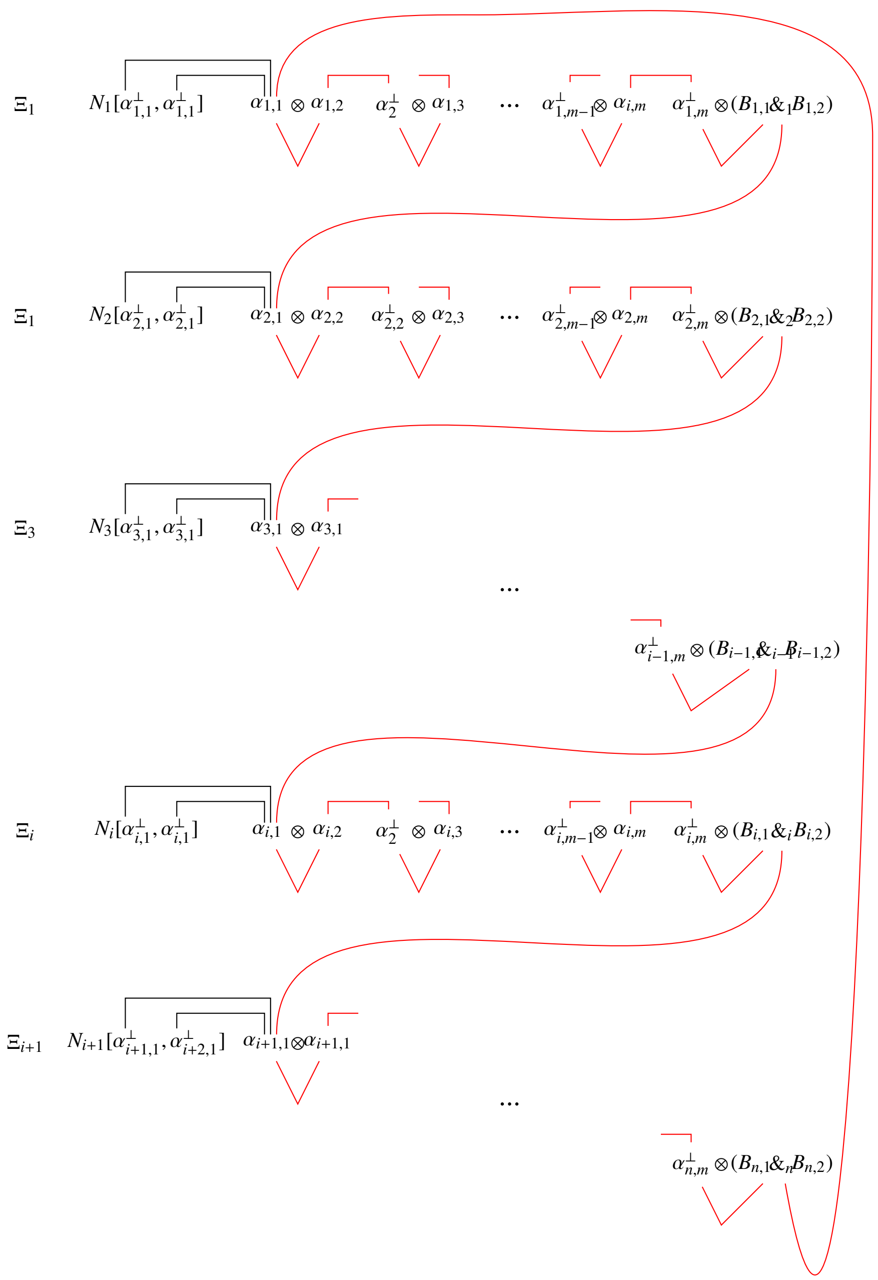


Figure 7.2: Global cycles in the final proof-structure

$\forall j.1 \leq j \leq m, b_{i,j} \neq a_{i,j}$.

$$\bot_{B_{i2}}, \bot_{M_i}[\bot_{C_{i,2}}[b_{i,1}]], b_{i,1} \otimes b_{i,2}, ..., b_{i,m-2} \otimes b_{i,m-1}, b_{i,m-1} \otimes b_{i,m}, b_{i,m}, \bot_{\Xi_i}$$

Finally, the third elements $x_{i:3}$ are designed to make the whole set $x_{i:1} \cup x_{i:2} \cup x_{i:3}$ strictly incoherent on all direct sub-formulas of G_i, except in $B_{i,1} \& B_{i,2}$ where we are unable to prevent it from being coherent. Formally, if $\alpha_{i,k}$ is positive, then we take $c_{i,k} \in [\![\alpha_{i,k}]\!]$ to be a third name. That is, writing X for the atomic formula such that $\alpha_{i,k} = X$, then $c_{i,k} \in \mathbb{A}_X \setminus \{a_{i,k}, b_{i,k}\}$. Otherwise, we take $c_{i,k} = \bot$. That way, $\{a_{i,k}, b_{i,k}, c_{i,k}\} \in \Gamma^{\bot,*}(\alpha_{i,k})$. Then the elements of $x_{i:3}$ are as follows:

$$\bot_{B_{i:1}}, \bot_{M_i}[\bot_{C_{i,1}}[c_{i,1}]], c_{i,1} \otimes c_{i,2}, ..., c_{i,m-2} \otimes c_{i,m-1}, c_{i,m-1} \otimes c_{i,m}, c_{i,m}, \bot_{\Xi_i}$$

In this paragraph, we study the local coherences of the set $S_i = x_{i:1} \cup x_{i:2} \cup x_{i:3}$ on the sub-formulas of G_i. As the elements x are designed so that $x_{i:k} \upharpoonright \alpha_{i,j}$ is a unique element, we then get that $x_{i:1} \upharpoonright \alpha_{i,j} = \{a_{i,j}\}$, $x_{i:2} \upharpoonright \alpha_{i,j} = \{b_{i,j}\}$ and $x_{i:3} \upharpoonright \alpha_{i,j} = \{c_{i,j}\}$. Therefore $S_i \upharpoonright \alpha_{i,k} \in \Gamma^{\bot*}(\alpha_{i,j})$ by design, and consequently $S_i \upharpoonright \alpha_{i,j}^{\bot} \in \Gamma^*(\alpha_{i,j}^{\bot})$. On the first formula of G_i we have $S \upharpoonright (B_{i,1} \& B_{i,2}) \in \Gamma^*(B_{i,1} \& B_{i,2})$ since the $S \upharpoonright B_{i,1} \neq \emptyset$ and $S \upharpoonright B_{i,2} \neq \emptyset$. At last we deal with the M_i part. Here, the reader should remember that $x_{i:k} \upharpoonright M_i$ is, in the general case, not a singleton. Indeed, nothing prevents taking two elements $x_{...,i:k,..}$ and $x'_{...,i:k,...}$ from choosing different additive resolutions in M_i, except for the distinguished $\oplus$, in which they have to agree by choosing either $C_{i,1}$ or $C_{i,2}$ by definition. However, the key point is that M is a context made exclusively of $⅋$ and $\oplus$. Therefore, two different $\bot$ elements in it are automatically hypercoherent. Furthermore $S_i \upharpoonright C_{i,1}[\alpha_{i,1}^{\bot}] \oplus C_{i,2}[\alpha_{i,1}^{\bot}]] \in \Gamma^{\bot,*}(C_{i,1}[\alpha_{i,1}^{\bot}] \oplus C_{i,2}[\alpha_{i,1}^{\bot}])$ since $S_i \upharpoonright C_{i,1}[\alpha_{i,1}^{\bot}] \neq \emptyset$ and $S_2 \upharpoonright C_{i,2}[\alpha_{i,1}^{\bot}] \neq \emptyset$. Finally, we get to $S_i \upharpoonright M_i[S_i \upharpoonright C_{i,1}[\alpha_{i,1}^{\bot}] \oplus C_{i,2}[\alpha_{i,1}^{\bot}]] \in \Gamma^{\bot,*}(M_[[C_{i,1}[\alpha_{i,1}^{\bot}] \oplus C_{i,2}[\alpha_{i,1}^{\bot}]])$. A similar reasoning leads us to conclude that $S_i \upharpoonright \Xi_i \in \Gamma^{\bot}(\Xi_i)$, although it might be that S_i is not strictly incoherent on Ξ_i.

So, we get the following figure 7.3, displaying local coherence (or incoherence) of S_i, where we write P for Γ^*, O for $\Gamma^{\bot*}$, and N for $\Gamma \cap \Gamma^{\bot}$,

$(B_{i1} \& B_{i2})$	$M_i[C_1[\alpha_{i,1}^{\bot}] \oplus C_2[\alpha_{i,1}^{\bot}]]$	$\alpha_{i,2}$	$\otimes$	$\alpha_{i,2}^{\bot}$	···	$\alpha_{i,m-1}$	$\otimes$	$\alpha_{i,m}^{\bot}$	$\alpha_{i,m}$	Ξ_i
P	O	O		P		O		P	O	N/O
			O				O			

Figure 7.3: Local coherence for S_i on G_i

So now let us look at the global picture by studying the coherence of the family $\{x_l\}$ on each sub-formula of F_i. Let us remind that $(x_l) \upharpoonright A_i = S_i \upharpoonright A_i$ for every sub-formula A_i of G_i, by definition of S_i. This leads us to the following figure 7.4.

$\alpha_{i-1,m} \otimes (B_{i-1,1} \& B_{i-1,2})$ $\quad M[C_1[\alpha_{i,1}^{\perp}] \oplus C_2[\alpha_{i,1}^{\perp}]]$ $\quad \alpha_{i,2} \otimes \alpha_{i,2}^{\perp}$ $\quad \cdots$ $\quad \alpha_{i,m-1} \otimes \alpha_{i,m}^{\perp}$ $\quad \Xi_i$ $\quad \alpha_{i,m} \otimes (B_{i,1} \& B_{i,2})$

$O \quad P \quad O \quad O \quad P \quad O \quad P \quad N/O \quad O \quad P$

$O \quad O \quad O \quad O$

Figure 7.4: Local coherences of S in F_i

That is, for all formulas A of the sequent Γ, we got $S \upharpoonright A \in \Gamma^{\perp}(A)$, and, for some formulas A, $S \upharpoonright A \in \Gamma^{\perp,*}(F)$. And therefore, as each $x_l \in \mathcal{R}$, there is a set S, such that $S \subseteq \mathcal{R}$ and $S \in \Gamma^{\perp,*}(\Gamma)$. Consequently, the relation $\mathcal{R}$ is not a clique, and such a proof structure is rejected. It entails that to all our morphisms can be associated a valid proof net of MALL^{-}, as summed up in the following proposition.

Proposition 7.58. ChuHypGraph *is a fully complete model of* MALL^{-}.

Let us note that for the proof we make full use of the structure of $[\![X]\!]$, that is, our proof would not work if we limited ourselves to the Sierpinski domain. The simplest domain we could have work with would be a three-elements domain (with one $\perp$ and two non-compatible primes, as the one presented in [14]), however, we find it more natural to work with names, as it gives some intentional meaning to the elements of the domain.

We are furthermore in position to make a stronger claim.

Theorem 7.59. *There is a one-to-one correspondence between the morphisms of* ChuHypGraph *on* MALL^{-} *and the equivalence classes of* MALL^{-} *proofs of linear logic.*

As we do not deal with the units, it is a bit dubious to speak about a free-category at this stage, and that is the reason why we refrain from doing so. To do it we would need to define what is a star-autonomous category without units and with binary products but no final elements. The theorem is based on the following propositions, all proven by Dominic Hughes and Rob Van Glabbeek in their paper on proof-nets for MALL^{-} [46, 40], and transposed verbatim here.

Proposition 7.60. *Each linking is a proof-net if it is the denotation of a proof.*

Proposition 7.61. *Two* MALL^{-} *proofs translate to the same proof net if and only if they can be converted into each other by a series of rule commutations.*

Proof of theorem 7.59. Let us consider a morphism of ChuHypGraph(A, B). It denotes a unique linking on $A^{\perp} \parr B$. This linking is the denotation a proof, by full-completeness. Hence this linking leads to a proof net by 7.60. Furthermore, the morphisms of ChuHypGraph are precisely defined by their linkings. That is, two morphisms are equal if and only if they have same linking, and consequently, same proof net. Therefore, there is a bijection between the set of morphisms of $\mathsf{ChuHypGraph}(A, B)$, where $A, B \in \mathsf{MALL}^-$ and the proofs nets of $\vdash A^\perp, B$ and, by the proposition 7.61 the equivalence classes of proofs of MALL^- of $\vdash A^\perp, B$. □

This concludes our thesis.

Chapter 8

Conclusion

8.1 Summary

The objective of this essay was to develop a suitable model of linear logic, that would enjoy being fully-complete whilst dealing with atomic variables without relying on 2-categorical tools. Inspired by the work on nominal games semantics, we developed categorical models where denotations of objects and morphisms relied on nominal sets, that provide an elegant formalism to carry properly the vision of linear logic as logic of resources. Using names provides an intensional account to the elements of the category, whilst enabling us to define the nature of the atomic elements of our models, such as the "particles" of the geometry of interaction or the elements of the webs.

The vision of type variables as typed resources allows for a natural nominal semantics, relying on sorted nominal sets. This framework enabled the construction, in chapter 3, of simple instances of categories suitable to model linear logic. The formalisation of the notion of resources permitted a neat tracking of them. Notably, the linear use of resources could be enforced in an effective manner thanks to minimal nominal techniques.

