ISSN 1470-5559

Reasoning about Action: Ray Reiter meets Fibred Categories

Graham White

Queen Mary
University of London
Department of Computer Science

RR-05-12 August 2005

Reasoning about Action: Ray Reiter meets Fibred Categories

Graham White
Department of Computer Science
Queen Mary, University of London
graham@dcs.qmul.ac.uk
http://www.dcs.qmul.ac.uk/~graham

August 27, 2005

Contents

List of Tables

0 Motivation

This paper is concerned with the frame problem – that is, the problem of logical reasoning about action and change, and, in particular, the problem of correctly reasoning about facts which a particular action leaves unchanged. We will concentrate on Reiter's work: it is insightful, and mathematically interesting.

0.1 Reiter

Ray Reiter [16] argues that, for a solution to the frame problem, what will suffice is to be able to solve the following three subproblems:

preconditions given an action α and a state s, find out whether the action can be executed in s, giving a transition $s \overset{\alpha}{\frown} t$, for some suitable state t.

progression given an action $s \overset{\alpha}{\frown} t$, and given a theory Ξ_s describing the state s, find the theory Ξ_t describing t.

regression given a transition as before, and given a proposition P, find a proposition P' which will be true at s iff P is true at t.

These three problems are fundamental, in the following sense. Given a theory describing the initial state, it is clear that we could then use the preconditions, and the solution to the progression problem, to list all the attainable states and their corresponding theories. But it turns out that we can, in turn, use a solution for the regression problem to solve the progression problem: [16, ch. 9] so the regression problem is, in fact, the more fundamental. It is also, as it happens, far easier to solve: the realisation that this is so is an important insight due to Reiter [15], Pednault [13, 14] and others.

This paper, then, will describe how to solve Reiter's problems, giving a solution to the regression problem first. Our machinery will allow us to give a more *general* – and, hopefully, a more perspicuous – solution than Reiter's.

0.2 Our Approach

Intuitively, our machinery will be as follows. Suppose that we have some sort of representation of states and actions: for definiteness, we could think of a labelled transition system, where the nodes represent states and the transitions represent actions. States can be total or partial: we can think of partial states as representing, nondeterministically, sets of total states.

We will distinguish between what philosophers call action *types* and action *tokens* (see Davidson [5, 4], Hornsby [7, 6], and Wetzel [21]). An action type (for example, the action of standing up) can be instantiated more than once as an action token. In the context of a transition system, we will label transitions with the action *types*, with individual transitions being the tokens.

For the purposes of AI we want something less abstract than merely a labelled transition system: specifically, we would like to be able to represent, and calculate with, this labelled transition system using the resources of formal logic. Specifically, we will give ourselves a language $\mathfrak{L}$, an entailment relation, and, for each node (i.e. state), we will want to find a theory, in the language, which represents the propositions true in the corresponding state. This theory should be categorical in the deterministic case, but, in general, it will not be. The essence of the frame problem, of course, is to be able to calculate "what is true" in a particular state: in more realistic terms, it is to determine what the corresponding theory is.

0.3 Our Relation with Orthodoxy

0.3.1 The Unique Names Assumption

Choices have been made here which are far from orthodox. Conceptually, we are taking the idea of a labelled transition system as primary: this is in sharp contrast with most work in AI [16, p. 83], which works with what the AI community calls "situations" – namely, sequences of actions starting from the initial state – and assumes that any two such situations are distinct. In a labelled transition system, by contrast, states – that is, "snapshots" of the system – are primary, and a single state may be attained by more than one sequence of actions. Representations such as ours – apart from their use in McCarthy and Hayes' early work [11] – are comparatively rare.

Our representation, although unorthodox, has important consequences. It allows eventualities in which, for example, the state attained by a non-trivial sequence of actions could be the same as the initial state. This means that, in general, states do not have canonical descriptions: in terms of the usual terminology, we are not making the unique names assumption for states. However, this is not necessarily a bad thing: as Reiter observes [16, pp. 94ff.], the unique names assumption is clearly not universally true of the real world.

Furthermore, the unique names assumption is mathematically difficult to handle: it amounts to insisting on the initial model of the relevant theory, and this is, in many circumstances, an artificial thing to do. We can, of course, always recover, from a labelled transition system, a tree of situations: all we have to do is to consider the set of paths from the initial state. The two will be, in a very strong sense, mathematically equivalent (they will be equivalent up to bisimulation [12, 19, 18, 9]): this equivalence is enough to ensure that, if we are concerned only with how these mathematical objects describe the world of states and actions, nothing will be lost by considering one rather than the other, and it will be mathematically more natural to consider the more general (that is, non-initial) model.

0.3.2 Fibrations

The other contentious choice is that we shall be working, as it were, locally: our formalism will give us, for each state, a logical theory, which describes what is

true *in that state or set of states*, but no more. By contrast, most work in AI considers global theories which describe all possible situations. These theories are typically expressed in untyped first-order languages, which have individuals corresponding to states and to reified predicates, and relations which describe the fact that a predicate holds in a situation. We can, using such languages, express extremely arcane relationships between situations in different states.

As with our other contentious choice, this can be justified. Our choice is notationally simpler: a global theory must do a good deal of indexing, in order to differentiate what is true in one situation from what is true in another. This extra notation can be quite overwhelming, as Reiter implicitly admits [16, pp. 117ff.]. Furthermore, one of the things that we frequently want to do, in these formalisms, is to say that a proposition is *about* a particular situation: this is, in the standard AI formalisms, a semantic rather than a syntactic matter (since situations are simply individuals alongside other individuals, and can be referred to using terms of arbitrary complexity). Consequently, one must, in the standard formalisms, resort to rather arbitrary definitions to try to express what it is for a proposition to be about a situation (see [16, pp. 45ff.]). Our formalism is much more straightforward: propositions are typed, their types are states, and a proposition is about a particular state if it has that type.

It might, one might think, be argued that our choice, although notationally simpler, is less expressive: we are limited to talking of one state at a time. However, this is not so. In addition to our descriptions of states, we will have Reiter's regression operators: if we have a transition between states s and t, there will be a regression operator, mapping propositions at t to propositions at s. And we can, given regression operators, use the machinery of fibred category theory to recover, from the local data, a suitable global logic [8, 1]; they are equivalent in the sense that systematic and natural translations can be given between one way of doing things and the other. And we can, in practice, express all that we need to in this global theory. So, despite appearances, we have not lost anything, and we have gained a great deal of notational – and, indeed, conceptual – clarity.

0.4 Definitions

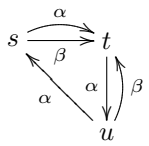
0.4.1 Transition Systems

Definition 1. A *labelled transition system* consists of

1. a set Σ of *states*,
2. a set A of *labels*,
3. a set $\mathcal{T} \subset A \times \Sigma \times \Sigma$ of *transitions*

Intuitively, each transition is given by specifying its label, its source, and its target: this triple gives us an element of $A \times \Sigma \times \Sigma$.

We will often draw labelled transition systems thus:



this will say that there are α-labelled transitions from s to t, from t to u, and from u to s, and β-labelled transitions from s to t and u to t; similarly, we will draw individual transitions thus

Remark 1. In practice, transitions will be action tokens, and the labels attached to those transition will be action types: will will, consequently, often refer to both transitions and labels as actions. This may seem to be ambiguous, but is, in practice, not so: ordinary language is good at making implicit type-token distinctions, and this will usually be sufficient for clarity.

Remark 2. Our definition of labelled transition systems – which is the standard one – does not rule out indeterminacy: thus, there may be more than one transition with the same starting point and the same label. However, we will work from now on with *deterministic* transition systems.

0.4.2 The Associated Category

Definition 2. Given a deterministic transition system, as above, the *associated category* has for objects the states of the system: a morphism between nodes s and t will be a composable sequence of transitions (which may have zero length). This morphism will be labelled with the sequence of the labels: the associated category is thus enriched in the free monoid on the set of labels.

1 Reiter's Approach

In this section we will describe the specifics of Reiter's approach.

1.1 Terminology and the Correspondence with Our System

As is standard in the area, Reiter uses a terminology of *situations* and *fluents*:

> a possible world history, which is simply a sequence of actions, is represented by a first-order term called a *situation*...
>
> Generally, the values of relations and functions in a dynamic world will vary from one situation to the next. Relations whose truth values vary from situation to situation are called *relational*

> *fluents*. They are denoted by predicate symbols taking a situation term as their last argument. Similarly, functions whose truth values vary from situation to situation are called *functional fluents*, and are denoted by functional symbols taking a situation term as their last argument. [16, p. 19]

Reiter's situations, then, are histories, whereas the objects in our category are instantaneous states. There may seem to be a conflict with our system here, but very little, in fact, depends on it: in practice, Reiter's fluents only depend on the last state in a history. Relational and functional fluents, evaluated at a history with last state s, will correspond, in our system, to relations or functions with the appropriate signature and with type s. Reiter furthermore works with a distinguished initial situation S_0: since, in his formalisation, situations are identical with their histories, this means that his situations form a tree with root node S_0.

1.1.1 Syntax

Reiter's language is many-sorted, with sorts for actions, situations and individuals. Situations are thus individuals in his language, of sort *situation*: his language, correspondingly, has a constant S_0, of sort *situation*, together with a function symbol $\mathsf{do} : \mathit{action} \times \mathit{situation} \to \mathit{situation}$. Situations are generated by successively applying actions to the initial situation: this corresponds on a syntactic level to his stipulations about the tree structure of the set of situations. [16, pp. 47ff.] Fluents are predicates, which have arguments of sort *situation*: a fluent will hold at a particular situation iff the corresponding predicate is satisfied by the situation.

The syntax is as follows [16, pp. 47ff.].

Definition 3. Define the language $\mathfrak{L}_{sitcalc}$ by the following clauses.

$$\langle \mathit{situation} \rangle ::= \mathsf{S}_0 \mid \mathsf{do}(\langle \mathit{action} \rangle, \langle \mathit{situation} \rangle) \mid s$$

$$\langle \mathit{action} \rangle ::= b(\langle \mathit{action} \rangle_1, \ldots, \langle \mathit{action} \rangle_k, \langle \mathit{object} \rangle_1, \ldots, \langle \mathit{object} \rangle_l) \mid \beta(\langle \mathit{action} \rangle_1, \ldots, \langle \mathit{action} \rangle_k, \langle \mathit{object} \rangle_1, \ldots, \langle \mathit{object} \rangle_l, \langle \mathit{situation} \rangle) \mid a$$

$$\langle \mathit{object} \rangle ::= t(\langle \mathit{action} \rangle_1, \ldots, \langle \mathit{action} \rangle_k, \langle \mathit{object} \rangle_1, \ldots, \langle \mathit{object} \rangle_l) \mid \tau(\langle \mathit{action} \rangle_1, \ldots, \langle \mathit{action} \rangle_k, \langle \mathit{object} \rangle_1, \ldots, \langle \mathit{object} \rangle_l, \langle \mathit{situation} \rangle) \mid x$$

$$\begin{aligned}\langle \mathit{proposition} \rangle ::= {} & p(\langle \mathit{action} \rangle_1, \ldots, \langle \mathit{action} \rangle_k, \langle \mathit{object} \rangle_1, \ldots, \langle \mathit{object} \rangle_l) \mid \\ & \phi(\langle \mathit{action} \rangle_1, \ldots, \langle \mathit{action} \rangle_k, \langle \mathit{object} \rangle_1, \ldots, \langle \mathit{object} \rangle_l, \langle \mathit{situation} \rangle) \mid \\ & \langle \mathit{situation} \rangle \sqsubset \langle \mathit{situation} \rangle \mid \mathsf{Poss}(\langle \mathit{situation} \rangle) \mid \\ & \langle \mathit{situation} \rangle = \langle \mathit{situation} \rangle \mid \langle \mathit{action} \rangle = \langle \mathit{action} \rangle \mid \langle \mathit{object} \rangle = \langle \mathit{object} \rangle \mid \\ & X(\langle \mathit{action} \rangle_1, \ldots, \langle \mathit{action} \rangle_k, \langle \mathit{object} \rangle_1, \ldots, \langle \mathit{object} \rangle_l) \mid \\ & Y(\langle \mathit{action} \rangle_1, \ldots, \langle \mathit{action} \rangle_k, \langle \mathit{object} \rangle_1, \ldots, \langle \mathit{object} \rangle_l, \langle \mathit{situation} \rangle) \mid \\ & \langle \mathit{proposition} \rangle \wedge \langle \mathit{proposition} \rangle \mid \neg \langle \mathit{proposition} \rangle \mid \\ & \exists s.p(s) \mid \exists a.p(a) \mid \exists x.p(x) \mid \exists X.p(X) \mid \exists Y.p(Y)\end{aligned}$$

Here s, a, and x stand for first-order variables of the appropriate sorts (there are countably many variables of each sort). Similarly, b, t, p, and X stand

for non-fluent function symbols of sorts *action* and *object*, non-fluent predicate symbols, and non-fluent propositional variables, whereas β, τ, ϕ and Y stand for functional fluents of sorts *action* and *object*, relational fluents, and fluent propositional variables.