On the second part, we embarked on the adventure of establishing full completeness. To that purpose, we introduced nominal asynchronous games semantics. This brought forth an unifying framework for semantics and syntax of tensorial logic 4. Exposing the handling of atomic resources within the lambda-terms, we defined the strategies accordingly. The resulting model yielded a sound interpretation of the proofs of tensorial logic with atomic variables 5, and was fully-complete. Projecting them on the category of nominal relations, we characterised precisely those that arised as denotations of proofs of MALL 6. However, the resulting model was not free, nominal relations being too "flat" to capture all the subtleties of the proofs.

Finally, we considered a more complex relational model in the third part 7, that took inspiration from concurrent games. We established a criterion characterising precisely those that denote MLL proofs. We then showed how the extensional content of concurrent strategies can be encoded into hypercoherence, and concluded with a full-completeness result for MALL without units. This model achieved the intended purpose of the thesis, though failing short of handling the units.

8.2 Further directions

This work is the first step towards the best possible result, that would consists in having a perfect syntax-free, quotient-free, 1-categorical model of linear logic. However, the final model of chapter 7 brings a solid working base for smaller intermediate results:

- **A single condition**: A relevant small step would be to unveil the relation between the handling the additives via hypercoherence, and the one that the Chu-conditions provide natively. If both agree, then it would allow us to present a simpler model, only relying on the Chu-conditions, and that would be fully-complete.
- **Graph game**: Another direction could be to further refine the graph games explored in the last chapter, and find the appropriate conditions that would make strategies on graph games fully complete. Furthermore, extensions to additives and units could be considered.
- **The units**: An important obstacle to overcome is the units. For both the additives or multiplicatives the challenge is strongly related, as it consists in incorporating positions that would correspond to untyped cells within the model. For the multiplicative units, a second possible way would be to generalise the notion of polarity, to account for the position corresponding to $1 \parr 1$ for instance. I have began some research in this direction.
- **The exponentials**: The second challenge are the exponentials. All the constructions that underline the model (hypercoherence [27], Chu-spaces, double-glueing [53]) have well-built exponentials, except for the nominal relations, that we need to further develop. One needs to make sure all these function well alongside another. An other possible approach would be to define them through pure categorical means [72].
- **Session types**: A deep correspondence between linear logic and session types was exhibited [92, 19], relating cut elimination steps and process reductions. Therefore, a model of linear logic should carry the necessary properties to form a model of session types. As the model we have found is fully-complete, it is hoped it can be adapted to form a fully-abstract model of the session types. Of course, the first step consists in restricting the study to the core of linear logic, the multiplicatives.
- **A new model of PCF**: Most of the current denotational fully-complete models of PCF rely on games [49, 7, 80]. As tensorial logic appears as the logic of games, it seems that they arise as a byproduct of the decomposition of the intuitonistic arrow into tensorial logic, and not the classical decomposition $A \Rightarrow B = !A \multimap B$ of linear logic. As a first step, it would be interesting to make that intuition clear. Secondly, the discovery of a static fully-complete model of linear logic might lead to a static fully-abstract model of PCF.

Of course, this relies on achieving step 2: a successful incorporation of the exponentials.

- **Relating static and dynamic model**: Finally, inline with the research relating games (or sequential algorithms) and hypercoherences [18, 29, 28, 68], it would be relevant to examine how our model of chapter 7 interconnects with the model of part 2. Establishing a formal equivalence should be particularly revealing, and a significant step towards a comprehensive model with units.

Chapter 9

Appendix

9.1 Appendix 1: Girard proof structures and hypercoherent relations

9.1.1 Hypercoherent nominal partial relations and proof structures

The whole point of this section is to repeat the arguments of [16] and check that they adapt smoothly within the context of hypercoherent partial nominal relations (that is, the category HypGraph), instead of di-natural transformations. We start by recalling the definitions of Girard proof structures. We then proceed to show that to each hypercoherent partial nominal relation a set of proof structures can be associated, and that this set can be narrowed down to a subset of canonical proof structures that satisfy additional desirable properties. We use these to reduce the cases that are relevant to prove a full completeness property.

As noted above, this section is a clear plagiarism from [16], simply replacing di-natural transformations with morphisms of HypGraph when needed. We present it for sake of completeness. Some proofs will be omitted.

9.1.1.1 Girard proof structures

The original definition of Girard proof structure can be found in [36]. The goal of the original paper was to find a suitable characterisation of proof structures that arise as denotations of MALL^- proofs.

Definition 9.1. *A proof structure Θ consists of the following :*

- *Occurrences of formulas and links. Each occurrence of link takes its premise(s) and conclusion(s) among the set of occurrences of formulas.*
- *A set of eigenweights $\{\mathsf{p}_1, ..., \mathsf{p}_n\}$ associated to $L_1, ..., L_n$ where $L_1, ..., L_n$ is the list of occurrences of &-links appearing in Θ. Each p_i is a boolean variable, that is, it can take*

value in the set $\{0, 1\}$.

- *For each occurrence of formula A or occurrence of link L a weight* $w(A)$ *(or* $w(L)$*), where the weight is a non-zero element of the boolean algebra generated by the* p_i.

Furthermore, the proof structure has to satisfy the following conditions.

- *Conditions on Links:*

Link L	$\frac{premise(s)}{conclusion(s)}\ L$	*weights of L and its premises*
Axiom Link	$\frac{(.)}{A\ A^{\perp}}\ \mathsf{ax}$	
⅋*-link*	$\frac{A\ \ B}{A ⅋ B}$ ⅋	$w(L) = w(A) = w(B)$
$\otimes$*-link*	$\frac{A\ \ B}{A \otimes B}\ \otimes$	$w(L) = w(A) = w(B)$.
&*-link*	$\frac{A\ \ B}{A \,\&\, B}\ \&_i$	$w(A) = \mathsf{p}_i.w(L)\ \ w(B) = (\neg \mathsf{p}_i).w(L)$
$\oplus_1$*-link*	$\frac{A}{A \oplus B}\ \oplus_1$	$w(A) = w(L)$
$\oplus_2$*-link*	$\frac{B}{A \oplus B}\ \oplus_2$	$w(B) = w(L)$

- *We require that for each occurrence of a formula A,* $w(A) = \sum_k L_k$ *where* (L_k) *is the family of links with conclusion A.*
- *Moreover, if* L_1, L_2 *are two distinct links sharing the same conclusion A, then* $w(L_1).w(L_2) = 0$.
- *Given an occurrence A of a formula that is not the premise of any link (that is, a conclusion of the proof structure), then* $w(A) = 1$.

Furthermore, a proof structure satisfies the two following conditions :

- ***dependency****: Every weight appearing in the proof structure is monomial, that is, is a product of eigenweights and negations of eigenweights.*
- ***technical*** *: For every weight v appearing in* Θ*, if v depends on* p_i *then* $v \subseteq w(L_i)$ *(and the weight of* L_i *does not depend on* p_i*).*

A formula occurrence is a **conclusion** of Θ if it is not a premise of any link. A link L is **terminal** if it has one conclusion, this conclusion is a conclusion of Θ, and if furthermore $w(L) = 1$. If a link is terminal, it might be removed, leading to one or two new proof structures. The removal of terminal links for proof structures is defined below. Let Θ be a proof structure. We denote $C\mathcal{L}(\Theta)$ the set of conclusions of Θ.

- If L is a terminal $\otimes$-link with conclusion $A \otimes B$, and premises A, B. Then the removal of $\otimes$ (when possible) consists in partitioning the formula occurrences of $\Theta \setminus \{A \otimes B\}$ into two sets X, Y, such that $A \in X$ and $B \in Y$. The partitioning must be done in such a way that whenever a link L' has a conclusion in X (respectively Y), then all the premises and conclusions of L' belong to X (respectively Y). It therefore leads to two new proof structures X and Y, that have A for X, B for Y among their conclusion(s).

- If L is a $\parr$-link with premises A, B and conclusion $A \parr B$, then the removal of L consists in removing $A \parr B$ and the link L from θ. It leads to a new proof structure with conclusions $C\mathcal{L}(\theta) \setminus \{A \parr B\} \uplus \{A, B\}$.
- If L is a $\oplus_1$-link with premise A and conclusion $A \oplus B$, then the removal of L consists in removing L together with its conclusion $A \oplus B$, leading to a proof structure with conclusion(s) $C\mathcal{L}(\Theta) \setminus \{A \oplus B\} \uplus \{A\}$.
- If L is a $\oplus_2$-link with premise B and conclusion $A \oplus B$, then the removal of L consists in removing L together with its conclusion $A \oplus B$, leading to a proof structure with conclusion(s) $C\mathcal{L}(\Theta) \setminus \{A \oplus B\} \uplus \{B\}$.
- If L is a &-link (call it L_i) with premises A, B and conclusion $A \,\&\, B$, then the removal of L_i consists in removing the link L_i together with its conclusion $A \,\&\, B$, and then forming two proof structures Θ_A and Θ_B. The proof structure Θ_A is formed by making the replacement $\mathsf{p}_i = 0$ and keeping only those formulas and links whose weights are non 0. Respectively, Θ_B is formed by taking $\mathsf{p}_i = 1$ and keeping only those formulas and weights that are non 0.
- In the case where we consider the MIX rule, then there is a 0-removal, that removes no link, but splits the proof structure into two new ones. That is, let us suppose that Θ can be partitioned into two proof structures Θ_1, Θ_2 such that whenever a link L has a conclusion in Θ_1 (resp Θ_2), then all its premises and conclusions belong in Θ_1 (resp Θ_2), then the MIX splitting consists in splitting the proof structure Θ into Θ_1, Θ_2.

Definition 9.2. *A proof structure is* MALL$^-$ ***sequentialisable*** *if it can be reduced to a set of axiom links by an iteration of link-removals. Furthermore, it is* MALL$^-$ + MIX *sequentialisable if it can be reduced to a set of axiom links by an iteration of link removals and* MIX*-splittings.*

By definition, if a proof structure is sequentialisable, then there is a proof π that can be canonically associated to it, as each link-removal basically corresponds to a rule of MALL$^-$, and the MIX-splitting corresponds to the use of the MIX rule. On the other hand, to each proof π of a MALL$^-$ sequent, one can associate a proof structure. However, this proof structure is not unique.

9.1.1.2 The criterion

Just as when dealing with MLL$^-$ proof structures, we look at switchings in order to establish if a MALL$^-$ proof structure is sequentialisable, though the notions of switchings differ between the two cases. We define switchings for MALL$^-$ proof structures.

Definition 9.3.
- *A **switching** for a* MALL$^-$ *proof structure consists in:*
  1. *A choice of a valuation ϕ_S which is a function $\{\mathsf{p}_1, ..., \mathsf{p}_n\} \to \{0, 1\}$. Therefore ϕ_S defines a function from weights to the boolean, that we also note ϕ_S by abuse of notation. The **slice** of the proof structure $sl(\phi_S(\Theta))$ corresponds to the restriction of Θ to those formulas and links such that $\phi_S(w) = 1$.*
  2. *For each $\parr$-link L of $sl(\phi_S(\Theta))$, the choice $S(L) \in \{L, R\}$.*

3. *For each &-link L_i of $sl(\phi_S(\Theta))$, the choice of a formula $S(L_i)$ in $sl(\phi_S(\Theta))$, called jump of L. This jump must depend on p_i in the sense that it must be the conclusion of a link L such that $\mathsf{p}_i \subseteq w(L)$. A jump is* ***normal*** *if $S(L)$ is a premise of L. Otherwise, it is* ***proper****.*

- *A* ***normal switching*** *is a switching without proper jumps.*
- *For a proof structure Θ together with a switching S, we define the undirected graph Θ_S as follows:*
  1. *The vertices of Θ_S are the formulas of $sl(\phi_S(\Theta))$.*
  2. *For each axiom link in $sl(\phi_S(\Theta))$, one draws an edge between the corresponding literals conclusions of the link.*
  3. *For each $\otimes$-link, one draws two edges from the conclusion $A \otimes B$ of the link onto its premises.*
  4. *For a $⅋$-link, one draws an edge from its conclusion to its occurrence chosen by the switching. That is, if the conclusion of the $⅋$-link L is $A ⅋ B$, and $S(L) = l$ (respectively $S(L) = r$), then one draws an edge from $A ⅋ B$ to A (respectively B).*
  5. *For a $\oplus_1$-link, one draws an edge between its premise occurrence A and its conclusion $A \oplus B$.*
  6. *For a $\oplus_2$-link, one draws an edge between its premise occurrence B and its conclusion $A \oplus B$.*
  7. *For a &-link L_i, one draws an edge between its conclusion $A \& B$ and its jump $S(L_i)$.*

With this definition, we are in position of establishing the correctness criterion. We define a **proof net** as a proof structure that is sequentialisable.

Theorem 9.4. *[36]*

- *A proof structure Θ is MALL^- sequentialisable if and only if for all switchings S the graph Θ_S is connected and acyclic.*
- *A proof structure Θ is $\mathsf{MALL}^- + \mathsf{MIX}$ sequentialisable if and only if for all switchings S the graph Θ_S is acyclic.*

In our case, we would like to use a slightly different characterisation, that will turn out to be more adapted for us.

Proposition 9.5. *[16] A proof structure Θ is MALL^- sequentialisable if for all switchings S the graph Θ_S is acyclic, and for all normal switchings S the graph Θ_S is connected.*

Indeed, the graph of a normal switching is approximatively the same graph as the one of a multiplicative switching (that is, the graph coming from a switching of a MLL^- proof structure). Hence, while working with a relation $\mathcal{R} \in \mathsf{ChuHypGraph}$, the fact that for all &-resolutions Ψ we get that $\mathcal{R} \upharpoonright \Psi \in \mathsf{ChuLinNom}$, and the Chu nominal partial relations are MLL complete, will entail that for each normal switching S the graph Θ_S (where Θ is a proof structure associated with $\mathcal{R}$) is acyclic and connected. Therefore, what will remain to be proven is the acyclicity of the proof-structure for all proper switchings.

9.1.1.3 HypGraph and proof structures

The gal of this section is to attribute to each morphism of HypGraph a non-empty set of proof structures.

Lemma 9.6. *In a Girard proof structure:*

- *if we replace an occurrence of a $\otimes$-link with a $⅋$-link, and we replace every occurrence of $\otimes$ in vertices (formulas) hereditarily below it in the proof structure with the corresponding $⅋$, then it remains a valid proof structure.*
- *if we replace an occurrence of a $⅋$-link with a $\otimes$-link, and we replace every occurrence of $\otimes$ in vertices hereditarily below it in the proof structure with the corresponding $\otimes$, then it remains a valid proof structure.*

We write MIX the operation on proof structure that replaces an occurrence of $\otimes$ with a $⅋$ as defined in the above lemma.

We recall that PALL^- is the fragment of MALL^- without $\otimes$. Each proof structure of PALL^-, that is, without $\otimes$-link, is PALL^- + MIX sequentialisable. That is, to each proof structure of PALL^- can be associated a proof of PALL^- + MIX. This follows from the following lemma, that was first proven in [42].

Lemma 9.7. *[42]* ***Softness of MALL-proof structure:*** *Let Θ be a MALL proof structure, such that Θ has at least, one occurrence of a &-link. There there is, at least, one &-link that has weight 1.*

Corollary 9.8. *Each proof structure of* PALL^- *is sequentialisable.*

The proof of the corollary follows approximatively the same lines as the proof of PALL^- full completeness for morphisms of HypGraph 7.54. The full proof can be found in [42].

Therefore, one can devise a way to associate some Girard proof structures to each morphism of HypGraph. First, we compose it with all the necessary mix rules to remove all the $\otimes$, and obtain a morphism of type PALL^-. This is the denotation of a morphism π of PALL^- + MIX (since HypGraph is fully complete) and we associate to it a set of proof structures. In these proof structures, we replace all the necessary $⅋$ with $\otimes$ in order to fit the type of the original relation. This remains a proof structure, this time of MALL^-. We define this procedure properly in the next paragraphs.

To do that formally, we first have to define the other direction, namely that to each MALL^- + MIX sequentialisable proof structure Θ, then Θ gives rise to a unique morphism of HypGraph. That follows also from the fact that HypGraph is a star-autonomous category with products (that accepts the mix-rule), and hence soundly model MALL + MIX. For recreational purposes, we re-prove it here in the context of proof-structure.

Proposition 9.9. *Let π be a* MALL$^-$+MIX *sequentialisable proof structure* Θ. *Then* Θ *determines a unique morphism of* HypGraph, $[\Theta]$, *such that* $[\Theta]$ *is the denotation of a* MALL + MIX *proof. Therefore we have a mapping:*

$$[\ .\ ] : \mathsf{MALL}^- + \mathsf{MIX}\ \textit{sequentialisable proof Structures of type}\ \Gamma \ \Rightarrow\ \mathsf{HypGraph}(I, \Gamma),$$

where the type of a proof structure is the $⅋$*-tensored type of its conclusion(s), seen as a list of formulas. That is, the type of* Γ *is* $⅋\, C\mathcal{L}(\Gamma)$*, where we assume there is a canonical ordering on the formulas of* $C\mathcal{L}(\Gamma)$.

Proof. The proof is done by induction on the number of &-links. If there is none, then Θ can be identified with a proof of M⊕LL+MIX, and then therefore it is simply a set of axiom links. These can be faithfully interpreted within the category of hypercoherent partial nominal relations. In case there is a &-link, then in particular, by softness of proof structures, there is one with weight 1. Name it L_i. Therefore, it corresponds to its namesake in the sequent. Consider the two proof structures obtained by taking $w_i = 0, 1$ respectively. Then to each of them can be assigned a unique hypercoherent partial nominal relation by induction hypothesis. Then, by combining them following the sequentialisation of the proof structure, one obtains a hypercoherent partial nominal relation associated with π. □

We extend this mapping to a wider class of proof structures. We call **extension** of a partial function, another function that has greater or equal domain of definition, and such that the two functions agree on their common domain.

Given $[\ .\ ]^k$ a function from proof structures to partial nominal relations, we define $[\ .\ ]^{k+1}$ as follows:

- If $\Theta \notin \mathsf{Dom}([\ .\ ]^k)$ but there is a $\otimes$-link in Θ such that $\mathsf{MIX} \circ \Theta \in \mathsf{Dom}([\ .\ ]^k)$, where MIX is applied to that occurrence of an $\otimes$-link .
- If, furthermore, given a nominal relation $\mathcal{R}$ such that $\mathcal{R}$ is of type Θ, and such that $\mathsf{mix} \circ \mathcal{R} = [\mathsf{MIX} \circ \Theta]^k$, (where, again, mix is applied to the nominal relation $\mathcal{R}$ in accordance with the MIX from the first point), then $\Theta \in \mathsf{Dom}([\ .\ ]^{k+1})$.
- Since mix is monic (in our case, it is actually the identity), this $\mathcal{R}$ is uniquely defined and we set $[\ \Theta\ ]^{k+1} = \mathcal{R}$.
- Since the applications of mix are commutative, the mapping $[\ .\ ]^k$ is well-defined.

Note that in the case of hypercoherent partial nominal relations, mix is the identity. In the following, we take $[\ .\ ]^1 = [\ .\ ]$, the function from proof structures to nominal relations previously defined. Note that $[\ .\ ]^{k+1}$ is indeed an extension of $[\ .\ ]^k$. We denote $[\ .\ ]^* = \cup_k [\ .\ ]^k$.

Lemma 9.10. ***Lifting property***. *Let* $\mathcal{R}, \mathcal{R}'$ *two relations such that* $\mathcal{R}' = \mathsf{mix} \circ \mathcal{R}$. *Let* Θ, Θ' *be proof structures such that* $\Theta' = \mathsf{MIX} \circ \Theta$. *Then if* $[\Theta']^* = \mathcal{R}'$ *and* Θ, $\mathcal{R}$ *are of the same type (that is, the* mix, *MIX are applied to the same part of the formula), then* $[\Theta]^* = \mathcal{R}$.

Given a morphism $\mathcal{R}$ of HypGraph, we can speak of its weakly assigned proof structures to be :

$$WPS(\mathcal{R}) = \{\Theta \mid [\Theta]^* = \mathcal{R}\}.$$

Our goal is to show that this function $[\ .\ ]^*$ is surjective, that is, that to each morphism $\mathcal{R}$ of HypGraph its set $WPS(\mathcal{R})$ is non-empty. Consider a nominal hypercoherent relation $\mathcal{R}$, of type Δ. Then replace all the $\otimes$ in Δ by $⅋$, leading to a new type $\Delta_{⅋}$. Then $\mathcal{R}$ can also be seen as a morphism of type $\Delta_{⅋}$. As the nominal hypercoherent relations are fully complete for $\mathsf{PALL}^- + \mathsf{MIX}$, there exists a proof π such that $[\![\pi]\!] = \mathcal{R}$, and therefore a proof structure $\Theta_{⅋}$ such that $[\Theta_{⅋}] = \mathcal{R}$. Now, we apply back the MIX-rules and we obtain a proof structure Θ of type Δ such that $[\Theta]^* = \mathcal{R}$, by the lifting property.

Corollary 9.11. *Given $\mathcal{R}$ a morphism of* HypGraph*, $WPS(\mathcal{R}) \neq \emptyset$.*

Lemma 9.12. *A nominal relation $\mathcal{R}$ denotes a MALL^--proof net, and hence, a proof, if and only if the set of weakly associated proof structures $WPS(\mathcal{R})$ contains a proof net Θ.*

Proof. The only if part is direct, as if $\mathcal{R}$ is a denotation of a proof, then there exists a proof net Θ such that $[\Theta] = \mathcal{R}$. So let us tackle the other direction, and assume there is a $\Theta \in WPS(R)$ such that Θ is a proof net. Then $\mathcal{R} = [\Theta]^*$ by definition. In that case, Θ is in the domain of $[\ .\ ]$ since Θ is a proof net. Therefore $\mathcal{R}$ is the denotation of a MALL^- proof by soundness. □

9.1.1.4 Strongly canonical proof structures

We add certain requirements to the domain of $[.]^*$, in order to obtain a subset $PS(\mathcal{R}) \subseteq WPS(\mathcal{R})$ that enjoys some desirable properties.

Definition 9.13. ***Semantical splitting of a hypercoherent partial nominal relation***. *Given a hypercoherent partial nominal relation $\mathcal{R}$ of MALL^--type we define a $\{\otimes, ⅋, \&, \oplus, \mathsf{mix}\}$-splitting as follows:*

- *A $\otimes$ (respectively $\&, \mathsf{mix}$) splitting of $\mathcal{R}$ is written $\mathcal{R}_1 \otimes \mathcal{R}_2$ (respectively $\mathcal{R}_1 \,\&\, \mathcal{R}_2$, $\mathcal{R}_1 \mathsf{mix} \mathcal{R}_2$), when $\mathcal{R}$ of type $\Delta_1, \Delta_2, A_1 \otimes A_2$ (respectively $\Delta, A \,\&\, B$, and Δ_1, Δ_2) is split into $\mathcal{R}_1 \otimes \mathcal{R}_2$ (respectively $\mathcal{R}_1 \,\&\, \mathcal{R}_2$, and $\mathcal{R}_1 \mathsf{mix} \mathcal{R}_2$) with $\mathcal{R}_i$ of type Δ_i, A_I (respectively Δ, A_i and Δ_i).*
- *A $⅋$-splitting (respectively $\oplus_1$, $\oplus_2$) of $\mathcal{R}$ of type $\Gamma, A ⅋ B$ (respectively $\Gamma, A \oplus B$, and $\Gamma, A \oplus B$) is a relation $\mathcal{R}_1$ of type Γ, A, B (respectively Γ, A and Γ, B) such that $\mathcal{R} = \mathcal{R}_1$ (respectively $\mathcal{R} = \mathsf{inl}(\mathcal{R}_1)$ and $\mathcal{R} = \mathsf{inr}(\mathcal{R}_1)$).*

Each splitting corresponds to a MALL + MIX rule.

A total splitting is a sequence of splittings such that no further splitting can be done. A total splitting terminates if the final relations obtained after the splittings are identities on literals. A total splitting is represented as a tree, where the root is the original relation, the binary nodes are the binary splittings and the unary nodes the unary ones.

Definition 9.14. *A total splitting α is **legal** if for every &-splitting appearing in α, the &-splitting happens provisio that it was impossible to execute any $\{\otimes, \mathsf{mix}, ⅋, \oplus_1, \oplus_2\}$-splitting to the relation at this stage.*

That is, a legal splitting happens to split the transformation with a &-splitting only when no other splitting is possible. Then of course, if a hypercoherent partial nominal relation has a total splitting that terminates, then it has a total legal splitting that terminates. Notably, each denotation of a proof has a total legal splitting.

Definition 9.15. *A proof structure is said to satisfy the :*

- ***Unique Link Property (UL)**: If L in θ is either a $\otimes$-link, ⅋-link or a &-link with conclusion an occurrence D, then it is the only link whose conclusion is that occurrence of D.*
- ***No duplicate axiom-link property (NDAL)**: There occurs in Θ no distinct axiom links $\mathsf{ax}_1, ..., \mathsf{ax}_n$ whose two conclusions coincide and whose sum of weights is 1.*

We try to rely on legal splittings to relate morphisms of HypGraph and proof-structures that satisfy UL and NDAL. For that, we rely on the following lemmas, that have been first proven in [42].

Lemma 9.16. *Suppose that a $\mathsf{MALL}^- + \mathsf{MIX}$ proof π is obtained from proof(s) π_i by means of a $\mathsf{MALL}^- + \mathsf{MIX}$-rule α. Then for any UL proof structure Θ_i whose sequentialisations are π_i, one can construct a canonical proof structure Θ such that its sequentialisation is π, its splitting corresponding to α (that is, removing of the terminal link corresponding to α), yields the proof structure(s) Θ_i, and furthermore Θ satisfies the UL.*

Lemma 9.17. *Given a proof structure, one can replace sets of axioms links sharing the same conclusions and such that the sums of their weights is 1 with unique axiom links. This process is confluent, terminates, and gives rise to a new proof satisfying the NDAL. Furthermore, calling Θ the original proof, and $\bar{\Theta}$ the newly obtained one, if $[\Theta] = \mathcal{R}$ then $[\bar{\Theta}] = \mathcal{R}$.*

Proposition 9.18. *Let $\mathcal{R}$ be a morphism of HypGraph. Then every terminating total legal splitting α of $\mathcal{R}$ can be canonically interpreted by a unique $\mathsf{MALL} + \mathsf{MIX}$ proof net Θ_α that satisfies UL and NDAL and such that $[\Theta_\alpha] = \mathcal{R}$.*

The proof relies heavily on propositions 9.16 and 9.17.

Proof. The proof is done by induction on the size of α. We simply do it when the last rule of α is a &. Then there is two total splitings α_i for two nominal relations $\mathcal{R}_i$ such that $\mathcal{R} = \mathcal{R}_1 \& \mathcal{R}_2$. By induction hypothesis, this leads to two proof structures Θ_i that are proof nets satisfying the UL and NDAL. So we can get a proof structure $\Theta_1 \& \Theta_2$, such that it satisfies UL following 9.16. Furthermore, following the lemma 9.17, as Θ_i is sequentialisable, so is $\bar{\Theta} = \Theta_i \& \Theta_2$. Finally, we replace possible sets of axiom links with same conclusions as in lemma 9.17 until we get a proof structure Θ_α that satisfies NDAL. Following lemma, $[\Theta_\alpha] = \mathcal{R}$. □

Given a total legal splitting α of a morphism $\mathcal{R}$ of HypGraph we write Θ_α for its associated proof structure.