We can (although Reiter does not do this explicitly) define a sublanguage $\underline{\mathfrak{L}}$ by leaving out the fluents; this language will be the language of "constant formulae", i.e. those formulae which take the same value in all situations.

Definition 4. Define the language $\underline{\mathfrak{L}}$ by the following clauses.

$\langle action \rangle ::= b(\langle action \rangle_1, \ldots, \langle action \rangle_k, \langle object \rangle_1, \ldots, \langle object \rangle_l) \mid a$
$\langle object \rangle ::= t(\langle action \rangle_1, \ldots, \langle action \rangle_k, \langle object \rangle_1, \ldots, \langle object \rangle_l) \mid x$
$\langle proposition \rangle ::= p(\langle action \rangle_1, \ldots, \langle action \rangle_k, \langle object \rangle_1, \ldots, \langle object \rangle_l) \mid$
$\langle action \rangle = \langle action \rangle \mid \langle object \rangle = \langle object \rangle \mid$
$X(\langle action \rangle_1, \ldots, \langle action \rangle_k, \langle object \rangle_1, \ldots, \langle object \rangle_l) \mid$
$\langle proposition \rangle \wedge \langle proposition \rangle \mid \neg \langle proposition \rangle \mid$
$\exists s.p(s) \mid \exists x.p(x) \mid \exists X.p(X)$

(The symbols have the same meanings as in the previous definition.)

We can also define the sublanguage of terms *uniform in a situation* [16, p. 58]:

Definition 5. Let σ be a term of sort *situation*. Define the sublanguage $\mathfrak{L}_\sigma$ by the clauses:

$\langle action \rangle ::= \underline{\alpha} \mid \alpha(\langle action \rangle_1, \ldots, \langle action \rangle_k, \langle object \rangle_1, \ldots, \langle object \rangle_l, \sigma)$
$\langle object \rangle ::= \underline{\tau} \mid \tau(\langle action \rangle_1, \ldots, \langle action \rangle_k, \langle object \rangle_1, \ldots, \langle object \rangle_l, \sigma)$
$\langle proposition \rangle ::= \underline{\phi} \mid \phi(\langle action \rangle_1, \ldots, \langle action \rangle_k, \langle object \rangle_1, \ldots, \langle object \rangle_l, \sigma) \mid$
$\neg \langle proposition \rangle \mid \langle proposition \rangle \wedge \langle proposition \rangle \mid \exists a.p(a), \exists x.p(x)$

Here $\underline{\alpha}$ and $\underline{\tau}$ are terms of the appropriate sort from $\underline{\mathfrak{L}}$: $\underline{\phi}$ is a proposition from $\underline{\mathfrak{L}}$.

Remark 3. Reiter stipulates that his predicates and function symbols should have arities of the form $(\langle action \rangle \cup \langle object \rangle)^n$. This is ambiguous: it could mean that each argument place is determinately of sort $\langle action \rangle$ or of sort $\langle object \rangle$, or that each argument place can is of a new sort, $\langle action \rangle \cup \langle object \rangle$. I have chosen the first alternative: it is more probable, and, in any case, there is no way of using predicates, or function symbols, of the second sort which cannot be disambiguated into the use of predicate symbols of the first sort, since all terms of the language, *even the variables*, are determinately of one sort or the other.

Remark 4. Reiter stipulates that predicate or function symbols have *at most one* argument place of sort $\langle situation \rangle$. This condition is clearly not preserved under the logical operators: for example, if s_1 and s_2 are variables of sort situation, then $p(s_1) \wedge q(s_2)$ has two argument places of that sort. Similarly, propositions of the form $s_1 \sqsubset s_2$ and $s_1 = s_2$ have more than one $\langle situation \rangle$ argument place.

We finally define the language of *regressible* propositions [16, p. 62]: these are, roughly speaking, the propositions which are uniform in situations specified by a sequence of actions, starting from the initial situation.

Definition 6. If $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k)$ is a sequence of actions, and if s is a situation, let

$$\mathsf{do}(\boldsymbol{\alpha}, s) = \mathsf{do}(\alpha_1, \mathsf{do}(\ldots \mathsf{do}(\alpha_k, s) \ldots))$$

The *regressible propositions*, $\mathfrak{L}^{regr}$, are the closure under the logical connectives of

$$\bigcup_{\boldsymbol{\alpha}} \left(\mathfrak{L}_{\mathsf{do}(\boldsymbol{\alpha}, \mathsf{S}_0)} \cup \bigcup \mathsf{Poss}(b(\alpha'_1, \ldots, \alpha'_k, t_1, \ldots, t_l), \mathsf{do}(\boldsymbol{\alpha}, \mathsf{S}_0)) \right)$$

Remark 5. Reiter is not quite consistent in his definition of uniform formulae: in [16, Definition Definition 4.4.1, p. 58], he gives a BNF-style definition which does not include propositional variables: but he then goes on to say, less formally, that "a formula of $\mathfrak{L}_{sitcalc}$ is uniform in σ if it does not ...", and giving a series of exclusions which does not exclude propositional variables. I have chosen the stricter definition, and have also aligned my definition of regressible formulae with that of the uniform formulae.

1.1.2 Action Theories

Given this language, Reiter then defines what he calls an *basic action theory*, $\mathcal{D}$: this is a theory which axiomatises the behaviour of the system. It has several components.

Foundational Axioms for Situations These axiomatise the fact that the situations form a tree, freely generated by the primitive axioms:

$$\mathsf{do}(a_1, s_1) = \mathsf{do}(a_2, s_2) \rightarrow a_1 = a_2 \wedge s_1 = s_2 \tag{1}$$

$$\forall X. X(\mathsf{S}_0) \wedge (\forall a, s. (X(s) \rightarrow X(\mathsf{do}(a, s)))) \rightarrow \forall X. \forall s. X(s) \tag{2}$$

and we also axiomatise the notion of a prehistory:

$$\neg(s \sqsubset \mathsf{S}_0) \tag{3}$$

$$s \sqsubset \mathsf{do}(a, s') \leftrightarrow s \sqsubseteq s' \tag{4}$$

This group of axioms, (1)–(4), will be referred to as $\boldsymbol{\Sigma}$ [16, p. 50.].

Unique Names for Actions We suppose the following unique names axioms for actions [16, p. 31]:

$$\alpha(a_1, \ldots, a_k, x_1, \ldots, x_l) \neq \alpha'(a'_1, \ldots, a'_k, x'_1, \ldots, x'_l) \tag{5}$$

for distinct function symbols α, α'

$$\alpha(a_1, \ldots, a_k, x_1, \ldots, x_l) = \alpha(a'_1, \ldots, a'_k, x'_1, \ldots, x'_l) \\ \rightarrow (\bigwedge_{i=1}^{k} a_i = a'_i) \wedge (\bigwedge_{i=1}^{l} x_i = x'_i) \tag{6}$$

The collection of all axioms (5) and (6) will be referred to as $\mathcal{D}_{una}$ [16, p. 31].

Action Precondition Axioms We suppose that the predicate Poss is defined by axioms of the following form, one for each action function symbol A:

$$\mathsf{Poss}(A(a_1, \ldots, a_k, x_1, \ldots, x_l, s), s) \leftrightarrow \Pi_A(a_1, \ldots, a_k, x_1, \ldots, x_l, s) \tag{7}$$

where the a_i are action variables, the x_i are object variables, s is a state variable, and $\Pi_A \in \mathfrak{L}_s$.

The collection of all of the axioms (7) will be referred to as $\mathcal{D}_{ap}$ [16, p. 59].

Successor State Axioms We assume axioms of the following form:

$$F(a_1, \ldots, a_k, x_1, \ldots, x_l, \mathsf{do}(a, s)) \leftrightarrow \Phi_F(a_1, \ldots, a_k, x_1, \ldots, x_l, a, s) \tag{8}$$

for each relational fluent F, where $a, a_1, \ldots, a_k$ are action variables, $x_1, \ldots, x_l$ are object variables, s is a situation variable, and $\Phi_F \in \mathfrak{L}_s$,

$$\alpha(a_1, \ldots, a_k, x_1, \ldots, x_l, \mathsf{do}(a, s)) = a \leftrightarrow A_\alpha(a_1, \ldots, a_k, x_1, \ldots, x_l, a, s) \tag{9}$$

for each action-valued functional fluent α, where $a, a_1, \ldots, a_k$ are action variables, $x_1, \ldots, x_l$ are object variables, s is a situation variable, and $A_\alpha \in \mathfrak{L}_s$,

$$\tau(a_1, \ldots, a_k, x_1, \ldots, x_l, \mathsf{do}(a, s)) = a \leftrightarrow \Phi_\tau(a_1, \ldots, a_k, x_1, \ldots, x_l, a, s) \tag{10}$$

for each object-valued functional fluent τ, where $a, a_1, \ldots, a_k$ are action variables, $x_1, \ldots, x_l$ are object variables, s is a situation variable, and $\Phi_\tau \in \mathfrak{L}_s$.

The collection of all the axioms (8)–(10) will be referred to as $\mathcal{D}_{ss}$ [16, pp. 59f.].

The Initial State We assume that we have a set of first-order sentences $\mathcal{D}_{\mathsf{S}_0} \subseteq \mathfrak{L}_{\mathsf{S}_0}$: these will axiomatise the initial situation [16, p. 60.].

Definition 7. A collection of axioms of the form

$$\mathbf{\Sigma} \cup \mathcal{D}_{una} \cup \mathcal{D}_{ap} \cup \mathcal{D}_{ss} \cup \mathcal{D}_{\mathsf{S}_0}$$

will be called a *basic action theory* provided that the right hand sides A_α and Φ_τ of (9) and (10) satisfy

$$\mathcal{D}_{una}, \mathcal{D}_{\mathsf{S}_0} \vdash A_\alpha \text{ defines a function} \tag{11}$$

and

$$\mathcal{D}_{una}, \mathcal{D}_{\mathsf{S}_0} \vdash \Phi_\tau \text{ defines a function} \tag{12}$$

(with the standard expansions of what it is for a relation to be functional, written out in full in [16, p. 60]).

1.2 The Regression Operator

1.2.1 Reiter's Definition

Given this language, one of Reiter's goals is to define what he calls a *regression operator*: that is, an operator R on a suitable class of propositions (the *regressible propositions* of Definition 6). R will take any regressible proposition into a proposition in $\mathfrak{L}_{\mathsf{S}_0}$. Reiter defines the regression operator R inductively, starting with terms $F(s)$, where $F(\cdot)$ is a fluent fluent atom, and s is a situation of the form $\mathsf{do}([\alpha_1, \ldots, \alpha_n], \mathsf{S}_0)$. If $n = 0$, then we are already at the initial situation, and we let $R(F(\mathsf{S}_0)) = F(\mathsf{S}_0)$. Otherwise, $s = \mathsf{do}(\alpha_1, s')$, for some ground situation s'; so, by the successor state axiom for F (8),

$$\mathcal{D}_{ss} \vdash F(s) \leftrightarrow \Phi(\mathsf{do}([\alpha_2, \ldots, \alpha_n], \mathsf{S}_0))$$

So we let $R(F(s)) = R(\Phi(\mathsf{do}([\alpha_2, \ldots, \alpha_n], \mathsf{S}_0)))$; note that $F_s \in \mathfrak{L}_{(}s)$ and $R(F(s)) \in \mathfrak{L}_{(}s')$, and $s' \sqsubset s$. Finally, if P is logically complex, we define R compositionally [16, pp. 64f.]:

$$R(\neg W) = \neg R(W) \tag{13}$$

$$R(W_1 \wedge W_2) = R(W_1) \wedge R(W_2) \tag{14}$$

$$R(\exists x.W) = \exists x.R(W) \tag{15}$$

An induction jointly over the logical complexity of formulae, and over the tree of situations, shows that R is well defined, and that $R(P) \in \mathfrak{L}_{\mathsf{S}_0}$, for any $P \in \mathfrak{L}^{regr}$.