Let denote the function $[\ .\ ]_-$ defined by:

$$[\ .\ ]_- : \text{Proof structures} \rightarrow \text{Nominal relations}$$

such that $[\ .\ ]$ is an extension of $[\ .\ ]_-$.

$$\Theta \in \mathsf{Dom}([\ .\ ]_-) \text{ if } \Theta = \Theta_\alpha \text{ for some legal splittings of } \alpha \text{ of } [\Theta].$$

Let $\mathcal{R}$ be in the image of $[\ .\ ]$. Then $\mathcal{R}$ is a denotation of a proof. Hence there exists a terminating legal splitting α. Therefore, one can associate to $\mathcal{R}$ a proof structure Θ_α as in the proposition above. Finally, $\mathcal{R} = [\Theta_\alpha]$. What it tells us is that the image of $[.]$ and $[.]_-$ is the same. We define $[\ .\ ]^*_-$ from $[\ .\ ]_-$ the exact same way we defined $[\ .\ ]^*$ from $[\ .\ ]$. Then once again $[\ .\ ]^*_-$ has the same image as $[\ .\ ]^*$, but every proof structure in its domain satisfies UL and NDAL.

As a result, we define:

$$PS(\mathcal{R}) = \{\Theta \mid [\Theta]^*_- = \mathcal{R}\}$$

Then $PS(\mathcal{R}) \subseteq WPS(\mathcal{R})$ and $WPS(\mathcal{R}) \neq \emptyset \Rightarrow PS(\mathcal{R}) \neq \emptyset$. In particular, to each hypercoherent partial nominal relation, one can assign to it a non-empty set $PS(\Theta)$ of proof structures satisfying UL and NDAL.

These proof structures can be built almost the exact same way as described in the paragraph above 9.1.1.3. We start with an hypercoherent nominal relation $\mathcal{R}$. We replace all the $\otimes$ by $⅋$ in the type of $\mathcal{R}$. By full completeness, it is the denotation of a proof π of $\mathsf{PALL}^- + \mathsf{MIX}$. This proof π leads to a sequentialisation α. This sequentialisation can be chosen to be legal. Hence we can assign a proof structure Θ_α as in proposition 9.18. This proof structure satisfies the UL and the NDAL. Now, we replace the appropriate $⅋$-links with $\otimes$-links, in order to get back a proof structure of the appropriate type.

Proposition 9.19. *A nominal relation $\mathcal{R}$ denotes a MALL^- proof if and only if there is a $\Theta \in PS(\mathcal{R})$ such that Θ is a proof net.*

The proof is the exact same as the one of proposition 9.12.

9.1.1.5 Canonical splitting

The goal of this section is to strengthen the above proposition. We would like to prove that a relation $\mathcal{R}$ denotes a proof if and only if all the proof structures in $PS(\mathcal{R})$ are proof nets. That way, one can consider any of them.

Proposition 9.20. *Suppose that a hypercoherent partial nominal relation $\mathcal{R}$ is such that $PS(\mathcal{R})$ is not empty, and $\mathcal{R}$ can be split via a @-splitting,* $@\in \{\otimes, \mathsf{mix}, \parr, \oplus_1, \oplus_2, \&\}$. *Then every $\Theta \in PS(\mathcal{R})$ has the corresponding @-splitting.*

The proof is exactly the same as in [16], simply replacing the word dinatural transformation by partial nominal relation.

Definition 9.21. *We say that a proof structure Θ **has a cycle** C if there is a switching S such that C appears in Θ_S. We say that a partial nominal relation $\mathcal{R}$ **yields a cycle** if $PS(\mathcal{R}) \neq \emptyset$ and $\exists\Theta \in PS(\mathcal{R})$ such that Θ has a cycle.*

The next corollary is fundamental.

Corollary 9.22. *Suppose that a* HypGraph *morphism $\mathcal{R}$ can be split into hypercoherent partial nominal relation(s) $\mathcal{R}_i$. Then if $\mathcal{R}$ yields a cycle, then so does $\mathcal{R}_i$, for at least one of the i.*

Proof. Suppose that a relation $\mathcal{R}$ can be split by @ into R_i. Suppose moreover that $\mathcal{R}$ yields a cycle, that is, there is a $\Theta \in PS(\mathcal{R})$, such that Θ has a cycle. Then Θ can also be split via @ into Θ_i. Hence if Θ has a cycle, then one of Θ_i must have it as well. Since $\Theta_i \in PS(\mathcal{R}_i)$, we conclude. □

Theorem 9.23. *Let $\mathcal{R}$ be a Chu partial nominal hypercoherent relation, that is $\mathcal{R} \in$* ChuHypGraph. *Then $\mathcal{R}$ denotes a proof if and only if $\forall\Theta \in PS(\mathcal{R})$, Θ is a proof net.*

Proof. We only need to prove the only if part. Let $\mathcal{R}$ be a nominal relation that is the denotation of a proof, and suppose that there exists $\Theta \in PS(\mathcal{R})$ such that Θ is not a proof net. As $\mathcal{R}$ is the denotation of proof, every additive resolution of Θ (that is, every normal switching of Θ), is connected. Hence, Θ is not a proof net only if Θ yields a cycle. Let @ be a splitting of $\mathcal{R}$, then it is also a splitting of Θ and this splitting yields proof structure(s) with cycle. So let α be a total splitting of $\mathcal{R}$ such that α terminates (it exists since $\mathcal{R}$ is the denotation of a MALL proof). We can apply α to Θ, and we obtain a set of proof structure(s), and at least one of them has a cycle. But each of this proof structure is an axiom on literals, and hence is acyclic. This is a contradiction. □

Corollary 9.24. *Let $\mathcal{R}$ be a morphism of* ChuHypGraph. *Then $\mathcal{R}$ is the denotation of a proof if and only if $\forall\Theta \in PS(\mathcal{R})$, Θ yields no cycle.*

The proof follows from the theorem above.

9.1.2 Narrowing down the cycles

This section aims at summing up the results obtained in [16] on cycles in proof structures. Unless deemed relevant for the understanding of the chapter, no proofs will be given. We start by laying some definitions. The canonical **orientation** of jump edges is from their conclusion $A \&_i B$ to their jump $S(L_i)$. A cycle in a graph Θ_S is **oriented** if there is an orientation of the whole cycle that respects the orientation of the jumps.

Lemma 9.25. *Let Θ be a proof structure that is connected for every normal switching S_0. Then given a switching S, every cycle of Θ_S can be transformed into an oriented cycle of $\Theta_{\bar{S}}$, where $\bar{S}$ is another switching that has same valuation as S.*

A cycle is furthermore **canonical** if:

- Every proper jump on the cycle is the conclusion of an axiom-link
- If A, B formulas on the cycle are nested in the syntactic formula tree, then the orientation of C induces a directed path from A to B or vice-versa. Then the unique path from A to B (or from B to A) in the cycle is the one corresponding to their nesting in the syntactic formula tree.

Lemma 9.26. *For an arbitrary proof structure Θ and a switching S, every cycle of Θ_S can be transformed into a canonical circle of $\Theta_{S'}$, for some switching S'. If the cycle of Θ_S was oriented, then so is the new canonical cycle.*

Therefore, it is enough to consider canonical oriented cycles. Given a canonical cycle, we denote $\&_i$, $1 \leq i \leq n$ the occurrences of &-links on which the cycle properly jump, and ax_{i+1} the jump of $\&_i$.

We go on narrowing down the class of cycles one should consider. A cycle C in a graph Θ_S is called **simple** if, for every link K whose conclusion is a proper jump $S(L_i)$ lying on C (so K is an axiom in the case of canonical cycles), then $w(K) = \epsilon.\mathsf{p}_i.v$, where $\epsilon \in \{1, \neg\}$, p_i is the weight associated with L_i, and v does not depends on any eigenweight associated with a &-link whose conclusion lies on C. In particular, a canonical cycle C is simple if for all i such that ax_{i+1} is the axiom whose conclusion is a jump from L_i lying in C, then $w(\mathsf{ax}_{i+1}) = \epsilon.\mathsf{p}_i.v$ where v does not depend on p_i ($1 \leq i \leq n$).

Lemma 9.27.

- *For a simple cycle $C \in \Theta_S$, let $\&_k$ ($1 \leq k \leq m$) denotes the list of all &-links whose conclusions lie on C. Then $w(\&_k)$ does not depend on p_i, $1 \leq i \leq n$.*
- *Every oriented cycle C of Θ_S can be transformed into a simple oriented cycle C' of Θ_S. Furthermore, if C is canonical, then so is C'.*

9.1.2.1 Global cycles

We add a final characterisation to the cycles. We want them to be "global". Unfortunately, a proof-structure might have a cycle without having a global one. However we will prove that if the category has relations that yield cyclic oriented proof-structures, then it has some that yield global oriented cycles. First, we fix some terminology. A cycle C passes trough L, (where L is a link), means that the conclusion(s) of L lie(s) in C. A cycle is **global** if it passes through all the &-links whose weights are 1.

Lemma 9.28. *Given a simple canonical oriented cycle C, C global entails:*

- *For all $\&_i$ occurrences of &-links that cause proper jumps on C, $w(\&_i) = 1$.*
- *For the weights $w(\mathsf{ax}_{i+1}) = \epsilon\, \mathsf{p}_i.v_i$, if the v_i depends on a eigenweight p_r, then $w(\&_r)$ depends on p_i.*
- *Given $1 \leq i \neq j \leq n$, if $w(\mathsf{ax}_{i+1}) = \epsilon\, \mathsf{p}_i.v_i$ and $w(\mathsf{ax}_{j+1}) = \epsilon\, \mathsf{p}_j.v_j$ then the sets of eigenweights on which v_i and v_j depends are disjoint.*

We are interested in proving some properties about the way axiom-links behave in global cycles.

Definition 9.29. *Let Θ be a NDAL proof structure and α a literal in Θ. We say that a valuation ϕ* ***yields two distinct axiom-links*** *with relation to an eigenweight p and a literal α if the following property holds. Let ϕ' be the same as ϕ for all eigenweights except for p where $\phi'(\mathsf{p}) = \neg\phi(\mathsf{p})$. Then $\alpha \in sl(\phi(\Theta))$, $\alpha \in sl(\phi'(\Theta))$, and the axiom links $L \in sl(\phi(\Theta))$ and $L' \in sl(\phi'(\Theta))$ that have conclusion α are different (that is, they have different conclusions).*

That entails that the two axiom links L, L' depends on p.

One can prove this fundamental property for global cycle.

Proposition 9.30. *Suppose a proof structure Θ has a global simple oriented cycle C living in a valuation w such that $\forall i.1 \leq i \leq n.w(\alpha_{i+1}) = 1$, where α_{i+1} is the conclusion of ax_{i+1} that lies on the cycle. Then there exists a switching S such that $C \in \Theta_S$ and ϕ_S yields two distinct axiom-links with respect to p_i and α_{i+1} for all $i \in \{1, ..., n\}$.*

9.1.3 Reduction to &-semi-simple sequents

Definition 9.31. *A context is a sequent generated from distinguished holes, noted $\star_i$, literals and MALL connectives, such that each hole appears at most once within the sequent. It is denoted $\Gamma[\star_1, ..., \star_n]$. A hole has a multiplicative occurence in the context if all the connectives in the parse tree from a root to this hole are multiplicative.*

We may substitute any hole with a formula in the context. This substitution is written $\Gamma[A_1, .., A_n]$. A $\mathsf{M}\oplus\mathsf{LL}$ context is a context where all the connectives are restricted to the $\mathsf{M}\oplus\mathsf{LL}$ fragment of MALL, that is, without &.

Definition 9.32. *A $\mathsf{M}\oplus\mathsf{LL}$ sequent Δ is* ***semi-simple*** *if it can be written $\Delta = \Gamma[l_{11}\otimes...\otimes l_{1k}, ..., l_{n1}\otimes ... \otimes l_{nm}]$ and Γ is a $\mathsf{M}\oplus\mathsf{LL}$ context without $\otimes$ (that is, a $⅋\oplus\mathsf{LL}$-context).*
A MALL sequent Δ is &− ***semi-simple*** *if it can be written $\Delta = \Gamma[A_{11}\&....\&A_{1k}, ..., A_{n1}\&...\&A_{nm}]$ and Γ is a $\mathsf{M}\oplus\mathsf{LL}$ semi-simple context (that is, if we replace holes by literals, it is a semi-simple sequent).*

Proposition 9.33. *Suppose Γ is a $\mathsf{M}\oplus\mathsf{LL}$ sequent. Then there exists a list of $\mathsf{M}\oplus\mathsf{LL}$ semi-simple sequents $\vdash \Gamma_1, ..., \vdash \Gamma_n$ such that $\vdash \Gamma$ is provable (within $\mathsf{M}\oplus\mathsf{LL} + \mathsf{MIX}$) if and only if for all i then Γ_i is provable (within $\mathsf{M}\oplus\mathsf{LL} + \mathsf{MIX}$).*

This follows from two observations .

Lemma 9.34. *Let* $\Gamma := \Gamma[A \otimes (B \parr C)]$ *a* MALL *sequent. Let* $\Gamma_1 = \Gamma[(A \otimes B) \parr C]$ *and* $\Gamma_2 = \Gamma[(A \otimes C) \parr B]$. *Then* Γ *is provable if and only if* Γ_1 *and* Γ_2 *are provable. Furthermore,* $\Gamma[(A \oplus B) \otimes (C \oplus D)]$ *is provable if and only if* $\Gamma[(A \otimes C) \oplus (A \otimes D) \oplus (B \otimes C) \oplus (B \otimes D)]$ *are provable.*

Proposition 9.35. *Suppose* Γ *is a* MALL *sequent. Then there exists a list of* MALL *semi-simple sequents* $\vdash \Gamma_1, ..., \vdash \Gamma_n$ *such that* $\vdash \Gamma$ *is provable within* MALL *if and only if for all* i Γ_i *is provable.*

The proof is the exact same as the one of the proposition before, but by replacing literals with & formulas.

9.1.3.1 &-semi-simple sequents and global cycles

The goal is to prove that one can consider only oriented canonical global cycles.

Proposition 9.36. *Consider the set of hypercoherent partial nominal relations* $\mathcal{R}$ *of &-semi-simple types such that there is a* $\Theta \in PS(\mathcal{R})$ *such that* Θ *has a oriented cycle. If this set is non-empty, then there is one hypercoherent partial nominal relation* $\mathcal{R}$ *such that there exists* $\Theta \in PS(\mathcal{R})$ *such that every oriented cycle in* Θ *is global.*

Proof. Let take $\mathcal{R}$ of T type minimal such that $\mathcal{R}$ has a oriented cycle, where the order on types is defined as follows:

$$(\text{number of } \otimes \text{ , number of } \{\parr, \&, \oplus\})$$

Then $\mathcal{R}$ cannot be split using any rule of MALL (otherwise, its proof structure would split as well, and one of them would have a oriented cycle). The resulting $\mathcal{R}'$ would have a lesser type in the above hierarchy, contradicting the minimality of $\mathcal{R}$. Therefore, $\forall \Theta \in PS(\mathcal{R})$, Θ has no terminal $\otimes$-splitting link, no terminal $\{\&, \parr, \oplus_1, \oplus_2\}$-link and is not the union of two proof structures.

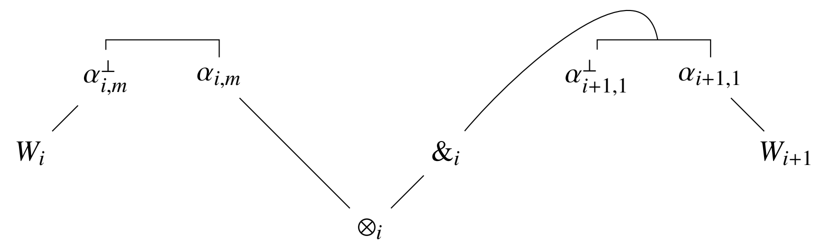
Let us consider a $\Theta \in PS(\mathcal{R})$, and a &-link L within it such that $w(L) = 1$. Then there is no &-link below it (otherwise its weight would not be 1). Now, if all links below it would be $\oplus$, $\parr$, then the proof structure would split. Hence there exists a $\otimes$-link below it. Now suppose that $\otimes$ is not directly below the &-link. If a $\oplus$ is below the &-link, then it has weight 1, and therefore the $\mathcal{R}$ comes from an injection, which is excluded. On the other hand, if a $\parr$ is just below the &, it means that there is a $\parr$ above the $\otimes$ in the parse tree, which contradicts the &-semi-simplicity of $\mathcal{R}$ (since the &-link L is minimal).

We know that Θ has a oriented cycle. Let us assume for contradiction that it has a non-global cycle. So there is a oriented cycle $C \in \Theta$, and a &-link L of weight 1 such that C does not pass through L. By the discussion above, this link has a $\otimes$ right below it. Since the cycle does not pass through L, this $\otimes$ can be replaced by a $\parr$ in Θ (and all occurrences below it), and the proof structure Θ' hence obtained still have the oriented cycle. Now note that the corresponding $\mathcal{R}'$ has a type lesser than $\mathcal{R}$. Finally, we note that pushing the newly created connective $\parr$ above using lemma 9.34 in order to obtain a semi-simple sequent does not change its number of connectives, hence we obtain a semi-simple sequent of lesser type, contradicting the minimality of the type of $\mathcal{R}$. □

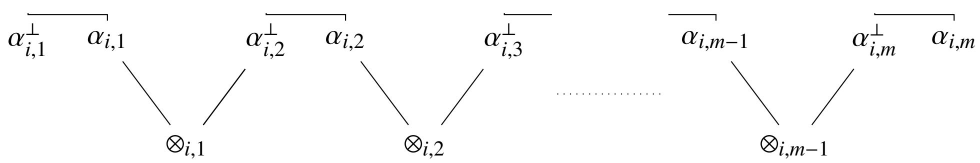
9.1.4 Final form for the decisive argument

We start by assuming that the category of Chu hypercoherent nominal partial relations is not fully complete. Therefore, this entails the existence of a morphism $\mathcal{R}_1$ of ChuHypGraph that has a canonical proof structure with a cycle. As explained above, we can reduce it to another morphism $\mathcal{R}_2$ of ChuHypGraph of &-semi-simple type. Now, as every normal switching of $\Theta_{\mathcal{R}_2}$ is connected, it can be transformed into an oriented cycle. Therefore, seeing it as a proof-structure coming from a morphism of HypGraph, proposition 9.36 entails that there is a morphism $\mathcal{R}_3$ of HypGraph, whose canonical associated proof structure $\Theta_{\mathcal{R}_3}$ has a oriented cycle, and such that every oriented cycle of it is global. In particular, it has a simple, oriented canonical global cycle.

So, as explained in [16], the shape of the cycle around the $\&_i$ is as follows:

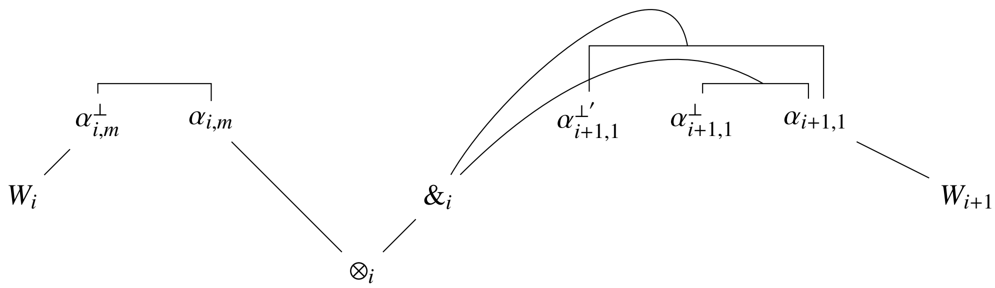


We each W_i is shaped as below, without any &-jump.

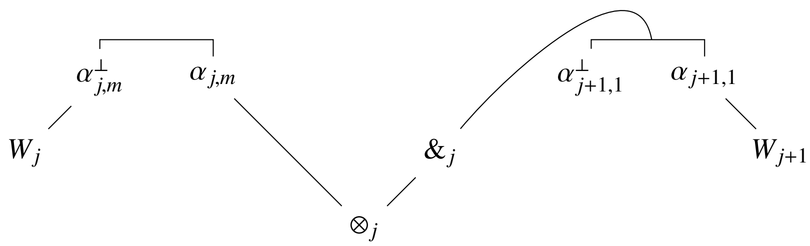


Furthermore, the W_i path does not contain any proper jump, and bounces on $\otimes$-links occurrences. Furthermore, by semi-simplicity, all the links between $\alpha^{i,k}$ and $\otimes_{i,k}$ are $\otimes$-links, and therefore $w(\alpha_{i,k}) = w(\otimes_{i,k})$. Finally, note that there are also only $\otimes$-links between α^i_1 and $\otimes_{i+1}$. Also there are only $\otimes$-links between $\otimes_{i+1}$ and $\&_{i+1}$. As the cycle is global, $w(\&_{i+1}) = 1$. This entails that $w(\otimes_{i+1}) = 1$, and finally $w(\alpha^i_1) = 1$. Finally, we obtain that $w(\alpha^i_k) = 1$ for all k and i.

However, if we change the valuation of ϕ we can make the axiom ax_i, between $(\alpha^i_m)^\perp$ and α^i_m disappears. On the other hand, as $w(\alpha^i_m) = 1$, it stays within the proof structure. So in the proof structure there must exists two axioms whose conclusions are α^i_m. One is ax_i, of weight p_{i-1} (or $\neg\mathsf{p}_{i-1}$ but we pick p_{i-1} without loss of generality), and the other one ax'_i has weight $\neg\mathsf{p}_{i-1}.v_i$. Let us show that ax'_i has weight exactly $\neg\mathsf{p}_{i-1}$. Let us note that we can draw a jump from $\&_{i-1}$ to ax'_i. Now, with a new valuation only changing p_{i-1}, all the $\&_{j-1}$ and ax_j remain present (for $j \neq i$), hence the simple oriented cycle remains almost unchanged. Furthermore, this cycle is global and $w(\mathsf{ax}'_i) = \neg\mathsf{p}_i$. Therefore, it looks like as displayed in the following figure.



Furthermore, picking a different $\&_j$, the cycle remains unchanged between the two valuations. That is, for the two different valuations switching p_i, the cycle around $\&_j$ remains as follows:



Now, since the hypercoherent partial nominal relations satisfy the mix rule, we apply mix to all the $\otimes$ occurrences in the type of $\mathcal{R}$ that do not appear on the cycle. Furthermore, by associativity and commutativity, we may assume that $\otimes$ appears just below the two literals. Hence the cycle remains. We then obtain a sequent of the following form :

$$\vdash F_1, ..., F_n$$
$$\text{where } F_i = \alpha_{i,m} \otimes (B^i_1 \,\&\, B^i_2), N_i[\alpha^\perp_{i+1,1}, \alpha^\perp_{i+1,1}], \alpha_{i+1,1} \otimes \alpha^\perp_{i+1,2}, ..., \alpha_{i+1,m-1} \otimes \alpha^\perp_{i+1,m}, \Xi_i$$

where m depends on i, where $N_i[\star_1, \star_2]$ is either $N_{i,1}[\star_1] \oplus N_{i,2}[\star_2]$ or $(N_{i,1}[\star_1] \oplus N'_{i,1}) \parr (N_{i,2}[\star_2] \oplus N'_{i,2})$, with all the connectives in N_i being $\parr$. Ξ_i represents the remaining formulas after having apply the mix transformations to every $\otimes$ that were not part of the cycle. That is $\Xi_i = E_{11} \oplus E_{12}, ..., E_{m1} \oplus E_{m2}, l_1, .., l_k$ where l_i are literals, and all connectives in E_{ij} being $\oplus$ and $⅋$.

So we have 2^n global circles whose structures are displayed in figure 7.2 of page 272, where the cycles are displayed in red, and the axiom links not part of the cycle in black.

So to conclude about full-completeness, one only needs to prove that there is no morphism of HypGraph of this form.

9.2 Appendix 2: Composition of frugal strategies

The goal of this section is to prove directly that the innocent, frugal, semi-linear strategies frugal form a category. The main property to target is the associativity of composition.

$$\mathsf{frugal}(\widehat{\mathsf{frugal}(\hat{X};_{Rel}\hat{Y})} ;_{\mathsf{Rel}} \hat{Z}) = \mathsf{frugal}(\hat{X} ;_{\mathsf{Rel}} \widehat{\mathsf{frugal}(\hat{Y} ;_{\mathsf{Rel}} \hat{Z})})$$

This is tackled by projecting arenas, and strategies, into a category analogous to the nominal separated polarised relations. In this category, only names of $\mathbb{A}_T$ are looked at as names. That is, we work within the nominal universe whose set of names is simply $\mathbb{A}_T$. In it, the names of $\mathbb{A}_{\mathsf{cells}}$ are simply elements of empty support. Then, we restrict to a certain class of strategies, called pre-linear. They are those such that a name cannot be played by proponent before opponent introduces it. Furthermore, proponent cannot play a name introduced by opponent more than once. These strategies project via a projection function that we will later define onto nominal relations $\mathcal{R}$ such that $\forall x \in \mathcal{R}.\nu(\mathsf{Pos}(x)) \subseteq \nu(\mathsf{Neg}(x))$. Then the construction done with nominal polarised separated relations, traces, and substitutions lifts straightforwardly to this case. This allows us to ensure that they form a category. Finally, we make use that the projection function from arenas and pre-linear strategies to set of separated lists and nominal pre-linear relations commutes with every construction defined, and therefore we conclude that pre-linear strategies forms a category.

We give the details below.

9.2.0.1 Technical details

Each arena projects into a set of annotated, polarised and separated lists. To make it clear, we introduce a linearisation of the partial order of vertices coming from the tree. More precisely, we introduce a function $f : V_A \to \mathbb{N}$, such that f is equivariant ($v \simeq v' \Rightarrow f(v) = f(v')$), f is almost injective ($v \neq v' \Rightarrow f(v) \neq f(v')$), and $v \leq v' \Rightarrow f(v) \leq f(v')$. Such a function is defined for instance below:

- Given a tree A with root v, $f_A(v) = 1$.
- If the root of A is justifying n equivalence classes of vertices $v_1, ..., v_n$ that are roots of subtrees $B_1, ..., B_n$, then writing $|B_i|$ for the number of equivalence classes of vertices in B_i, and taking a $w \in B_j$, $f_B(w) = f_{B_j}(w) + \sum_{i<j} |B_i| + 1$.