We have

Theorem 1 (Reiter's Regression Theorem). *For any regressible P, $R(P) \in \mathfrak{L}_{\mathsf{S}_0}$, and*

$$\mathcal{D} \vdash P$$
$$\textit{iff} \quad \mathcal{D}_{S_0} + \mathcal{D}_{una} \vdash R(P) \tag{16}$$

We give our own proof of this: we will need lemmas from this proof later on, and it seems sensible to integrate them with a proof of the regression theorem. First a series of lemmas:

Lemma 1. *If $\gamma \in \mathfrak{L}^{regr}$ for all $\gamma \in \Gamma$, and if $P \in \mathfrak{L}^{regr}$, then, if*

$$\Gamma \vdash P \tag{17}$$

we have

$$R(\Gamma) \vdash R(P) \tag{18}$$

Proof. This follows from the inductive clauses (13)–(15).

Lemma 2. *For P in $\mathfrak{L}^{regr}$,*

$$\mathcal{D} \vdash P \tag{19}$$

iff

$$\mathcal{D}_{una}, \mathcal{D}_{ap}, \mathcal{D}_{ss}, \mathcal{D}_{\mathsf{S}_0} \vdash P \tag{20}$$

Proof. That (20) entails (19) is trivial, since $\mathcal{D}_{una} \cup \mathcal{D}_{ap} \cup \mathcal{D}_{ss} \cup \mathcal{D}_{\mathsf{S}_0} \subseteq \mathcal{D}$. Now suppose that

$$\mathcal{D} \nvdash P :$$

then there is a model M of $\mathcal{D}_{una} \cup \mathcal{D}_{ap} \cup \mathcal{D}_{ss} \cup \mathcal{D}_{\mathsf{S}_0}$ in which P is false. Now models of $\mathcal{D}_{una} \cup \mathcal{D}_{ap} \cup \mathcal{D}_{ss} \cup \mathcal{D}_{\mathsf{S}_0}$ look like models of $\mathcal{D}$ except that there will be, in general, extra situations beyond the accessible tree: but we can produce a model M' of $\mathcal{D}$ from M simply by restricting the set of situations to be the accessible ones. M' will continue to satisfy the axioms in $\mathcal{D}_{una} \cup \mathcal{D}_{ap} \cup \mathcal{D}_{ss} \cup \mathcal{D}_{\mathsf{S}_0}$ because none of these axioms quantifies over situations. Similarly, P will be false in M' because it is in $\mathfrak{L}^{regr}$ and it only involves situations in the accessible tree. M' is thus a countermodel to (19), and we have

$$\mathcal{D}_{una}, \mathcal{D}_{ap}, \mathcal{D}_{ss}, \mathcal{D}_{\mathsf{S}_0} \nvdash P.$$

Remark 6. In the analogy with arithmetic which Reiter develops in [16, pp. 48ff], we can regard $\mathcal{D}_{una} \cup \mathcal{D}_{ap} \cup \mathcal{D}_{ss} \cup \mathcal{D}_{\mathsf{S}_0} \subseteq \mathcal{D}$ as corresponding to an axiomatisation of primitive recursive arithmetic. The elements of $\mathfrak{L}^{regr}$ will correspond to the primitive recursive predicates, and Lemma 2 will correspond to a well known conservativity theorem.

Lemma 3. *If*

$$\mathcal{D}_{una}, \mathcal{D}_{ap}, \mathcal{D}_{ss}, \mathcal{D}_{\mathsf{S}_0} \vdash P, \tag{21}$$

for $P \in \mathfrak{L}^{regr}$, then there is a collection Γ of instantiations at ground situations of elements of $\mathcal{D}_{ap}, \mathcal{D}_{ss}, \mathcal{D}_{\mathsf{S}_0}$ such that

$$\mathcal{D}_{una}, \Gamma \vdash P$$

Proof. An induction on the structure of a cut free proof of (21).

Proof of Reiter's Regression Theorem. Suppose that we have $\mathcal{D} \vdash P$, for $P \in \mathfrak{L}^{regr}$. By Lemmas 2 and 3, there is a set Γ of instantiations of elements of $\mathcal{D}_{ap}, \mathcal{D}_{ss}, \mathcal{D}_{\mathsf{S}_0}$ with

$$\mathcal{D}_{una}, \Gamma \vdash P.$$

So, by Lemma 1, we have

$$R(\mathcal{D}_{una}), R(\Gamma) \vdash R(P).$$

Now, the elements of $\mathcal{D}_{una}$ do not involve situations, so R acts trivially on them. Examination of the definition of R shows that $R(\gamma) = \top$ for every instantiation of an element of $\mathcal{D}_{ap} \cup \mathcal{D}_{ss} \cup \mathcal{D}_{\mathsf{S}_0}$ at a ground situation: so we have

$$\mathcal{D}_{una} \vdash R(P),$$

which was to be proved.

Remark 7. The equivalence (16) shows that Reiter's regression operator, for a transition α, is a weakest preconditions operator for α. If we have a sequence of actions $\boldsymbol{\alpha}$, and a fluent predicate ψ, then $R(\psi(\mathsf{do}(\alpha, S_0)))$ will – because the only situation it mentions is S_0 – be of the form $\phi(S_0)$. Thus, (16) becomes

$$\begin{aligned} & \mathcal{D} \vdash \psi(\mathsf{do}(\alpha, S_0)) \qquad (22) \\ \text{iff} \quad & \mathcal{D}_{S_0} + \mathcal{D}_{\text{una}} \vdash \phi(S_0) \end{aligned}$$

which is exactly the weakest preconditions condition.

1.2.2 Analysis

In order to compare Reiter's system with ours, we need a rather more abstract view of what he is doing. The core of the definition is given by the correspondence (22) between predicates evaluated at $\mathsf{do}(\alpha, S_0)$ and predicates evaluated at S_0: we can regard it as simply a mapping between predicates, which – in the case considered in (22) – takes the predicate ψ into the predicate $\phi = R(\psi)$. (22) trivially entails that

$$R(\top) = \top \qquad (23)$$

$$R(\psi_1 \wedge \psi_2) = R(\psi_1) \wedge R(\psi_2) \qquad (24)$$

However, other properties (in particular, whether R commutes with $\vee$ or preserves $\bot$) are not trivially entailed by eqrefeq:reiterCore, and there are circumstances in which one may well not have this: for example, if one has actions with "disjunctive effects" – that is, if one uses successor state axioms to specify, not the effect of actions on propositional atoms, but their effect on disjunctions – then we find that we can define a regression operator, and prove the analogue of the regression theorem for it, but that this operator will preserve neither $\vee$ nor $\bot$. The fact that our regression operator can be consistently defined to preserve $\vee$ and $\bot$ is quite closely tied to the syntactic form of the effect axioms, and, specifically, the fact that the successor state axioms are defined for propositional atoms.

We can, then, differentiate between properties of R which only depend on rather formal features of its definition – such as the fact that it preserves $\top$ and

commutes with $\wedge$ – an others – preserving $\bot$ and commuting with $\vee$ – which seem to be less fundamental and more dependent on particular details of the system. This turns out to be the case: in particular, we will be able to develop, as well as a system very strictly analogous to Reiter's, one in which the weakest precondition operators only respect $\wedge$ and $\top$. Furthermore, if we have classical logic, and if the regression operator respects $\wedge$, $\vee$, $\bot$, and $\top$, then it must respect $\neg$ as well, since any lattice homomorphism between boolean algebras is, in fact, a boolean algebra homomorphism [10, p. 4].

1.3 The Progression Operator

1.3.1 Reiter's Definition

Reiter defines progression [16, pp. 207ff.] in the following terms.

Definition 8. Suppose we have a basic action theory $\mathcal{D} = \mathbf{\Sigma} \cup \mathcal{D}_{una} \cup \mathcal{D}_{ap} \cup \mathcal{D}_{ss} \cup \mathcal{D}_{\mathsf{S}_0}$. Let S_α be a situation: then a set of sentences, $\mathcal{D}_{S_\alpha}$, is a *progression* of $\mathcal{D}_{\mathsf{S}_0}$ if

1. $\mathcal{D}_{S_\alpha} \subseteq \mathfrak{L}_{S_\alpha}$
2. Let $\mathcal{D}_\alpha = \mathbf{\Sigma} \cup \mathcal{D}_{una} \cup \mathcal{D}_{ap} \cup \mathcal{D}_{ss} \cup \mathcal{D}_{S_\alpha}$: then $\mathcal{D} \models \mathcal{D}_\alpha$
3. For every model M_α of $\mathcal{D}_\alpha$, there is a model of $\mathcal{D}$ such that
   (a) M and M_α have the same domains
   (b) M and M_α interpret $\underline{\mathfrak{L}}$ identically
   (c) M and M_α interpret all fluents about S_α and its future identically
   (d) M and M_α interprets Poss identically in S_α and its future.

Such a progression operator is second-order definable, and Reiter also gives two cases in which one can define it in first-order terms [16, pp. 210ff.].

1.3.2 Analysis

First, a lemma:

Lemma 4. *For $P \in \mathfrak{L}_{\mathsf{S}_0}$,*

$$\mathcal{D} \vdash P$$

iff

$$\mathcal{D}_{\mathsf{S}_0}, \mathcal{D}_{una} \vdash P$$

Proof. It is easy to check, by examining the definition of the regression operator, that, for $P \in \mathfrak{L}_{\mathsf{S}_0}$, $R(P) = P$. We then apply the regression theorem (Theorem 1), and obtain the equivalence.

Now suppose that we have a basic action theory for which we can define progression for arbitrary finite theories: by restricting to theories consisting of one proposition, we get a map (where S_α is a ground situation)

$$\eta_{\mathsf{S}_\alpha} : \mathfrak{L}_{\mathsf{S}_0} \to \mathfrak{L}_{\mathsf{S}_\alpha},$$

while the restriction of the regression operator gives a map

$$R_{\mathsf{S}_\alpha} : \mathfrak{L}_{\mathsf{S}_\alpha} \to \mathfrak{L}_{\mathsf{S}_0}$$

We have the following:

Proposition 1. *For suitable entailment relations on $\mathfrak{L}_{\mathsf{S}_0}$ and $\mathfrak{L}_{\mathsf{S}_\alpha}$, η_{S_α} is left adjoint to R_{S_α}: to be precise, for $P \in \mathfrak{L}_{\mathsf{S}_0}$ and $Q \in \mathfrak{L}_{\mathsf{S}_\alpha}$, we have*

$$\mathcal{D}_{una}, P \vdash R_{\mathsf{S}_\alpha}(Q)$$

iff

$$\mathcal{D}_{una}, \eta_{\mathsf{S}_\alpha}(P) \vdash Q.$$

Proof. We proceed semantically, and then use soundness and completeness to derive the syntactic entailments. We will use Reiter's terminology for a basic action theorem with $\mathcal{D}_{\mathsf{S}_0} = \{P\}$: thus, we will have throughout

$$\mathcal{D} = \mathbf{\Sigma} \cup \mathcal{D}_{una} \cup \mathcal{D}_{ap} \cup \mathcal{D}_{ss} \cup \{P\}$$

and

$$\mathcal{D}_\alpha = \mathbf{\Sigma} \cup \mathcal{D}_{una} \cup \mathcal{D}_{ap} \cup \mathcal{D}_{ss} \cup \{\eta_{\mathsf{S}_\alpha}(P)\}$$

So, by the regression theorem,

$$\begin{array}{ll} & \mathcal{D}_{una}, P \models R_{\mathsf{S}_\alpha}(Q) \\ \text{iff} & \mathcal{D} \models Q \\ \text{iff} & \text{every model of } \mathcal{D} \text{ models } Q \end{array}$$

iff (by the definition of progression)

$$\begin{array}{ll} & \text{every model of } \mathcal{D}_\alpha \text{ models } Q \\ \text{iff} & \mathcal{D}_\alpha \models Q \end{array}$$

iff (by the lemma)

$$\mathcal{D}_{una}, \eta_{\mathsf{S}_\alpha}(P) \models Q$$

2 The Systems

We want to start with a fixed deterministic labelled transition system (or, more generally, from a category derived therefrom) and construct a logic, together with operators α^*, which will allow us to solve the precondition and regression problems. We will also want extra machinery to solve the progression problem: that is, we will want α^* to have a left adjoint, which we will write $\alpha_!$.