Given a vertex v, we define $\llcorner v \lrcorner$ as the triple $(\ulcorner\ulcorner v \urcorner\urcorner, \mathsf{pol}(v), f(v))$. We define the function $\mathsf{projlist}$ from positions to lists as the function that, given a position, returns its unique list of $\llcorner v \lrcorner$, for all vertices v present in the position, ordered increasingly. For a position p, we write $V(p)$ for the set of all vertices that appear in p. Then, for a position p, we have $\mathsf{projlist}(p) = l_1 ... l_k . l_{k+1} .. l_n$ such that:

- $\forall v \in V(p), \exists i \leq n. l_i = \llcorner v \lrcorner$.
- $\forall i \leq n. \exists v \in V(p). l_i = \llcorner v \lrcorner$.
- $\forall i \leq n-1$, given v, v' such that $\llcorner v \lrcorner = l_1$ and $\llcorner v' \lrcorner = l_{i+1}$ then $f(v) < f(v')$.

And finally, given an arena A, we define $\mathsf{projlist}(A) = \{\mathsf{projlist}(p) \mid p \in \mathsf{Frugal}(A)\}$. We want to show that $\mathsf{projlist}$ is a faithful functor from the category of semi-linear frugal innocent strategies to the category of semi-linear polarised annotated relations. We name this category $\mathsf{SemNomLinPol}$. It has objects orbit-finite nominal sets of polarised, annotated, separated lists:

$$L \ := \ (a, p, n) \ \mid \ \mathsf{inl}(L) \ \mid \ \mathsf{inr}(L) \ \mid \ L_1 . L_2$$

where $a \in \mathbb{A} \cup \{\bullet\}$, $p = \{-1, 1\}$, $n \in \mathbb{N}$, and $L_1 \#_{\mathsf{pol},T} L_2$. We write $L_1 \#_{\mathsf{pol},T} L_2$ to indicate that our lists are polarised separated only with regard to names in $\mathbb{A}_T$. In this category, only the names of $\mathbb{A}_T$ are dealt with as names, those are $\mathbb{A}_{\mathsf{cells}}$ are simply considered as elements of empty support. That is, each element of type (a, p, n), $a \in \mathbb{A}_{\mathsf{cells}}$ is dealt with as a $(\bullet, p)$ in the category $\mathsf{NomLinPol}$. The morphisms of $\mathsf{SemNomLinPol}$ are nominal (with regards to $\mathbb{A}_T$) relations $\mathcal{R}$ such that, for each element x of $\mathcal{R}$, $\nu_T(\mathsf{pos}(x)) \subseteq \nu_T(\mathsf{neg}(x))$. That is, they are not linear anymore, but only **semi-linear**.

Just as in section 3, where it was made clear that the composition of nominal polarised relations was ultimately relying on tracing of permutations, the composition of semi-linear polarised relations ultimately relies on tracing of injective partial functions. Injective partial functions form a category, whose objects are finite sets. Its trace is the one of partial functions, defined in 5.2.4, which is itself a simple adaption of the one of permutations to partial functions.

The category $\mathsf{SemNomLinPol}$ organises itself as a category whose composition is defined similarly than the one of $\mathsf{NomLinPol}$. That is, given two semi-linear nominal polarised relations $\mathcal{R} : A \to B$ and $\mathcal{Q} : B \to C$ we get :

$$\mathcal{R} ; \mathcal{Q} = \{r \in A^{\perp} \star_{pol} C \mid \exists r_1 \in \widehat{\mathcal{R}}, r_2 \in \widehat{\mathcal{Q}}, r \upharpoonright A = r_1 \upharpoonright \widehat{A}, r_1 \upharpoonright \widehat{B} = (r_2 \upharpoonright \widehat{B})^{\perp}, r_2 \upharpoonright \widehat{C} = r \upharpoonright C\}.$$

Equivalently, in analogy with what was defined above, writing Frugal for the partial function that discriminates the elements of $\widehat{D}$ that can be seen as separated polarised lists of D , we get:

$$\mathcal{R}; Q = \mathsf{Frugal}(\widehat{R};_{\mathsf{Rel}} \widehat{Q}).$$

Given a morphism $\sigma^\bullet : A \triangleright B$, then the positions of $\sigma^\bullet$ projects into lists $\mathsf{projlist}(A) \times \mathsf{projlist}(B)$, and therefore $\mathsf{projlist}(\sigma^\bullet) : \mathsf{projlist}(A) \to \mathsf{projlist}(B)$. Furthermore, one can see straightforwardly that $\mathsf{projlist}(\widehat{\sigma}) = \widehat{\mathsf{projlist}}(\sigma)$. Finally, as composition is defined similarly for innocent and frugal strategies seen as relations and their lists counterparts, one can devise that :

$$\mathsf{projlist}(\mathsf{frugal}(\hat{X};_{\mathsf{Rel}} \hat{Y})) = \mathsf{frugal}(\mathsf{projlist}(\widehat{X}); \mathsf{projlist}(\widehat{Y})).$$

This amounts to:

$$\mathsf{projlist}(\sigma; \tau) = \mathsf{projlist}(\sigma); \mathsf{projlist}(\tau)$$

Now, using the fact that $\mathsf{SemNomLinPol}$ is a category, we get:

$$\begin{aligned} \mathsf{projlist}((\sigma; \tau); \varsigma) &= \mathsf{projlist}(\sigma; \tau); \mathsf{projlist}(\tau) \\ &= (\mathsf{projlist}(\sigma); \mathsf{projlist}(\tau)); \mathsf{projlist}(\varsigma) \\ &= \mathsf{projlist}(\sigma); (\mathsf{projlist}(\tau); \mathsf{projlist}(\varsigma)) \\ &= \mathsf{projlist}(\sigma; (\tau; \varsigma)) \end{aligned}$$

And, using the fact that $\mathsf{projlist}$ is faithful, we obtain $(\sigma; \tau); \varsigma = \sigma; (\tau; \varsigma)$.

Moreover, in the case of identity:

$$\sigma; \mathsf{id} = \mathsf{frugal}(\widehat{\sigma};_{\mathsf{Rel}} \widehat{\mathsf{id}}) = \mathsf{frugal}(\widehat{\sigma}) = \sigma.$$

And similarly for identity acting on the left.

Therefore, the frugal-innocent, pre-linear frugal strategies compose as in the category of semi-linear nominal relations, and therefore form a category.

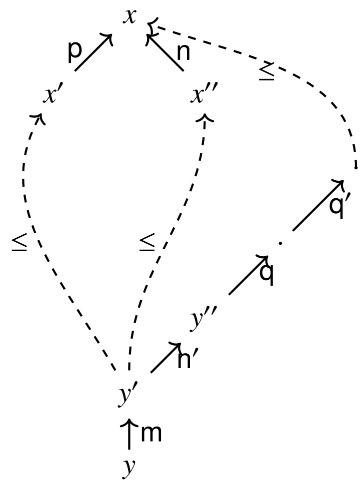
9.3 Appendix 3: Backward confluence

Proposition 9.37. *The nine properties of definable sets entail backward confluence.*

Proof. Let $x \in X$, we consider y maximal in X strictly below x. Then by mutual attraction, there is an O-move $y \xrightarrow{\mathsf{m}} y'$ and a player move $x' \xrightarrow{\mathsf{n}} x$ such that $y' \leq x'$. If, supposedly, $y' \neq x'$, then, as $y' \leq x$, there is a player move $y' \xrightarrow{\mathsf{n}''} x''$ such that $x'' \leq x$. If $x'' = x$, then there would be two paths from y' to x, one by is n'', and the second is a path to x' followed by a move n. As the graph is acyclic, we conclude that $x'' \neq x$ and therefore $x'' < x$. Thus, we get a chain $y < x'' < x$

whose three elements belong in X. Or, y was supposed to be maximal under y. Therefore, there is no such x'' and y is only two moves away from x.

Let $x \in X$, $\mathsf{p} : x' \to x$ a player move, and we consider y maximal below x' in X. Then $y \leq x' < x$ and by mutual attraction there is an O-move $y \xrightarrow{\mathsf{m}} y'$ and a player move $\mathsf{n} : x'' \to x$ such that $y' < x$. Let us suppose that $y' \neq x'$. Hence $y' < x'$. We seek a contradiction. By forward confluence, there is a second player move $\mathsf{n}' : y' \to y''$ such that $y'' \leq x$. We draw about this case below :



Then, by assumption, $\neg(y'' \leq x')$, as y is supposed to be maximal in X under x'. However, $y'' \leq x$. Therefore, $\mathsf{n}' = \mathsf{p}$. Furthermore, assume that $y'' < x''$, and therefore $x' \neq y'$. Then let us consider an O-move above y'' such that $\mathsf{q} : y'' \to z$ and z below x. From the assumption that every position of X is reached by an alternative play, such a move exists. Now, as x' is a legal position, this move q is not justified by $\mathsf{p} = \mathsf{n}'$, and neither by m since they are both O-moves. Therefore, the position $y \uplus \mathsf{q}$ is legal, and below x, x'. By forward confluence there is a player move q' above $y \uplus \mathsf{q}$ such that $y \uplus \mathsf{q} \uplus \mathsf{q}' \in X$ and lies below x. By maximality of y under x', $\mathsf{q}' = \mathsf{p}$. Furthermore, as they are both under x, $(y \uplus \mathsf{q} \uplus \mathsf{p}) \uparrow y''$. Therefore, their intersection $y \uplus \mathsf{p}$ belongs in X. However, this is a unbalanced position, and therefore contradicts the definition of X. Finally, we conclude that $y'' = x$, that is, the maximal position of X below x' is only one O-move below x'.

Finally, let us assume that there are two O-moves below x' that reach two different positions of X. That is, two moves $\mathsf{m} : y \to x'$ and $\mathsf{m} : y' \to x'$, such that $y, y' \in X$, and $y \neq y'$. Then, as they are both under x, so is their intersection. Or this one is 3-moves away from x, so either x, or the intersection cannot be reached by an alternating play, which is, a contradiction.

So finally, given $x \in X$, and a player move $\mathsf{p} : x' \to x$, there is a unique O-move $\mathsf{m} : y \to x'$, such that $y \in X$.

Remaining is to prove that given $X \ni z \leq x'$, then $z \leq y$. If it were not the case, it would mean $\mathsf{m} \in z$. Then by compatibility under union, $X \ni z \sqcup y$ is below x', and above $y \uplus \mathsf{m}$. Hence it is x', which is, once again, a contradiction. So finally $z \leq y$, which concludes the proof.

□

9.4 Appendix 4: Substitutions

We recall that for an element x, given $A \subseteq_{\mathsf{fin}} \mathbb{A}$, we write $[x]_A$ for its A-orbit, that is, its set of renamed variants under permutations fixing atoms in A.

$$[x]_A = \{\pi \cdot x \mid \pi \# A\}$$

If A is empty, $[x]_\emptyset$ is simply the orbit of x. Given an element x and a name a, we define the **atom-abstraction** $\langle a \rangle x$ by:

$$\langle a \rangle x = [(a, x)]_{\nu(x) \setminus \{a\}}$$

In particular, note that $\nu(\langle a \rangle x) = \nu(x) \setminus \{a\}$. The following defininition of substitutions originates from [32]. We recall that $\mathcal{V}_{FM}$ is the Fraenkel-Mostowski model of set theory, whose objects are recursively finitely supported sets and elements. This is presented with more details in Section 4.1 of the thesis.

Definition 9.38. *[32] A* ***nominal substitution*** *is a set-theoretic nominal function on $\mathcal{V}_{FM}$ written $[a \to x] \cdot z$, expressed in the language of nominal set theory, such that for all z, x (sets or elements) and atoms a:*

- $b \# z.[a/x] \cdot z = [b/x] \cdot ((b, a) \cdot z)$
- $a \# z.[a/x] \cdot z = z$
- $[a/x] \cdot a = x$
- $[a/a] \cdot z = z$
- $c \# x \Rightarrow [a/x] \cdot (\langle c \rangle z) = \langle c \rangle ([a/x] \cdot z)$
- $a \# y \Rightarrow [b/y] \cdot ([a/x] \cdot z) = [a/([b/y] \cdot x] \cdot ([b/y]/z)$

In our case, we will restrict all along to strict substitutions.

Definition 9.39. *A substitution is* ***strict*** *if it only substitutes sorted atoms $a \in \mathbb{A}_X$ with other atoms $b \in \mathbb{A}_X$ of the same sort. More generally, we call strict substitution the composition of a sequence of strict subsitutions.*

In the following, we will write $[a_n/b_n][a_{n-1}/b_{n-1}]...[a_1/b_1]$ for the strict substitution $[a_n/b_n] \cdot ([a_{n-1}/b_{n-1}] \cdot (...([a_1/b_1] \cdot \_)...))$.

Lemma 9.40. *Let $[a/b]$ be a strict substitution such that $a \# b$. Then $\forall x. a \# [a/b] \cdot x$.*

Proof. Given a subset $A \subseteq \mathbb{A}$, and two atomic elements a, b, define $A(a \mapsto b)$ by $(A \setminus a) \cup b$ if $a \in A$, or just A otherwise. We prove that $\nu([a/b] \cdot x) \subseteq \nu(x)(a \mapsto b)$. As the substitution function is equivariant, we know that :

$$\nu([a/b] \cdot x) \subseteq \nu(x) \cup \nu(a) \cup \nu(b).$$

Now let us pick c fresh, $c\#x, a, b$. From the first axiom of substitutions, we get:

$$[a/b] \cdot x = [c/b] \cdot ((c, a) \cdot x)$$

and using again the fact that the substitution is an equivariant function:

$$\nu([c/b] \cdot ((c, a) \cdot x)) \subseteq \nu((c, a) \cdot x) \cup \nu(c) \cup \nu(b)$$

and as $c\#x$, it entails $\nu((c, a) \cdot x) = (c, a) \cdot \nu(x) = \nu(x)(a \mapsto c)$ since c is fresh. So we know that $\nu([a/b] \cdot x)$ lives at the intersection of both sets. By noticing that c is not in the first one, neither that is a in the second, we conclude that :

$$\nu([a/b] \cdot x) \subseteq (\nu(x) \setminus a) \cup \nu(b)$$

Furthermore, in the case where $a \notin \nu(x)$, by the axiom (2) of substitution, $\nu([a/b] \cdot x) = \nu(x)$. Putting together the case $a \in \nu(x)$ and the case $a \notin \nu(x)$ we obtain:

$$\nu([a/b] \cdot x) \subseteq (\nu(x))(a \mapsto b),$$

which is the required property. This entails $a\#[a/b] \cdot x$. □

Corollary 9.41.
- *If $b \neq a$ then $[a/c][a/b] \cdot x = [a/b] \cdot x$.*
- $[b/c][a/b] \cdot x = [b/c][a/c] \cdot x$
- *if $b\#z$ then $[a/b] \cdot z = (b, a) \cdot z$.*

Proof. The first bullet follows from the property (2) of substitutions, the second from property (6), and the last from the conjunction of the axioms (2) and (4) that entails the following sequence of equations $[a/b] \cdot z = [b/b] \cdot ((b\ a) \cdot z) = (b, a) \cdot z$. □

Lemma 9.42. *Let e be a strict substitution. Then there exists n a natural number, and $(a_i)_{i=1..n}, (b_i)_{i=1..n}$ two families of names such that $a_i \in \mathbb{A}_X \Rightarrow b_i \in \mathbb{A}_X$ and:*
- $e = [a_n/b_n]..[a_1/b_1]$
- $\forall i \geq j \in [1, n]. a_i \neq b_j$
- $\forall i \neq j \in [1, n]. a_i \neq a_j$

Proof. Let $[c_j/d_j]...[c_1/d_1]$ be any strict sequential substitution. We do the proof by induction on j, the length of the sequence. If $j = 0$ then every property stated above trivially hold. So let us consider the case $e_{j+1} = [c_{j_1}/d_{j+1}][c_j/d_j]...[c_1/d_1]$. By applying the induction hypothesis on $e_j = [c_j/d_j]...[c_1/d_1]$, we get a number n and two families $(a_i)_{i\leq n}, (b_i)_{i\leq n}$ such that $e_j = [a_n/b_n]...[a_1/b_1]$. From there, $e_{j+1} = [c_{j+1}/d_{j+1}][a_n/b_n]...[a_1/b_1]$. Either this is the required form and the result holds at this stage, or at least one of the two remaining properties fail. That is, either $\exists i.b_i = c_{j+1}$ or $\exists i, a_i = c_{j+1}$. We show that in both cases we can push $[c_{j+1}/d_{j+1}]$ along the sequence until it hits the $[a_i/b_i]$ with whom there is a conflict without creating additional conflict.

So let us assume we have successfully push $[c_{j_1}/d_{j+1}]$ down the sequence (maybe modifying some b_l in it in passing), so that now $e_{j+1} = [a_n, b'_n]...[a_{k+1}/b'_{k+1}][c_{j+1}/d_{j+1}][a_k/b_k]...[a_i/b_i]...[a_1/b_1]$ and the only conflict remains between $[c_{j+1}/d_{j+1}]$ and $[a_i/b_i]$. We deal with the different cases:

- if $\{a_k, b_k\} \cap \{c_{j+1}, d_{j+1}\} = \emptyset$ then $[a_k/b_k][c_{j+1}/d_{j+1}] = [c_{j+1}/d_{j+1}][a_k/b_k]$.
- If $\{a_k, b_k\} \cap \{c_{j+1}, d_{j+1}\} \neq \emptyset$ then the cases $c_{j+1} = a_k$ and $c_{j+1} = b_k$ are excluded since they correspond to the conflictual case. So remaining is either $b_k = d_{j+1}$, in which case they permute as above, or $a_k = d_{j+1}$ then $[c_{j+1}/d_{j+1}][a_k/b_k] = [c_{j+1}/d_{j+1}][a_k/d_{j+1}]$ (by the second point of 9.41)

By doing this procedure we have changed the occurrences $\{b_k\}$ hence possibly breaking the second condition. If $\exists l \geq k$, $a_l = d_{j+1}$, then this would have been in conflict with $[c_{j+1}/d_{j+1}]$ in the first place, which is a contradiction. Therefore we can continue this series of permutations until we hit $[a_i/b_i]$ which is conflicting. We deal with the three possible different conflicts.

- The first case is $c_{j+1} = b_i$, corresponding to the condition (1) being broken. In that case $[c_{j+1}/d_{j+1}] \cdot [a_i/b_i] = [b_i/d_{j+1}][a_i/b_i] = [b_i/d_{j+1}][a_i/d_{j+1}]$ (this is the point (2) of corollary 9.41). In that case the conflict is solved, and this transformation does not create additional conflicts.
- The second case is $c_{j+1} = a_i$. In that case we have $[c_{j+1}/d_{j+1}][a_i/b_i] = [a_i/d_{j+1}][a_i/b_i] = [a_i/b_i]$ (this is the point (1) of corollary 9.41). Once again, this transformation does not create additional conflicts.
- The third case would be a conjunction of the previous cases, that is: $c_{j+1} = b_i$ and $c_{j+1} = a_i$. However, in that case $a_i = b_i$, which is excluded.

Therefore, by doing this procedure we have resolved the left most conflict, without creating additional ones. Following the same procedure we can resolve additional conflicts in a finite number of steps, leading to a final form without conflicts. This one hence satisfies the required properties.

□

Lemma 9.43. *Let e a subsitutial strict substitution such that* $e = [a_1/b_1]...[a_n/b_n] = [a'_1/b'_1]...[a'_{n'}/b'_{n'}]$*, and the families* $\{(a_i, b_i)\}, \{(a'_i, b'_i)\}$ *satisfy the conditions of the lemma above. Then* $n' = n$ *and there is a permutation* $\sigma \in S_n$ *such that* $a'_{\sigma(i)} = a_i$*,* $b'_{\sigma(i)} = b_i$*.*

Proof. If we apply e to a_i then because of the conditions on the family $\{(a_i, b_i)\}$, $e \cdot a_i = b_i$. So as e does not let a_i invariant, there must be a k such that $a_i = a'_k$. Now, because the same condition holds for the family $\{(a'_i, b'_i)\}$, $e \cdot a_k = b'_k = b_i$. Therefore, there is a injection $\mathsf{inj}; [1, n]/[1, n']$ such that $a_i = a'_{\mathsf{inj}(i)}$ and $b_i = b'_{\mathsf{inj}(i)}$. By doing the reverse direction, we can conclude of a reverse injection, and therefore $n' = n$ and $\mathsf{inj} \in S_n$. □

Therefore, the families $(a_i), (b_i)$ are canonical, and one can speak of **canonical form** as well as **length** of a strict sequential substitution.

Lemma 9.44. *Let x an element of nominal set, and e a strict sequential substitution. Then $e \cdot x = e' \cdot (\pi \cdot x)$, where $e' = [a_n/b_n]...[a_1/b_1]$ a canonical form, and, writing e'_i for $[a_i/b_i]...[a_1/b_1]$, then $a_{i+1}, b_{i+1} \in \nu(e'_i \cdot (\pi \cdot x))$ and π is a nominal permutation.*

Proof. We write $e = [a_n/b_n]...[a_1/b_1]$ for a canonical form of e. We do the proof by induction on the length i of e. So let us consider $([a_{i+1}/b_{i+1}][a_i/b_i]...[a_1/b_1]) \cdot (\pi \cdot x)$. Then either $a_{i+1} \notin \nu(e'_i \cdot (\pi \cdot x))$ and $[a_{i+1}/b_{i+1}].e_i \cdot (\pi \cdot x) = e_i \cdot (\pi \cdot x)$. So suppose $a_{i+1} \in \nu((e_i \cdot (\pi \cdot x))$, then either $b_{i+1} \in \nu(e_i \cdot (\pi \cdot x))$ as in the lemma, or $[a_i/b_{i+1}] \cdot (e_i \cdot (\pi \cdot x)) = (a_{i+1}, b_{i+1}) \cdot (e_i \cdot (\pi \cdot x)) = ((a_{i+1}, b_{i+1}) \cdot e_i) \cdot ((a_{i+1}, b_{i+1}) \circ \pi) \cdot x$, allowing us to conclude. □

This lemma allows us to conclude that the action of a strict substitution on a element consists of a permutation followed by a name-merging. For instance, note that $\nu(e_i \cdot (\pi \cdot x)) \subset \nu(\pi \cdot x)$ for all i. This is established by simple induction.

Corollary 9.45. *Let e as above, such that $e \cdot x = e' \cdot (\pi \cdot x)$, and $e' = [a_n/b_n]...[a_1/b_1]$ as above Then the following properties hold:*

- $\forall i, j. \in [1, n].i \neq j \Rightarrow a_i \neq a_j$.
- $\forall i, j \in [1, n].a_i \neq b_j$
- $\forall \sigma \in S_n.e' = [a_{\sigma(1)}/b_{\sigma(1)}]...[a_{\sigma(n)}/b_{\sigma(n)}]$

Proof. The first property follows from the first condition on normal forms. The third is a direct consequence of the two first. So we need to focus on the second one. Suppose there exist i, j such that $a_i = b_j$. The case $i \geq j$ is forbidden by the canonical form conditions, so we restrict ourselves to the case $i < j$, and we consider for contradiction the smallest j greater than i such that $a_i = b_j$. Then, by lemma 9.40, we know that $a_i \notin \nu(e_i \cdot (\pi \cdot x))$. Furthermore, as j is the smallest number such that $b_j = a_i$, $e_{j-1} \cdot (\pi \cdot x)\#a_i$. As $b_j = a_i$, this contradicts the fact that $b_j \in \nu(e_{j-1} \cdot (\pi \cdot x))$. We conclude. □

To end-up this paragraph, we introduce this lemma that clarifies the relation between substitution and permutations.

Proposition 9.46. *Let x an element of a nominal set, and π a nominal permutation. Then $\exists e \in \Xi$ such that $\pi \cdot x = e \cdot x$*

Proof. Let $Z \subseteq \mathbb{A}$ be a finite subset of same cardinal of $\nu(\pi)$ and such that $Z\#\pi, x$. Let f be a bijection between Z and $\nu(\pi)$, and consider ρ the associated permutation, such that $\nu(\rho) = \nu(\pi) \sqcup Z$, defined by $\rho(a) = f(a)$ if $a \in \nu(x)$, $\rho(a) = f^{-1}(a)$ if $a \in Z$, $\rho = \mathsf{id}$ otherwise. Consequently, ρ is an involution : $\rho \circ \rho = \mathsf{id}$. We define $\varrho = \pi \circ \rho$. Hence $\varrho \circ \rho = \pi$. We define the associated substitutions. Let $\{a_i \mid i \in [1, n]\}$ be any numbering of $\nu(x)$. We first define $e = [a_i/f(a_i) = \rho(a_i)]_{i=1..n}$. Then as $f(a_i)$ is always fresh for a_i, x, then $[a_i/f(a_i)] \cdot x = (a_i, f(a_i)) \cdot x$ and therefore $e \cdot x = \rho(x)$. We define equally e' for ϱ. Then $e' \cdot e \cdot x = \varrho \circ \rho(x) = \pi(x)$. □

Index

Bibliography

[1] S. Abramsky. “Abstract Scalars, Loops, and Free Traced and Strongly Compact Closed Categories”. In: *Proceedings of the First International Conference on Algebra and Coalgebra in Computer Science*. CALCO’05. Springer-Verlag, 2005, pp. 1–29.

[2] S. Abramsky. “Event Domains, Stable Functions and Proof-Nets”. In: *Electr. Notes Theor. Comput. Sci.* 172 (2007), pp. 33–67.

[3] S. Abramsky. “Sequentiality vs. Concurrency in Games and Logic”. In: *Mathematical. Structures in Comp. Sci.* 13.4 (2003), pp. 531–565.

[4] S. Abramsky and R. Duncan. “A categorical quantum logic”. In: *Mathematical Structures in Computer Science*. Vol. 16. 3. 2006, pp. 469–489.

[5] S. Abramsky and R. Jagadeesan. “Games and Full Completeness for Multiplicative Linear Logic”. In: *J. Symb. Log.* 59.2 (1994), pp. 543–574.

[6] S. Abramsky and R. Jagadeesan. “New Foundations for the Geometry of Interaction”. In: *Proceedings of the Seventh Annual Symposium on Logic in Computer Science*. 1992, pp. 211–222.

[7] S. Abramsky, R. Jagadeesan, and P. Malacaria. “Full abstraction for PCF”. In: *Information and Computation* 163.2 (1996), pp. 409–470.

[8] S. Abramsky and G. McCusker. “Call-by-Value Games”. In: *Selected Papers from the11th International Workshop on Computer Science Logic*. CSL ’97. Springer-Verlag, 1998, pp. 1–17.

[9] S. Abramsky and G. McCusker. “Game Semantics”. In: *Computational Logic: Proceedings of the 1997 Marktoberdorf Summer School*. Ed. by H. Schwichtenberg and U. Berger. Springer-Verlag, 1999, pp. 1–56.

[10] S. Abramsky and P-A. Melliès. “Concurrent Games and Full Completeness”. In: *14th Annual IEEE Symposium on Logic in Computer Science*. 1999, pp. 431–442.