Our system will be typed: a proposition will be *about* a state (or set of states), and we will express this by making the states (or sets of states) into types. More formally, we start with a category $\mathcal{C}$, together with a functor into the free monoid over the set of labels, and let our types be the objects of this category. Propositions will be generated by our language $\mathfrak{L}$ together with the operators α^* and $\alpha_!$. The types, and syntax, are summarised in Table 1.

2.1 The Intuitionistic System

At this stage we need to make some decisions about the syntax and semantics of our logic. Given only the operators α^*, we can easily write down entailments of the form $P \vdash \alpha^* Q$ between single propositions: here, if $s \overset{\alpha}{\frown\!\!\rightarrow} t$, P would be a proposition at s and Q would be a proposition at t. We would here have something very like Hoare triples.

However, in order to have something which is more like a logical system, we need to have some notion of contexts. An initial idea would be to consider entailments such as

$$P_1, \ldots, P_n \vdash \alpha^* Q, \tag{25}$$

where (with the above assumptions) $P_1, \ldots, P_n$ are all propositions at s and Q is a proposition at t. But this will not do: entailments of the form (25) are not closed under cut, and there is no obvious way to give rules for $\rightarrow$ or for $\neg$. So, if we are to have non-trivial contexts on the left – or, indeed, on the right – then they must be capable of incorporating propositions located at different states.

A convenient way of doing this is as follows. We will consider, at first, left contexts only. Left contexts are normally generated by a context constructor, which is syntactically ‘ , ’, and which has the semantics of $\wedge$. We add another context constructor $\{\cdot\}_{\alpha:s\to t}$, which will have (for a given α) the semantics of $\alpha_!$. An entailment of the form (25) can be written, using these contexts, as

$$\{P_1, \ldots, P_n\}_{\alpha:s\to t} \vdash Q, \tag{26}$$

and the adjunction between α^* and $\alpha_!$ means that (25) and (26) are equivalent.

Remark 8. Suppose that we have $\alpha : s \to t$, and that Γ is a flat context at s (i.e. that it is constructed using only , and not $\{\cdot\}_\cdot$). We then have

$$\frac{\Gamma : s \vdash \alpha^*(Q : t)}{\{\Gamma\}_\alpha : t \vdash Q : t}$$

Note that, in the sequent on the top, both sides are of the same type. Our cut elimination theorem will show that α^* is exactly a weakest preconditions operator. This is, in fact, the content of Reiter's "Regression Theorem" [16, Thm. 4.55, p 65], or our Theorem 1.

But, as well as entailments such as (26), we can also have entailments with nested occurrences of the $\{\cdot\}_.$ operator, or entailments where different $\{\cdot\}_.$ operators (or none) apply to different parts of the context. In this way, we can have a cut rule, and we can have rules for $\rightarrow$ and $\neg$. If we have left contexts of this form, our entailments will have single propositions on the right, and full contexts on the left. This will, in the usual way, mean that our logic is intuitionistic.

Remark 9. We are using, here, a context constructor $\{\cdot\}_\alpha$, which has the semantics of $\alpha_!$. We could ask whether we could, instead, use an operator with the semantics of α^*: but, in this case, we could not. The validity of the left rule for $\vee$ will, when the principal formula is nested at some depth, require that the context constructor commutes with $\vee$. In the intuitionistic case, the weakest preconditions operator, α^*, is right adjoint to $\alpha_!$: consequently, α^* commutes with $\wedge$, but not necessarily with $\vee$, whereas $\alpha_!$ commutes with $\vee$. We could argue in exactly the same way about the requirements for the validity of the left rule for $\neg$.

2.2 The Classical System

We can also define a system with classical semantics, although we need a good deal more machinery.

There are several issues to deal with. The first is that, in the standard sequent calculus formulations of classical logic, we have non-trivial contexts both on the left and on the right. In standard classical logic, the only context constructor, on the left or on the right, is the comma, but it has different semantics on each side: $\wedge$ on the left and $\vee$ on the right. In order for the classical $\neg$ rule should be valid, the two interpretations of the comma must be de Morgan dual to each other.

So, if we want to have full contexts on the right, we need to have suitable context constructors on both sides, and these context constructors need to be de Morgan dual to each other. The comma is fine: we just have it represent $\wedge$ on the left, and $\vee$ on the right. However, we also need another constructor, one which moves between situations. Our analysis of Reiter's system shows that, in the classical case, we can reasonably expect α^* to be a Boolean algebra homomorphism, and thus to have both a right and a left adjoint. So, there are two choices for the semantics of our context constructor:

1. the semantics of α^* on both sides. Notice that α^*, being a Boolean algebra homomorphism, is de Morgan self dual, and commutes with $\wedge$ and $\vee$.

2. the semantics of $\alpha_!$ on the left and the semantics of α_* on the right. Notice here that $\alpha_!$ commutes with $\vee$, α_* commutes with $\wedge$, and (because α^* is a Boolean algebra homomorphism) $\alpha_!$ is de Morgan dual to α_*.

The first choice gives us a system with good properties. The second choice, although it yields rules for $\wedge$ and $\vee$, still suffers from problems. $\alpha_!$ does not commute with $\wedge$, and α_* does not commute with $\vee$: and this turns out to mean that the second choice does not give us a valid axiom rule for entailments like

$$\{P\}_\alpha \vdash \{P\}_\alpha \,,$$

or a valid cut rule for premises like

$$\Gamma \vdash \{\Delta, P\}_\alpha \quad \{P, \Gamma'\}_\alpha \vdash \Delta'.$$

The second issue is this: we need to give a right rule for $\alpha_!$ (and, dually, a left rule for α_*). Now the *intuitionistic* right rule for $\alpha_!$ (fully annotated with the types for propositions and contexts) is

$$\frac{\Gamma : s \vdash Q : s}{\{\Gamma\}_{\alpha:s\to t} : t \vdash (\alpha_! Q) : t}\, \alpha_! \mathrm{R} \tag{27}$$

and we should notice that this relates entailments at two *different* types: the antecedent is at s and the consequent at t. We can do this in the intuitionistic system, because there is only one proposition on the right, and we can transport the left context to type t simply by enclosing it in $\{\cdot\}_\alpha$.

However, with the classical calculus, we can have nested contexts on the right, and we may want to apply the classical right rule for $\alpha_!$ – however we might choose to formulate it – at any depth of nesting. In such a case, the type of a proposition is tied to the type of the hole in the context into which it fits: we must thus have a way of making $\alpha_! P$ fit into the hole in which P went. We should, then, have a means of transforming contexts so that, if we have a morphism $\alpha : s \to t$, and if $\Gamma[]$ has a hole of type s, we can form a context – which we can write $\langle \Gamma[] \rangle^\alpha$ – which will have a hole of type t. This can be done – this is what the operator defined in Definition 13 – but it will require our base category $\mathcal{C}$ to have pullbacks.

Example 1. There are two significant examples in which we can define such pullbacks. One of them is Reiter's case, when the base category is a tree: here pullbacks are trivial, but they certainly exist. The other case is where the base category has one object, and the endomorphisms of that object are all invertible and hence form a group: this is a very useful object for reasoning about symmetries. Here pullbacks are given by conjugation in the group.

2.3 The Calculi

We can now present our two systems, intuitionistic and classical. The syntax for propositions and contexts is given by Table 1; note that we have explicit notation – ε – for the empty context. There are also rules for equivalence of contexts, and these are given in Table 2: we indicate the rules that only apply classically or intuitionistically. The rules for equivalence of contexts themselves rely on the rules for the equality of morphisms in a category, which we have written down in Table 3 (for these, see, for example, [20, Chapter 8]).

Remark 10. We will, almost exclusively, write $\{\Gamma\}_\alpha \vdash P$ instead of the more pedantic $\{\Gamma\}_{\alpha:s\to t} \vdash P$: in most cases, the source and target of α can be inferred from the context. There is some scope for ambiguity here, however: we might have two actions of type α terminating in a state t, in which case the type of $\alpha^*\Gamma$ ought to be disambiguated by an explicit typing.

We will also need a notion of a context-with-a-hole, or, as we shall describe it, a *marked context*:

Definition 9. A *marked intuitionistic context* is given by the clauses:

$$\begin{aligned}\Gamma[] : t \;&:=\; [: t] \\ &\quad \mid \Gamma[] : t, \Gamma : t \;\mid \Gamma : t, \Gamma[] : t \\ &\quad \mid \{\Gamma[] : s\}_\alpha \quad (\alpha : s \to t),\end{aligned}$$

and a *marked classical context* by the clauses:

$$\begin{aligned}\Gamma[] : t \;&:=\; [: t] \\ &\quad \mid \Gamma[] : t, \Gamma : t \;\mid \Gamma : t, \Gamma[] : t \\ &\quad \mid \{\Gamma[] : u\}^\beta \quad (\beta : t \to u)\end{aligned}$$

where Γ stands for a context (of the apppropriate sort) as defined in Table 1. (Type symbols are optional, and will frequently be omitted when no confusion arises).

The point of marked contexts is that we should be able to substitute into them, so next we define an appropriate notion of substitution:

Definition 10. If $\Gamma[: t]$ is a marked classical or intuitionistic context, and if $\Gamma' : t$ is a context (either classical or intuitionistic, respectively, and either marked or unmarked), then we define $\Gamma[\Gamma']$, the *subsitution* of Γ' in $\Gamma[]$ by the clauses

$$\begin{aligned}[\Gamma'] \;&:=\; \Gamma' \\ \Gamma[\Gamma'], \Gamma'' \;&:=\; (\Gamma[\Gamma'']), \Gamma'' \\ \Gamma'', \Gamma[\Gamma'] \;&:=\; \Gamma'', (\Gamma[\Gamma']) \\ \{\Gamma[\Gamma']\}_\alpha \;&:=\; \{(\Gamma[\Gamma'])\}_\alpha \qquad \text{in the intuitionistic case} \\ \{\Gamma[\Gamma']\}^\alpha \;&:=\; \{(\Gamma[\Gamma'])\}^\alpha \qquad \text{in the classical case}\end{aligned}$$

The result is a marked or unmarked context, depending on what Γ' is.

Lemma 5. *The operation of substitution is a congruence for equivalence of contexts.*

Proof. Elementary.

Remark 11. Notice that $\Gamma[]$ denotes, in all cases, a *marked* context: since we have an explicit notation for the empty context, we can distinguish between $\Gamma[]$ and $\Gamma[\varepsilon]$.

Definition 11. The *nesting*, $\boldsymbol{\nu}$, of either sort of context is a morphism of $\mathcal{C}$ defined as follows:

$$\begin{aligned}\boldsymbol{\nu}([:t]) &= \mathrm{Id}_t\\ \boldsymbol{\nu}(\Gamma[],\Gamma') &= \boldsymbol{\nu}(\Gamma[])\\ \boldsymbol{\nu}(\Gamma,\Gamma'[]) &= \boldsymbol{\nu}(\Gamma'[])\\ \boldsymbol{\nu}(\{\Gamma[]\}_\alpha) &= \alpha\circ\boldsymbol{\nu}(\Gamma[])\\ \boldsymbol{\nu}(\{\Gamma[]\}^\alpha) &= \boldsymbol{\nu}(\Gamma[])\circ\alpha\end{aligned}$$

Two nestings will be equal if they are equal *as morphisms of* $\mathcal{C}$.

Lemma 6. *The nesting of a marked context is invariant under equivalence of contexts.*

Proof. Elementary.

Notice that the nesting of an intuitionistic context behaves covariantly, whereas the nesting of a classical context behaves contravariantly: and, in fact, we have

Proposition 2. *If* $\Gamma[:s]:t$ *is a marked intuitionistic context, then*

$$\boldsymbol{\nu}(\Gamma[:s]:t)\ :s\to t,$$

whereas, if $\Gamma[:s]:t$ *is a marked classical context,*

$$\boldsymbol{\nu}(\Gamma[:s]:t)\ :t\to s.$$

Proof. Obvious.

We will need the following two notions to define our classical system.

Definition 12. Two marked contexts, $\Gamma[]$ and $\Gamma'[]$, will be *similarly nested* if their nestings are equal.

Definition 13. If $(\Gamma[:t]):u$ is a marked classical context, and if $\mathcal{C}$ has pullbacks, and if $\alpha:s\to t$ is a morphism, then we define the marked context $\langle\Gamma[:t]\rangle^\alpha$, which has type $s\times_t u$, by the following inductive clauses:

$$\begin{aligned}\langle[:t]\rangle^\alpha &= [:s]\\ \langle\Gamma,\Gamma'[:t]:u\rangle^\alpha &= \{\Gamma\}^{\alpha\times_t u},\langle\Gamma'[:t]:u\rangle^\alpha\\ \langle\Gamma[:t],\Gamma':u\rangle^\alpha &= \langle\Gamma[:t]\rangle^\alpha,\{\Gamma'\}^{\alpha\times_t u}\\ \left\langle\{\Gamma[:t]:u\}^{\beta:u'\to u}\right\rangle^\alpha &= \{\langle\Gamma[:t]\rangle^\alpha:s\times_t u\}^{(s\times_t u)\times_u\beta}:s\times_t u'\end{aligned}$$

Lemma 7. *If* $\Gamma[:t]:u$ *is a marked classical context, and if* $\alpha:s\to t$*, then*

$$\boldsymbol{\nu}(\langle\Gamma[]\rangle^\alpha)\ =\ \alpha\times_t\boldsymbol{\nu}(\Gamma[]).$$

Proof. Induction.

Lemma 8. *The operation $\langle\cdot\rangle^{\cdot}$ is a congruence with respect to equivalence of classical contexts.*

Proof. The obvious induction.

The following is merely a restatement of the last clause of Definition 13, but it will be convenient to be able to cite it.

Lemma 9. *If $\Gamma[:t]$ is a marked classical context, and if $\alpha : s \to t$, then*

$$\left\langle \{\Gamma[:t] : s\}^{\beta : u' \to u} \right\rangle^{\alpha} = \{\langle \Gamma[:t] \rangle^{\alpha} : s \times_t u\}^{(s \times_t u) \times_u \beta} : s \times_t u'$$

Remark 12. We should point out that $\langle\cdot\rangle^{\cdot}$ is *not* a context constructor: the only context constructors are the comma and $\{\cdot\}^{\cdot}$. $\langle\cdot\rangle^{\cdot}$ is an *operation* on contexts, defined by the above clauses: it takes a context, constructed with the usual constructs, and it yields another context, constructed with those same constructors. Correspondingly, Lemma 9 says that two context are *equal*, rather than equivalent, and equality here means typographical equality.

Remark 13. Notation such as $\langle\Gamma[\Gamma']\rangle^{\alpha}$ might seem to be ambiguous: is the $\langle\cdot\rangle^{\alpha}$ performed before, or after, the substitution into the marked context? However, it is not: $\langle\cdot\rangle^{\alpha}$ can only be performed on a marked context, so that the substitution must be performed *after* $\langle\cdot\rangle^{\alpha}$.

So, given these preliminaries, the intuitionistic system is given by the rules in Table 4, and the classical system by the rules in Table 5.

Remark 14. From either set of rules, it follows that

$$\frac{P : s \vdash \alpha^*(Q : t)}{\alpha_!(P : s) \vdash Q : t} \tag{28}$$

and so $\alpha_!$ is, as promised, a left adjoint to α^*. From the classical rules, we can derive

$$\frac{\alpha^* P : t \vdash Q : s}{P : s \vdash \alpha_* Q : s,} \tag{29}$$

and so, in the classical calculus, α_* is a right adjoint to α^*.

3 Semantics: The Intuitionistic System

The goal of this and the following sections will to prove soundness and completeness for both systems, relative to their intended semantics. We start with the intuitionistic system.

The semantics of this logic should be a category, fibred over a labelled transition system (regarded as a category together with a functor into the free

propositions	$P : t \;\; (P \in \mathfrak{L}, t \in \mathrm{Ob}(\mathcal{C}))$
	$\dfrac{P : t \quad \alpha : s \to t}{\alpha^* P : s}$
	$\dfrac{P : s \quad \alpha : s \to t}{\alpha_! P : t}$
(classical case only)	$\dfrac{P : s \quad \alpha : s \to t}{\alpha_* P : t}$
intuitionistic contexts	$\Gamma_i : t \;:=\; P : t \mid \Gamma_i : t, \Gamma_i : t$ $\mid \{\Gamma_i : s\}_\alpha \quad (\alpha : s \to t)$
classical contexts	$\Gamma_c : t \;:=\; P : t \mid \Gamma_c : t, \Gamma_c : t$ $\mid \{\Gamma_c : u\}^\alpha \quad (\alpha : t \to u)$
intuitionistic entailments	$\dfrac{\Gamma_i : t \quad P : t}{\Gamma_i \vdash P \quad \text{well-formed}}$
classical entailments	$\dfrac{\Gamma_c : t \quad \Delta_c : t}{\Gamma_c \vdash \Delta_c \quad \text{well-formed}}$

Table 1: Types and Syntax

common	$\dfrac{\Gamma \approx \Gamma' \quad \Gamma' \approx \Gamma''}{\Gamma \approx \Gamma''}$	$\dfrac{}{\Gamma \approx \Gamma}$
	$\dfrac{}{\Gamma, \Gamma' \approx \Gamma', \Gamma}$	$\dfrac{\Gamma \approx \Gamma' \quad \Gamma_1 \approx \Gamma_1'}{\Gamma, \Gamma_1 \approx \Gamma', \Gamma_1'}$
	$\dfrac{}{\Gamma, \varepsilon \approx \Gamma}$	
intuitionistic only	$\dfrac{}{\{\Gamma\}_{\mathrm{Id}} \approx \Gamma}$	$\dfrac{}{\{\Gamma\}_{\beta\circ\alpha} \approx \{\{\Gamma\}_{\alpha}\}_{\beta}}$
	$\dfrac{\alpha = \beta}{\{\Gamma\}_{\alpha} \approx \{\Gamma\}_{\beta}}$	$\dfrac{\Gamma : s \approx \Gamma' : s \quad \alpha : s \to t}{\{\Gamma\}_{\alpha} \approx \{\Gamma'\}_{\alpha}}$
classical only	$\dfrac{}{\{\varepsilon\}^{\alpha} \approx \varepsilon}$	$\dfrac{}{\{\Gamma\}^{\mathrm{Id}} \approx \Gamma}$
	$\dfrac{}{\{\Gamma\}^{\alpha\circ\beta} \approx \left\{\{\Gamma\}^{\beta}\right\}^{\alpha}}$	$\dfrac{\alpha = \beta}{\{\Gamma\}^{\alpha} \approx \{\Gamma\}^{\beta}}$
	$\dfrac{\Gamma : t \approx \Gamma'' : t \quad \alpha : s \to t}{\{\Gamma\}^{\alpha} \approx \{\Gamma'\}^{\alpha}}$	

Table 2: Rules for Equivalence of Contexts

$$\frac{}{\mathrm{Id} \circ \alpha = \alpha} \qquad \frac{}{\alpha = \alpha \circ \mathrm{Id}}$$

$$\frac{}{(\alpha \circ \beta) \circ \gamma = \alpha \circ (\beta \circ \gamma)}$$

$$\frac{\alpha = \beta}{\gamma \circ \alpha = \gamma \circ \beta} \qquad \frac{\alpha = \beta}{\alpha \circ \gamma = \beta \circ \gamma}$$

Table 3: Rules for Category Theory

$$\frac{\Gamma \vdash Q \quad \Gamma \approx \Gamma'}{\Gamma' \vdash Q} \approx \qquad \frac{\Gamma[\{\Gamma'\}_\alpha, \{\Gamma''\}_\alpha] \vdash P}{\Gamma[\{\Gamma', \Gamma''\}_\alpha] \vdash P} \text{join}$$

$$\frac{}{\Gamma, P \vdash P} \text{Axiom} \qquad \frac{}{\Gamma[\bot] \vdash Q} \bot \text{L}$$

$$\frac{\Gamma[\varepsilon] \vdash Q}{\Gamma[\Gamma'] \vdash Q} \text{LW} \qquad \frac{\Gamma \vdash}{\Gamma \vdash Q} \text{RW}$$

$$\frac{\Gamma[P, P] \vdash Q}{\Gamma[P] \vdash Q} \text{LC}$$

$$\frac{\Gamma[P_1] \vdash Q \quad \Gamma[P_2] \vdash Q}{\Gamma[P_1 \vee P_2] \vdash Q} \vee\text{L} \qquad \frac{\Gamma \vdash Q_i}{\Gamma \vdash Q_1 \vee Q_2} \vee\text{R}_i$$

$$\frac{\Gamma[P_1, P_2] \vdash Q}{\Gamma[P_1 \wedge P_2] \vdash Q} \wedge\text{L} \qquad \frac{\Gamma \vdash Q_1 \quad \Gamma \vdash Q_2}{\Gamma \vdash Q_1 \wedge Q_2} \wedge\text{R}$$

$$\frac{\Gamma', P \to Q \vdash P : s \quad \Gamma[\Gamma', Q : s] \vdash R}{\Gamma[\Gamma', P \to Q : s] \vdash R} \to\text{L} \qquad \frac{\Gamma, P \vdash Q}{\Gamma \vdash P \to Q} \to\text{R}$$

$$\frac{\Gamma[P : t] \vdash Q \quad \alpha : s \to t}{\Gamma[\{\alpha^* P\}_\alpha] \vdash Q} \alpha^*\text{L} \qquad \frac{\{\Gamma\}_\alpha \vdash Q}{\Gamma \vdash \alpha^* Q} \alpha^*\text{R}$$

$$\frac{\Gamma[\{P\}_\alpha] \vdash Q}{\Gamma[\alpha_! P] \vdash Q} \alpha_!\text{L} \qquad \frac{\Gamma \vdash Q}{\{\Gamma\}_\alpha \vdash \alpha_! Q} \alpha_!\text{R}$$

$$\frac{\Gamma \vdash P : s \quad \Gamma'[P : s] \vdash Q}{\Gamma'[\Gamma] \vdash Q} \text{cut}$$