[11] J-M. Andreoli. “Logic Programming with Focusing Proofs in Linear Logic”. In: *J. Log. Comput.* 2.3 (1992), pp. 297–347.

[12] P. Baillot et al. “Timeless games”. In: *Computer Science Logic: CSL ’97*. Ed. by Mogens Nielsen and Wolfgang Thomas. 1998, pp. 56–77.

[13] E. S. Bainbridge et al. “Functorial Polymorphism”. In: *Theor. Comput. Sci.* 70 (1990).

[14] P. Di Giam Berardino. "Proof nets and semantics: coherence and acyclicity". Institut de Mathematiques de Luminy, Marseille, France.

[15] A. Blass. "A Game Semantics for Linear Logic". In: *Ann. Pure Appl. Logic* 56.1-3 (1992), pp. 183–220.

[16] R. Blute, M. Hamano, and P. J. Scott. "Softness of hypercoherences and MALL full completeness". In: *Ann. Pure Appl. Logic* 131.1-3 (2005), pp. 1–63.

[17] M. Bojanczyk. *Slightly infinite sets*. Tech. rep. 2016.

[18] P. Boudes. "Hypercoherences et Jeux". PhD thesis. Université de la Mediterranée - Aix Marseille II, 2002.

[19] L. Caires and F. Pfenning. "Session Types As Intuitionistic Linear Propositions". In: *Proceedings of the 21st International Conference on Concurrency Theory*. CONCUR'10. Springer-Verlag, 2010, pp. 222–236.

[20] P.-H. Chu. "Constructing star-autonomous categories. M. Sc. thesis". 1978.

[21] P-L. Curien and H. Herbelin. "Abstract machines for dialogue games". In: *CoRR* abs/0706.2544 (2007).

[22] V. Danos, H. Herbelin, and L. Regnier. "Game semantics and abstract machines". In: *Proceedings 11th Annual IEEE Symposium on Logic in Computer Science*. 1996, pp. 394–405.

[23] V. Danos and L. Regnier. "The structure of multiplicatives". In: *Archive for Mathematical Logic* 28.3 (1989), pp. 181–203.

[24] H. Devarajan et al. "Full Completeness of the Multiplicative Linear Logic of Chu Spaces". In: *14th Annual IEEE Symposium on Logic in Computer Science*. 1999, pp. 234–243.

[25] Hughes D.J.D, R. Houston, and A. Schalk. "Modelling Linear Logic without Units (Preliminary Results)".

[26] T. Ehrhard. "A new correctness criterion for MLL proof nets". In: *CSL-LICS '14*. 2014, 38:1–38:10.

[27] T. Ehrhard. "Hypercoherences: A Strongly Stable Model of Linear Logic". In: *Mathematical Structures in Computer Science*. Cambridge University Press, 1993.

[28] T. Ehrhard. "Parallel and serial hypercoherences". In: *Theoretical Computer Science* 247.1 (2000), pp. 39–81.

[29] T. Ehrhard. "Projecting sequential algorithms on strongly stable functions". In: *Annals of Pure and Applied Logic* 77.3 (1996), pp. 201–244.

[30] M. Gabbay and D. R. Ghica. "Game Semantics in the Nominal Model". In: *Electr. Notes Theor. Comput. Sci.* 286 (2012), pp. 173–189.

[31] M. Gabbay and A. M. Pitts. "A New Approach to Abstract Syntax with Variable Binding". In: *Formal Asp. Comput.* 13.3-5 (2002), pp. 341–363.

[32] M. J. Gabbay. “A study of substitution, using nominal techniques and Fraenkel-Mostowksi sets”. In: *Theor. Comput. Sci.* 410.12-13 (2009), pp. 1159–1189.

[33] J-Y. Girard. “Linear Logic”. In: *Theor. Comput. Sci.* 50 (1987), pp. 1–102.

[34] J-Y. Girard. “Linear Logic: Its Syntax and Semantics”. In: *Proceedings of the Workshop on Advances in Linear Logic* (1995), pp. 1–42.

[35] J-Y. Girard. “Locus Solum: From the Rules of Logic to the Logic of Rules”. In: *Mathematical. Structures in Comp. Sci.* 11.3 (June 2001), pp. 301–506.

[36] J-Y. Girard. “Proof-nets: The parallel syntax for proof-theory”. In: *Logic and Algebra*. Marcel Dekker, 1996, pp. 97–124.

[37] J-Y. Girard. *The blind spot – lectures on logic*. European Mathematical Society, 2011.

[38] J-Y. Girard. “Towards a Geometry of Interaction”. In: (1989). 69-108.

[39] J-Y. Girard, P. Taylor, and Y. Lafont. *Proofs and Types*. Cambridge University Press, 1989.

[40] R. J. van Glabbeek and D. J. D. Hughes. “MALL proof nets identify proofs modulo rule commutation”. In: *CoRR* abs/1609.04693 (2016).

[41] J. Guilhem and T. Nikos. “Trace Semantics for Polymorphic References”. In: LICS ’16 (2016), pp. 585–594.

[42] M. Hamano. “Softness of MALL proof-structures and a correctness criterion with Mix”. In: *Archive for Mathematical Logic* 43.6 (2004), pp. 751–794.

[43] K. Honda and N. Yoshida. “Game-theoretic analysis of call-by-value computation”. In: *Theoretical Computer Science* 221.1–2 (1999), pp. 393–456.

[44] R. Houston. “Finite products are biproducts in a compact closed category”. In: *Journal of Pure and Applied Algebra* 212 (2008), pp. 394–400. ISSN: 0022-4049.

[45] D. J. D. Hughes. “Games and Definability for System F”. In: *Proceedings, 12th Annual IEEE Symposium on Logic in Computer Science*. 1997, pp. 76–86.

[46] D. J. D. Hughes and R. J. Van Glabbeek. “Proof Nets for Unit-free Multiplicative-Additive Linear Logic”. In: *In 18th IEEE Symposium on Logic in Computer Science (LICS 2003*. 2003, pp. 1–10.

[47] D. J.D. Hughes. “Simple free star-autonomous categories and full coherence”. In: *Journal of Pure and Applied Algebra* 216.11 (2012), pp. 2386 –2410.

[48] D. J.D. Hughes. “Simple multiplicative proof nets with units”. In: *arXiv* (2005). arXiv:math/0507003.

[49] J. Martin E. Hyland and C.-H. Luke Ong. “On full abstraction for PCF: I, II, III”. In: *Information and Computation* 163.2 (2000), pp. 285–408.

[50] J.M.E. Hyland and C.H.L. Ong. “Fair games and full completeness for multiplicative linear logic without the mix-rule”. In: *preprint* 190 (1993).

[51] M. Hyland and A. Schalk. “Abstract Games for Linear Logic Extended Abstract”. In: *Electronic Notes in Theoretical Computer Science* 29 (1999), pp. 127–150.

[52] M. Hyland and A. Schalk. "Games on Graphs and Sequentially Realizable Functionals". In: *17th IEEE Symposium on Logic in Computer Science (LICS 2002)*. 2002, pp. 257–264.

[53] M. Hyland and A. Schalk. "Glueing and orthogonality for models of linear logic". In: *Theoretical Computer Science* 294.1 (2003), pp. 183–231.

[54] A. Joyal, R. Street, and D. Verity. "Traced monoidal categories". In: *Mathematical Proceedings of the Cambridge Philosophical Society* 119.3 (Apr. 1996), pp. 447–468.

[55] G. Kahn and D. Macqueen. *Coroutines and Networks of Parallel Processes*. Research Report. 1976, p. 20.

[56] J. Laird. "A game semantics of names and pointers". In: *Annals of Pure and Applied Logic* 151 (2008). GaLoP '05: First Games for Logic and Programming Languages Workshop (post-proceedings), pp. 151–169.

[57] J. Laird. "Game Semantics for Call-by-Value Polymorphism". In: *Proceedings of ICALP*. Ed. by Samson Abramsky et al. Vol. 6199. Lecture Notes in Computer Science. Springer, 2010, pp. 187–198.

[58] S. M. Lane. *Categories for the Working Mathematician*. Springer-Verlag, New York, 1998.

[59] O. Laurent. *A proof of the focalization property of linear logic*. 2004.

[60] R. Loader. "Linear Logic, Totality and Full Completeness". In: *Proceedings of the Ninth Annual Symposium on Logic in Computer Science*. 1994, pp. 292–298.

[61] P. Lorenzen. "Ein dialogisches konstruktivitätskriterium". In: *Infinitistic Methods: Proceed. Symp. Foundations of Math* (1961), pp. 193–200.

[62] P. Lorenzen. "Logik Und Agon". In: *Atti Del XII Congresso Internazionale di Filosofia* 4 (1960), pp. 187–194.

[63] P-A. Melliès. "Asynchronous games 1: Uniformity by group invariance". 2003.

[64] P-A. Melliès. "Asynchronous games 2: The true concurrency of innocence". In: *Theoretical Computer Science* 358.2–3 (2006). (CONCUR 2004), pp. 200–228.

[65] P-A. Melliès. "Asynchronous Games 3 An Innocent Model of Linear Logic". In: *Electronic Notes in Theoretical Computer Science* 122 (2005), pp. 171–192.

[66] P-A. Mellies. "Asynchronous Games 4: A Fully Complete Model of Propositional Linear Logic". In: *Proceedings of the 20th Annual IEEE Symposium on Logic in Computer Science*. LICS '05. IEEE Computer Society, 2005, pp. 386–395.

[67] P-A. Melliès. "Categorical semantics of linear logic". In: *Interactive Models of Computation and Program Behaviour, Panoramas et Synthèses 27, Société Mathématique de France 1–196*. 2009.

[68] P-A. Melliès. "Sequential algorithms and strongly stable functions". In: *Theoretical Computer Science* 343.1 (2005), pp. 237–281.

[69] P-A. Melliès. *Une étude micrologique de la négation, Habilitation à Diriger des Recherches*. 2016.

[70] P-A Melliès and S. Mimram. "Asynchronous Games: Innocence Without Alternation". In: *CONCUR 2007 - Concurrency Theory, 18th International Conference*. 2007, pp. 395–411.

[71] P-A. Melliès and N. Tabareau. "Resource modalities in tensor logic". In: *Ann. Pure Appl. Logic* 161.5 (2010), pp. 632–653.

[72] P-A. Melliès, N. Tabareau, and C. Tasson. "An Explicit Formula for the Free Exponential Modality of Linear Logic". In: *Automata, Languages and Programming, 36th Internatilonal Colloquium, ICALP*. 2009, pp. 247–260.

[73] Paul-André Melliès. "Game Semantics in String Diagrams". In: *LICS 2012*. 2012, pp. 481–490.

[74] S. Mimram. "Sémantique des jeux asynchrones et réécriture 2-dimensionnelle". PhD thesis. Paris Diderot, 2009.

[75] E. Moggi. "Notions of Computation and Monads". In: *Information and Computation* 93.1 (1991), pp. 55–92.

[76] A. S. Murawski and N. Tzevelekos. "Algorithmic Games for Full Ground References". In: *ICALP*. 2012, pp. 312–324.

[77] A. S. Murawski and N. Tzevelekos. "Game Semantics for Good General References". In: *Proceedings of LICS*. IEEE Computer Society, 2011, pp. 75–84.

[78] A. S. Murawski and N. Tzevelekos. "Game semantics for interface middleweight Java". In: *Proceedings of POPL*. Ed. by Suresh Jagannathan and Peter Sewell. ACM, 2014, pp. 517–528.

[79] A. S. Murawski and N. Tzevelekos. "Game Semantics for Nominal Exceptions". In: *FOSSACS 2014*. 2014, pp. 164–179.

[80] H. Nickau. "Hereditarily Sequential Functionals". In: *Proceedings of the Third International Symposium on Logical Foundations of Computer Science*. LFCS '94. Springer-Verlag, 1994, pp. 253–264.

[81] A. M. Pitts. "Nominal logic, a first order theory of names and binding". In: *Information and Computation* 186.2 (2003). (TACS 2001), pp. 165–193.

[82] A. M. Pitts. *Nominal Sets: Names and Symmetry in Computer Science*. Vol. 57. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2013. ISBN: 9781107017788.

[83] C. Retoré. "A semantic characterisation of the correctness of a proof net". In: *Mathematical Structures in Computer Science* 7.05 (1997), pp. 445–452.

[84] A. Schalk and H. Steele. "Constructing Fully Complete Models for Multiplicative Linear Logic". In: *Proceedings of the 2012 27th Annual IEEE/ACM Symposium on Logic in Computer Science*. LICS '12. 2012.

[85] T. Seiller. "Interaction Graphs: Full Linear Logic". In: *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*. 2016, pp. 427–436.

[86] M. Shirahata. *A Sequent Calculus for Compact Closed Categories*. 1996.

[87] P. Tranquilli. "A Characterization of Hypercoherent Semantic Correctness in Multiplicative Additive Linear Logic". In: *Proceedings of the 22Nd International Workshop on Computer Science Logic*. CSL '08. Springer-Verlag, 2008, pp. 246–261.

[88] D. C. Turner. "Nominal domain theory for concurrency". PhD thesis. University of Cambridge, UK, 2009.

[89] N. Tzevelekos. "Full abstraction for nominal general references". In: *Logical Methods in Computer Science* 5.3 (2009).

[90] N. Tzevelekos. "Nominal Game Semantics". PhD thesis. University of Oxford, 2008.

[91] N. Tzevelekos and S. Abramsky. *Introduction to categories and categorical semantics*. 2008.

[92] P. Wadler. "Propositions As Sessions". In: *SIGPLAN Not.* 47.9 (2012), pp. 273–286.