Table 4: The Intuitionistic Sequent Calculus

$$\frac{\Gamma \vdash \Delta \quad \Gamma \approx \Gamma' \quad \Delta \approx \Delta'}{\Gamma' \vdash \Delta'} \approx$$

$$\frac{\Gamma[\{\Gamma'\}^\alpha, \{\Gamma''\}^\alpha] \vdash \Delta}{\Gamma[\{\Gamma', \Gamma''\}^\alpha] \vdash \Delta} \text{joinL} \qquad \frac{\Gamma \vdash \Delta[\{\Delta'\}^\alpha, \{\Delta'\}^\alpha]}{\Gamma \vdash \Delta[\{\Delta', \Delta''\}^\alpha]} \text{joinR}$$

$$\frac{\Gamma[\{\Gamma', \Gamma''\}^\alpha] \vdash \Delta}{\Gamma[\{\Gamma'\}^\alpha, \{\Gamma''\}^\alpha] \vdash \Delta} \text{splitL} \qquad \frac{\Gamma \vdash \Delta[\{\Delta', \Delta''\}^\alpha]}{\Gamma \vdash \Delta[\{\Delta'\}^\alpha, \{\Delta'\}^\alpha]} \text{splitR}$$

$$\frac{}{\Gamma[P] \vdash \Delta[P]} \text{Axiom} \qquad \frac{\Gamma : t \vdash \Delta : t}{\{\Gamma\}^{\alpha:s\to t} \vdash \{\Delta\}^{\alpha:s\to t}} \{\}^\alpha$$

$$\frac{}{\Gamma[\bot] \vdash \Delta} \bot\,\text{L} \qquad \frac{}{\Gamma \vdash \Delta[\top]} \top\text{R}$$

$$\frac{\Gamma[\varepsilon] \vdash \Delta}{\Gamma[\Gamma'] \vdash \Delta} \text{LW} \qquad \frac{\Gamma \vdash \Delta[\varepsilon]}{\Gamma \vdash \Delta[\Delta']} \text{RW}$$

$$\frac{\Gamma[P, P] \vdash \Delta}{\Gamma[P] \vdash \Delta} \text{LC} \qquad \frac{\Gamma \vdash \Delta[P, P]}{\Gamma \vdash \Delta[P]} \text{RC}$$

$$\frac{\Gamma[P_1] \vdash Q \quad \Gamma[P_2] \vdash Q}{\Gamma[P_1 \vee P_2] \vdash Q} \vee\text{L} \qquad \frac{\Gamma \vdash \Delta[Q1, Q2]}{\Gamma \vdash \Delta[Q_1 \vee Q_2]} \vee\text{R}$$

$$\frac{\Gamma[P_1, P_2] \vdash \Delta}{\Gamma[P_1 \wedge P_2] \vdash \Delta} \wedge\text{L} \qquad \frac{\Gamma \vdash \Delta[Q_1] \quad \Gamma \vdash \Delta[Q_2]}{\Gamma \vdash \Delta[Q_1 \wedge Q_2]} \wedge\text{R}$$

$$\frac{\Gamma[\varepsilon] \vdash \Delta[P]}{\Gamma[\neg P] \vdash \Delta[\varepsilon]} \neg\text{L} \qquad \frac{\Gamma[Q] \vdash \Delta[\varepsilon]}{\Gamma[\varepsilon] \vdash \Delta[\neg Q]} \neg\text{R}$$

$$\frac{\Gamma[\{P\}^\alpha] \vdash \Delta}{\Gamma[\alpha^* P] \vdash \Delta} \alpha^*\text{L} \qquad \frac{\Gamma \vdash \Delta[\{Q\}^\alpha]}{\Gamma \vdash \Delta[\alpha^* Q]} \alpha^*\text{R}$$

$$\frac{\langle \Gamma[P : s] \rangle^{\alpha:s\to t} \vdash \{\Delta\}^{\alpha\times_t u}}{\Gamma[(\alpha_! P) : t] : u \vdash \Delta : u} \alpha_!\text{L} \qquad \frac{\Gamma \vdash \Delta[Q]}{\Gamma \vdash \Delta[\{\alpha_! Q\}^\alpha]} \alpha_!\text{R}$$

$$\frac{\Gamma[P] \vdash \Delta}{\Gamma[\{\alpha_* P\}^\alpha] \vdash \Delta} \alpha_*\text{L} \qquad \frac{\{\Gamma\}^{\alpha\times_t u} \vdash \langle \Delta[Q : s] \rangle^{\alpha\times_t u} : s \times_t u}{\Gamma : u \vdash \Delta[(\alpha_* Q) : t] : u} \alpha_*\text{R}$$

$$\frac{\Gamma : s \vdash \Delta[P : t] : s \qquad \Gamma'[P : t] : u \vdash \Delta' : t \qquad \boldsymbol{\nu}(\Delta[]) = \alpha,\ \boldsymbol{\nu}(\Gamma[]) = \beta}{\{\Gamma\}^{s\times_t \beta}, \{\Gamma'[\varepsilon]\}^{\alpha\times_t u} \vdash \{\Delta[\varepsilon]\}^{s\times_t \beta}, \{\Delta'\}^{\alpha\times_t u}} \text{cut}$$

Axiom, ¬L, ¬R: $\Gamma[]$, $\Delta[]$ similarly nested.

Table 5: The Classical Sequent Calculus

monoid over the set of action labels). The fibres should be models of intuitionistic logic (i.e. complete Heyting algebras), and the reindexing functors should be $\wedge$-semilattice homomorphisms: that is, they should preserve finitary and infinitary meets – that is, $\wedge$ and $\bigwedge$ – but not necessarily $\vee$, $\neg$ or $\rightarrow$. These conditions mean that we have left adjoints to reindexing.

Suppose that we have our semantics: that is, a category $\mathcal{E}$ fibred over $\mathcal{C}$, whose fibres are Heyting algebras, and with left adjoints to reindexing. Being Heyting algebras, the fibres are partial orders, and we will write the partial order $\leq$.

Definition 14. An *assignment* is an assignment, to every $t \in \mathrm{Ob}(\mathcal{C})$ and every atomic $P \in \mathfrak{L}$, an element $[\![P]\!]_t \in \mathrm{Ob}(\mathcal{E}_t)$.

Given an assignment, we can define, for each $P : t$, its semantic value $[\![P]\!]_t$ by induction on its syntactic complexity: the intuitionistic operators are interpreted in the usual way, α^* is interpreted as the reindexing functor (also written α^*), and $\alpha_!$ is interpreted as the left adjoint to reindexing.

We also define $[\![\Gamma]\!]_t$ for contexts:

Definition 15. The semantic value of a context is given by the clauses

- $[\![\Gamma]\!]_t = [\![P]\!]_t$ if $\Gamma = P$
- $[\![\Gamma, \Gamma']\!]_t = [\![\Gamma]\!]_t \wedge [\![\Gamma']\!]_t$
- $[\![\{\Gamma\}_\alpha]\!]_t = \alpha_![\![\Gamma]\!]_s$ (for $\alpha : s \rightarrow t$).

And, finally, a definition of semantic entailment:

Definition 16.

$$\Gamma : t \Vdash P : t \tag{30}$$

iff

$$[\![\Gamma]\!]_t \leq [\![P]\!]_t \tag{31}$$

for every assignment

$$. \tag{32}$$

3.1 Soundness

We now prove soundness. The proofs for rules involving the standard connectives will be similar to the standard proofs for intuitionistic logic: the proofs for the rules for α^* and $\alpha_!$ will need some attention to the category theory. We give the proofs of these rather systematically, in preparation for the more complex

classical case. First we formulate the category theoretic facts; here X and Y are appropriate objects of the total category:

$$X \leq \alpha^* Y \quad \text{iff}$$
$$\alpha_! X \leq Y \tag{33}$$
$$X \leq \alpha^* \alpha_! X \tag{34}$$
$$\alpha_! \alpha^* X \leq X \tag{35}$$
$$\alpha_!(X \wedge Y) \leq \alpha_! X \wedge \alpha_! Y \tag{36}$$
$$\alpha_!(X \vee Y) = \alpha_! X \vee \alpha_! Y \tag{37}$$
$$\alpha_! \perp = \perp \tag{38}$$

Here (33) is a statement of the adjunction between $\alpha_!$ and α^*, (34) is its unit, and (35) its counit; (36) follows from the functoriality of $(\cdot)_!$; while (37) and (38) follow from the fact that $(\cdot)_!$ is a left adjoint.

3.1.1 The Structural Rules

We first prove soundness for the structural rules: that is, the relevant rules in Table 2, together with the join rule from Table 4.

Proposition 3. *The intuitionistic rules for the equivalence of contexts in Table 2 are sound with respect to $\Vdash$: that is, if $\Gamma : t \approx \Gamma' : t$, then $[\![\Gamma]\!]_t = [\![\Gamma']\!]_t$ (and so, in particular, we have, for any P, $\Gamma \Vdash P$ iff $\Gamma' \Vdash P$).*

Proof. This follows from the functoriality of $(\cdot)_!$.

Proposition 4. *The join rule is sound for $\Vdash$.*

Proof. This follows from (36).

3.1.2 The Standard Rules

We can now move on to consider the soundness of the "standard" rules – that is, the rules which use only the standard connectives of intuitionistic logic – with these contexts. The right rules are dealt with in the same way as for intuitionistic logic:

Proposition 5. *The right rules for $\wedge$, $\vee$ and $\rightarrow$ are sound for our contexts.*

For the left rules, we need

Lemma 10 (Semantic Monotonicity). *Consider a context $\Gamma[\Gamma' : s] : t$. Then, for any $\Gamma'' : s$, if $[\![\Gamma']\!]_s \leq [\![\Gamma'']\!]_t$, $[\![\Gamma[\Gamma']]\!]_t \leq [\![\Gamma[\Gamma'']]\!]_t$.*

Proof. An induction on the complexity of contexts, using the functoriality of the context constructors.

Proposition 6. *The left rules for $\wedge$ and $\vee$, $\rightarrow$ and $\neg$ are sound for our contexts.*

Proof. $\wedge$ and $\vee$ are straightforward: they rely on the two identities

$$[\![\Gamma[P \wedge Q]]\!]_t = [\![\Gamma[P, Q]]\!]_t \tag{39}$$
$$[\![\Gamma[P \vee Q]]\!]_t = [\![\Gamma[P]]\!]_t \vee [\![\Gamma[Q]]\!]_t. \tag{40}$$

The first follows from the functoriality of the context constructors: the second follows from an induction on the depth of nesting, using (37).

The proof for $\rightarrow$ is again very similar to the standard proof. If we have the premises of the rule, that is, that $[\![\Gamma', P \rightarrow Q]\!]_s \leq [\![P]\!]_s$ and $[\![\Gamma[Q]]\!]_t \leq [\![R]\!]_t$, then we argue as follows:

$$[\![\Gamma', P \rightarrow Q]\!]_s \leq [\![P]\!]_s,$$

and so

$$\begin{aligned} [\![\Gamma', P \rightarrow Q]\!]_s &\leq [\![\Gamma', P \rightarrow Q, P]\!]_s \\ &\leq [\![Q]\!]_s \end{aligned}$$

and so, by semantic monotonicity,

$$\begin{aligned} [\![\Gamma[\Gamma', P \rightarrow Q]]\!]_t &\leq [\![\Gamma[Q]]\!]_t \\ &\leq [\![R]\!]_t \end{aligned}$$

which was to be proved.

Finally $\bot$:

Proposition 7. *The rule $\bot$ L is sound for $\Vdash$.*

Proof. We use (37), together with an induction on the depth of nesting.

3.1.3 α^* and $\alpha_!$

Proposition 8. *The right rule for α^* is sound for $\Vdash$.*

Proof. Suppose that we have the premise of the right rule, namely

$$[\![\{\Gamma\}_\alpha]\!]_t \leq [\![P]\!]_t \quad \text{where } \alpha : s \rightarrow t.$$

Then, by definition of $[\![\cdot]\!].$,

$$\alpha_!([\![\Gamma]\!]_s) \leq [\![P]\!]_t$$

and so, by the adjunction (33)

$$\begin{aligned} [\![\Gamma]\!]_s &\leq \alpha^*([\![P]\!]_t) \\ &= [\![\alpha^* P]\!]_s \end{aligned}$$

which was to be proved.

Proposition 9. *The left rule for α^* is sound for $\Vdash$.*

Proof. Suppose that $\alpha : s \to t$, and that we have the premise for the left rule, namely

$$[\![\Gamma[P : t]]\!]_u \leq [\![Q]\!]_u$$

Using the counit (35) of the adjunction, we have

$$[\![\{\alpha^* P\}_\alpha]\!]_t \leq [\![P]\!]_t$$

By semantic monotonicity, we then have

$$[\![\Gamma[\{P\}_\alpha]]\!]_u \leq [\![Q]\!]_u$$

which is the conclusion.

Proposition 10. *The right rule for $\alpha_!$ is sound for $\Vdash$.*

Proof. This follows from the functoriality of $\alpha_!$.

Proposition 11. *The left rule for $\alpha_!$ is sound for $\Vdash$.*

Proof. This follows from the definition of the semantic value of $\alpha_!$.

3.1.4 The Cut Rule

Proposition 12. *The cut rule is sound for $\Vdash$.*

Proof. This follows trivially from semantic monotonicity.

3.2 Completeness

We can now prove completeness for the intuitionistic logic.

Theorem 2. *Suppose that we have*

$$\Gamma : t \Vdash P : t$$

Then

$$\Gamma : t \vdash P : t$$

Proof. We prove this by contraposition: we suppose that $\Gamma : t \nvdash P : t$, and we concoct a model, and an assignment, which provides a counterexample to $\Gamma : t \Vdash P : t$. The model will be what is, in categorical semantics, called the *generic model* [8, p. 249].

Let us first define a category $\mathcal{E}$ as follows:

objects will be equivalence classes of typed propositions $P : t$, under the equivalence relation

$$P : t \cong Q : t \text{ iff } P : t \vdash Q : t \text{ and } Q : t \vdash P : t$$

morphisms There is a morphism from $P : s$ to $Q : t$, over $\alpha : s \to t$, iff

$$\{P : s\}_\alpha \vdash Q : t$$

for some $\alpha : s \to t$.

identity of morphisms Two morphisms, with the same source and target, are identical iff they lie over the same morphism in the base.

Then we establish the following:

1. There is a functor $\pi : \mathcal{E} \to \mathcal{C}$, which takes a morphism in $\mathcal{E}$ – i.e. a proof of $\{P : s\}_\alpha \vdash Q : t$ – to the morphism α in $\mathcal{C}$.

2. π is a fibration, with the α^* as reindexing functors: that is, if we have $\alpha : s \to t$ and $\beta : t \to u$, then, given

   $$\{P\}_{\alpha;\beta} \vdash R$$

   we have

   $$\{P\}_\alpha \vdash \beta^* R$$

   and

   $$\{\beta^* R\}_\beta \vdash R$$

   The first of these follows from the right rule for α^* (together with the equivalence $\{\Gamma\}_{\alpha;\beta} \approx \{\{\Gamma\}_\alpha\}_\beta$) and the second is just the left rule for α^*.

3. The fibres of π are Heyting algebras. This follows straightforwardly from the rules, and the fact that the fibres are partial orders.

4. $\alpha_!$ is left adjoint to α^*. This follows straightforwardly from the rules.

5. α^* commutes with $\wedge$ and $\top$. This follows either directly from the rules, or from the previous assertion.

We now define an interpretation $[\![\cdot]\!]_\cdot$ of the typed language into $\mathcal{E}$, by mapping *typed propositional letters* of $\mathfrak{L}$ into their equivalence classes: a tedious induction on complexity shows that $[\![P : t]\!]_t$ is the equivalence class of $P : t$ for *general* P.

Finally, we have our result: if we do not have $\{\Gamma : s\}_\alpha \vdash P : t$, then induction on the complexity of environments shows that there is no morphism from $[\![\Gamma]\!]_s$ to $[\![P]\!]_t$ over α. Consequently (by the adjunction and the properties of fibred categories)

$$[\![\{\Gamma\}_\alpha]\!]_t \not\leq [\![P]\!]_t,$$

which is what was to be proved.

4 Semantics: The Classical System

We can now prove analogous results for our classical system.

The semantics of this logic should be a category, fibred over a labelled transition system (regarded as a category together with a functor into the free monoid over the set of action labels). The fibres should be models of classical logic (i.e. Boolean algebras), and the reindexing functors should be boolean algebra homomorphisms. These conditions mean that we have left and right adjoints to the substitution functors.

Definition 17. Let $\mathcal{C}$ be a category with pullbacks. A *classical Reiter category* over $\mathcal{C}$ is a category fibred by Boolean algebras over $\mathcal{C}$: the reindexing functors α^* should be Boolean algebra homomorphisms, and should have both left and right adjoints, α^* and α_*, and the adjoints should satisfy the following Beck-Chevalley conditions [8, p. 97]:

$$(\alpha)^*\beta_!P \dashv\vdash (s \times \beta)_!(\alpha \times_t u)^*P$$
$$\alpha)^*\beta_*P \dashv\vdash (s \times \beta)_*(\alpha \times_t u)^*P$$

for morphisms $\alpha : s \to t, \beta : u \to t$ in the base and for $P : u$.

We make the usual definitions, as in the intuitionistic case: only the definition of $[\![\cdot]\!]_t$ is appreciably different.

Definition 18. An *assignment* is an assignment, to every $t \in \mathrm{Ob}(\mathcal{C})$ and every atomic $P \in \mathfrak{L}$, an element $[\![P]\!]_t \in \mathrm{Ob}(\mathcal{E}_t)$.

Given an assignment, we can define, for each $P : t$, its semantic value $[\![P]\!]_t$ by induction on its syntactic complexity: the intuitionistic operators are interpreted in the usual way, α^* is interpreted as the reindexing functor (also written α^*), $\alpha_!$ is interpreted as the left adjoint, and α_* the right adjoint, to reindexing.

We also define $[\![\Gamma]\!]_t$ for contexts: here we have to make a distinction between left and right contexts, since $\{\cdot\}_\alpha$ is intepreted differently on the left and on the right.

Definition 19. The semantic value of a left context is given by the clauses

- $[\![\Gamma]\!]_t = [\![P]\!]_t$ if $\Gamma = P$
- $[\![\Gamma, \Gamma']\!]_t = [\![\Gamma]\!]_t \wedge [\![\Gamma']\!]_t$
- $[\![\{\Gamma : u\}^\beta]\!]_t = \beta^*([\![\Gamma]\!]_u)$ (for $\beta : t \to u$).

The semantic value of a right context is given by the clauses

- $[\![\Delta]\!]_t = [\![P]\!]_t$ if $\Delta = P$
- $[\![\Delta, \Delta']\!]_t = [\![\Delta]\!]_t \vee [\![\Delta']\!]_t$
- $[\![\{\Delta : u\}^\beta]\!]_t = \beta^*([\![\Delta]\!]_u)$ (for $\beta : t \to u$).

And, finally, a definition of semantic entailment:

Definition 20.

$$\Gamma : t \Vdash \Delta : t$$

iff

$$[\![\Gamma]\!]_t \leq [\![\Delta]\!]_t$$

for every assignment.

4.1 Soundness

First a pair of lemmas:

Lemma 11 (Semantic Monotonicity: Classical Case). *If $(\Gamma[\Delta : u]) : t$ is a left or right context, and if $[\![\Delta]\!]_u \leq [\![\Delta']\!]_u$ for some $\Delta' : u$, then $[\![\Gamma[\Delta]]\!]_t \leq [\![\Gamma[\Delta']]\!]_t$.*

Proof. The obvious induction, using the functoriality of $\alpha_!$ or α_*, as appropriate, and α^*.

Lemma 12. *Up to equivalence of contexts and applications of the classical structural rules, every classical marked context $\Gamma[: t] : s$ is equivalent to one of the form*

$$\Gamma' : s, \{[: t]\}^\alpha$$

for some $\alpha : s \to t$.

Proof. We can use the split and join rules to reduce $\Gamma[: t] : s$ to a context of the form

$$\Gamma' : s, \left\{ \cdots \{[: t]\}^{\beta_k} \cdots \right\}^{\beta_1},$$

where the context after the comma consists of a number of nested $\{\cdot\}^{\beta_i}$ with nothing in them apart from a single occurrence of $[: t]$: we then use the category-theoretic rules, together with the rules for equivalence of contexts, to compose the β_i into a single morphism α.

4.1.1 The Structural Rules

We now prove the soundness of the structural rules, that is, the rules in Table 2, together with joinL, joinR, splitL, and splitR.

Proposition 13. *The rules for the equivalence of left and right contexts in Table 2 are sound with respect to $\Vdash$: that is, if $\Gamma : t \approx \Gamma' : t$, then $[\![\Gamma]\!]_t = [\![\Gamma']\!]_t$ (and so, in particular, if $\Gamma : t \approx \Gamma' : t$ and $\Delta : t \approx \Delta' : t$, we have $\Gamma \Vdash \Delta$ iff $\Gamma' \Vdash \Delta'$).*

Proof. This follows from the functoriality of $\alpha_!$, α_*, and α^* and the definitions of the semantic value of contexts.

Proposition 14. *The rules joinL, joinR, splitL, and splitR, are sound for* $\Vdash\vdash$.

Proof. Clear.

4.1.2 The Standard Classical Rules

Proposition 15. *The rules LW, RW, LC, RC,* $\vee L$, $\vee R$, $\wedge L$ *and* $\wedge R$ *are sound for* $\Vdash\vdash$.

Proof. Standard, using semantic monotonicity as necessary.

Proposition 16. *The rules* $\neg L$ *and* $\neg R$ *are sound for* $\Vdash\vdash$.

Proof. Consider ¬L: ¬R is entirely similar. Using Lemma 12, we can assume without loss of generality that the premise of the rule is of the form

$$\Gamma \vdash \{P\}^{\alpha}, \Delta$$

and the conclusion is of the form

$$\Gamma, \{\neg P\}^{\alpha} \vdash \Delta.$$

Suppose that $\alpha : s \to t$. We have, then, to prove that if

$$[\![\Gamma]\!]_s \leq [\![\{P\}^{\alpha}]\!]_s \vee [\![\Delta]\!]_s$$

then

$$[\![\Gamma]\!]_s \wedge [\![\{\neg P\}^{\alpha}]\!]_s \leq [\![\Delta]\!]_s.$$

But, since

$$\begin{aligned} [\![\{\neg P\}^{\alpha}]\!]_s &= \alpha^*([\![\neg P]\!]_t) \\ &= \alpha^*(\neg [\![P]\!]_t) \\ &= \neg \alpha^* [\![P]\!]_t \end{aligned}$$

(since α^* is a Boolean algebra homomorphism), we have the result.

4.1.3 The Functorial Rules

In this section we consider the rules for α^*, $\alpha_!$ and α_*.

Proposition 17. *The rules $\alpha^* L$ and $\alpha^* R$ are sound for the classical semantics.*

Proof. By definition, $[\![\alpha^* P]\!]_s = \alpha^* [\![P]\!]_t = [\![\{P\}^\alpha]\!]_s$, and the result follows.

Proposition 18. *The rules $\alpha_* L$ and $\alpha_! R$ are sound for the classical semantics.*

Proof. Consider $\alpha_!$R: by semantic monotonicity, it suffices to prove that, for any $P : s$ and for $\alpha : s \to t$,

$$\begin{aligned} [\![P]\!]_s &\leq [\![\{\alpha_! P\}^\alpha]\!]_s \\ &= \alpha^* [\![\alpha_! P]\!]_t \\ &= \alpha^* \alpha_! [\![P]\!]_s, \end{aligned}$$

but this is simply the unit of the adjunction $\alpha_! \dashv \alpha^*$.

The proof for α^*L is dual.

Proposition 19. *The classical rule $\alpha_! L$ is sound for the classical semantics.*

Proof. By Lemma 12, we can assume that the conclusion is of the form

$$\Gamma, \{\alpha_! P\}^\beta \vdash \Delta,$$

where $P : s$, $\alpha : s \to t$, and $\beta : u \to t$. This means that the premise is of the form

$$\{\Gamma\}^{\alpha \times_t u}, \{P\}^{s \times_t \beta} \vdash \{\Delta\}^{\alpha \times_t u}$$

Now the semantic value of the premise is

$$(\alpha \times_t u)^* [\![\Gamma]\!]_u \wedge (s \times_t \beta)^* [\![P]\!]_s \leq (\alpha \times_t u)^* [\![\Delta]\!]_u,$$

whereas the semantic value of the conclusion is

$$[\![\Gamma]\!]_u \wedge \beta^* \alpha_! [\![P]\!]_s \leq [\![\Delta]\!]_u,$$

which, by the Beck-Chevalley condition, is equivalent to

$$[\![\Gamma]\!]_u \wedge (\alpha \times_t u)_! (s \times_t \beta)^* [\![P]\!]_s \leq [\![\Delta]\!]_u;$$

the result follows from the adjunction $\alpha_! \dashv \alpha^*$, together with standard properties of Boolean algebras.

Proposition 20. *The classical rule $\alpha_* R$ is sound for the classical semantics.*

Proof. Dual to the above.

4.1.4 The Cut Rule

Proposition 21. *The cut rule is sound for the classical semantics.*

Proof. Suppose that $\boldsymbol{\nu}(\Delta[]) = \alpha : s \to t$, and that $\boldsymbol{\nu}(\Gamma') = \beta : u \to t$. We apply Lemma 12, and can assume that the rule application is of the form

$$\frac{\Gamma \vdash \Delta, \{P\}^{\alpha} \quad \{P\}^{\beta}, \Gamma' \vdash \Delta'}{\{\Gamma\}^{s \times_t \beta}, \{\Gamma'\}^{\alpha \times_t u} \vdash \{\Delta\}^{s \times_t \beta}, \{\Delta'\}^{\alpha \times_t u}}$$

So, the semantic values of the premises of the cut rule are

$$[\![\Gamma]\!]_s \leq [\![\Delta]\!]_s \vee \alpha^* [\![P]\!]_t$$

and

$$[\![\Gamma']\!]_t \wedge \beta^* [\![P]\!]_u \leq [\![\Delta']\!]_u,$$

from which we can derive, respectively,

$$(s \times_t \beta)^* [\![\Gamma]\!]_s \leq (s \times_t \beta)^* [\![\Delta]\!]_s \vee (\alpha \times_t \beta)^* [\![P]\!]_t$$

and

$$(\alpha \times_t u)^* [\![\Gamma']\!]_t \wedge (\alpha \times_t \beta)^* [\![P]\!]_u \leq (\alpha \times_t u)^* [\![\Delta']\!]_u,$$

whereas the semantic value of the conclusion is

$$(s \times_t \beta)^* [\![\Gamma]\!]_s \wedge (\alpha \times_t u)^* [\![\Gamma']\!]_s \leq (s \times_t \beta)^* [\![\Delta]\!]_s \vee (\alpha \times_t u)^* [\![\Delta']\!]_s;$$

soundness follows from standard Boolean algebra.

So, putting all these results together, we have

Theorem 3. *The classical semantics is sound for the classical calculus.*

4.2 Completeness

Theorem 4. *The classical semantics is complete: that is, if, for a given base category $\mathcal{C}$, and for an object t of $\mathcal{C}$,*

$$[\![\Gamma]\!]_t \leq [\![\Delta]\!]_t$$

for two contexts $\Gamma : t$ and $\Delta : t$, then

$$\Gamma \vdash \Delta.$$

This theorem will be proved by constructing a term, or generic, model, which we define as follows.

Definition 21. Let $\mathcal{C}$ be a category with fibre products. The term model, $\mathcal{E}_{\mathcal{C}}$, over $\mathcal{C}$ is given by the following data:

the objects of the model are given by pairs $P : s$, where P is a proposition and s is an object of $\mathcal{C}$

morphisms between $P : s$ and $Q : t$ are given by proofs

$$P \vdash \{Q\}^{\alpha}$$

for some morphism $\alpha : s \to t$ of $\mathcal{C}$. Two such morphisms are equal iff their source and target are the same, and the corresponding morphisms of $\mathcal{C}$ are equal.

the composition of morphisms corresponding to proofs

$$\begin{array}{c}\Pi\\ \vdots\\ P : s \vdash \{Q : t\}^{\alpha}\end{array} \quad \text{and} \quad \begin{array}{c}\Pi'\\ \vdots\\ Q : t \vdash \{R : u\}^{\beta}\end{array}$$

is given by the proof

$$\dfrac{\dfrac{\begin{array}{c}\Pi\\ \vdots\\ P : s \vdash \{Q : t\}^{\alpha}\end{array} \qquad \dfrac{\dfrac{\begin{array}{c}\Pi'\\ \vdots\\ Q : t \vdash \{R : u\}^{\beta}\end{array}}{\{Q : t\}^{\alpha} \vdash \left\{\{R : u\}^{\beta}\right\}^{\alpha}}\{\}^{\alpha}}{\{Q : t\}^{\alpha} \vdash \{R : u\}^{\alpha\circ\beta}} \approx}{P : s, \{\}^{\alpha} \vdash \{\}^{\alpha}, \{R : u\}^{\alpha\circ\beta}}\,\text{cut}}{P : s \vdash \{R : u\}^{\alpha\circ\beta}} \approx$$

identity morphisms are given by the proofs

$$\dfrac{\dfrac{}{P : t \vdash P : t}\,\text{Axiom}}{P : t \vdash \{P : t\}^{\text{Id}}} \approx$$

the display functor p, maps a typed propositions $P : t$ to the object t, and a proof of $P : s \vdash \{Q : t\}^{\alpha}$ to the morphism $\alpha : s \to t$.

liftings of base morphisms are chosen as follows. Let $\alpha : s \to t$ be a morphism in the base, and let $P : t$ be an object of the total category over t: let the lifting of α be the following proof:

$$\dfrac{\dfrac{}{\{P : t\}^{\alpha} \vdash \{P : t\}^{\alpha}}\,\text{Axiom}}{(\alpha^{*}P : t) : s \vdash \{P : t\}^{\alpha}}\,\alpha^{*}\text{L}$$

In the indexed category viewpoint, this corresponds to using α^{*} as substitution functors.

left and right adjoints to the substitution functors are given by $\alpha_!$ and α_*.

We now prove

Proposition 22. *$\mathcal{E}_\mathcal{C}$ is a classical Reiter category.*

Proof. It is clear than $\mathcal{E}_\mathcal{C}$ is a category (equality of morphisms is so strong that laws like associativity are immediate). It is likewise clear that p is a functor. We have to check that the liftings are Cartesian: so, consider composable morphisms $\alpha : s \to t$ and $\beta : t \to u$ in the base, together with a proof Π of $P : s \vdash \{R : u\}^{\beta\circ\alpha}$ lying over $\alpha \circ \beta$. We need to produce a proof of $P : s \vdash \{(\alpha^* R) : t\}^\alpha$ (commutativity of the resulting diagram is trivial). But this is immediate:

$$\dfrac{\dfrac{\begin{array}{c}\Pi\\ \vdots\\ P : s \vdash \{R : t\}^{\alpha\circ\beta}\end{array}}{P : s \vdash \left\{\{R : t\}^\beta\right\}^\alpha}\approx}{P : s \vdash \{(\beta^* R) : t\}^\alpha}\,\beta^*\mathrm{R}$$

This establishes the functoriality of the α^*. Consequently, $\mathcal{E}_\mathcal{C}$ is fibred over $\mathcal{C}$: the fibres are Boolean algebras, because the inference rules are a superset of the normal classical inference rules.

We need to show that $\alpha_!$ and α_* are left and right adjoint functors to the α^*. Functoriality is easy. For example, the following construction, which produces a proof of $\alpha_* P \vdash \alpha_* Q$ from a proof of $P \vdash Q$, establishes functoriality for α_*:

$$\dfrac{\dfrac{\begin{array}{c}\Pi\\ \vdots\\ P : s \vdash Q : s\end{array}}{\{(\alpha_* P) : t\}^\alpha \vdash Q : s}\,\alpha_*\mathrm{L}}{(\alpha_* P) : t \vdash (\alpha_* Q) : t}\,\alpha_*\mathrm{R}$$

Given functoriality, we only need to establish the unit and counit for each adjunction. The unit for $\alpha_! \dashv \alpha^*$ is proven like this:

$$\dfrac{\dfrac{\dfrac{}{P : s \vdash P : s}\,\mathrm{Axiom}}{P : s \vdash \{\alpha_! P : s\}^\alpha}\,\alpha_!\mathrm{R}}{P : s \vdash \alpha^*\alpha_! P : s}\,\alpha^*\mathrm{R}$$

and the counit like this:

$$\dfrac{\dfrac{\dfrac{}{\{Q : t\}^\alpha \vdash \{Q : t\}^\alpha}\,\mathrm{Axiom}}{\alpha^* Q : t \vdash \{Q : t\}^\alpha}}{\alpha_!\alpha^* Q : t \vdash Q : t}\,\alpha_!\mathrm{L}$$

The proof of the adjunction for α_* is dual.

Finally, we must verify the Beck-Chevalley conditions: for $\alpha_!$, we prove these as follows. Suppose that $\alpha : s \to t$, $\beta : u \to t$, and that $P : u$. Then we have

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\{P\}^{\alpha \times_t u} \vdash \{P\}^{\alpha \times_t u}}\text{Axiom}}{\{P\}^{\alpha \times_t u} \vdash (\alpha \times_t u)^* P}(\alpha \times_t u)^*\mathrm{R}}{\{P\}^{\alpha \times_t u} \vdash \{(s \times_t \beta)_!(\alpha \times_t u)^* P\}^{s \times_t \beta}}(s \times_t \beta)_!\mathrm{R}}{\{\beta_! P\}^{\alpha} \vdash (s \times_t \beta)_!(\alpha \times_t u)^* P}\beta_!\mathrm{L}}{\alpha^* \beta_! P \vdash (s \times_t \beta)_!(\alpha \times_t u)^* P}\alpha^*\mathrm{L}$$

and

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\{P\}^{\alpha \times_t u} \vdash \{P\}^{\alpha \times_t u}}\text{Axiom}}{(\alpha \times_t u)^* P \vdash \{P\}^{\alpha \times_t u}}(\alpha \times_t u)^*\mathrm{L}}{(\alpha \times_t u)^* P \vdash \left\{\{\beta_! P\}^{\beta}\right\}^{\alpha \times_t u}}\beta_!\mathrm{R}}{(\alpha \times_t u)^* P \vdash \{\beta_! P\}^{(\alpha \times_t u) \circ \beta}}\approx}{(\alpha \times_t u)^* P \vdash \{\beta_! P\}^{(s \times_t \beta) \circ \alpha}}\approx}{(\alpha \times_t u)^* P \vdash \{\{\beta_! P\}^{\alpha}\}^{s \times_t \beta}}\approx}{(\alpha \times_t u)^* P \vdash \{\alpha^* \beta_! P\}^{s \times_t \beta}}\alpha^*\mathrm{R}}{(s \times_t \beta)_!(\alpha \times_t u)^* P \vdash \alpha^* \beta_! P}(s \times_t \beta)_!\mathrm{L}$$

The proofs of the Beck-Chevalley conditions for α_* are dual. This concludes the proof that $\mathcal{E}_{\mathcal{C}}$ is a classical Reiter category.

Definition 22. Let Γ be a left context: let the *propositionalisation* of Γ, $\overline{\Gamma}$, be defined by

$$\begin{aligned} \overline{P} &= P \\ \overline{\Gamma', \Gamma''} &= \overline{\Gamma'} \wedge \overline{\Gamma''} \\ \overline{\{\Gamma\}^{\alpha}} &= \alpha^* \overline{\Gamma} \end{aligned}$$

Similarly, if Δ is a left context, the *propositionalisation* of Δ, $\overline{\Delta}$, is defined by

$$\begin{aligned} \overline{P} &= P \\ \overline{\Delta', \Delta''} &= \overline{\Delta'} \vee \overline{\Delta''} \\ \overline{\{\Delta\}^{\alpha}} &= \alpha^* \overline{\Delta} \end{aligned}$$

Lemma 13. *For any* Γ *and* Δ*,*

$$\Gamma \vdash \Delta$$

iff

$$\overline{\Gamma} \vdash \overline{\Delta}$$

Proof. The obvious induction.

Proof of Theorem 4. Suppose that $\Gamma : t \Vdash \Delta : t$. Define an interpretation of the language in $\mathcal{E}_{\mathcal{C}}$ by sending $P : t$ to $P : t$ as an object of $\mathcal{E}_{\mathcal{C}}$. We establish, by induction, that, with respect to this interpretation, $[\![\Gamma]\!]_t = \overline{\Gamma}$, and $[\![\Delta]\!]_t = \overline{\Delta}$. Since $\Gamma : t \Vdash \Delta : t$, we must have $[\![\Gamma]\!]_t \leq [\![\Delta]\!]_t$, and, consequently, $\overline{\Gamma} \leq \overline{\Delta}$: by the definition of $\mathcal{E}_{\mathcal{C}}$, this means that $\overline{\Gamma} \vdash \overline{\Delta}$. By the lemma, we have $\Gamma \vdash \Delta$.

5 Proof Theory

In this section we prove cut elimination for our systems: first the intuitionistic system, and then the classical. Although the result is proved in broadly the same way for both systems, the details are rather different.

5.1 The Intuitionistic System

We carry out an induction simultaneously on cut depth and on the complexity of the cut formula: as usual, we only present the interesting cases.

5.1.1 The Base Cases

The base cases of the induction are where one premise is either an axiom, $\bot$ L, or $\top$R: apart from some care over the more elaborate contexts, these cases are handled exactly as in the cut elimination theorem for intuitionistic logic, and we will not describe them here.

5.1.2 Permuting the Cut Upwards

We now turn to the inductive cases: that is, the case where neither of the premises of a topmost cut is a leaf node. First, then, we consider cases where the cutformula is non-principal on the left side. The cases where we have rule applications of the normal connectives are, with a little extra care because of our more elaborate contexts, more or less standard. The cases of interest are the applications of α^* and $\alpha_!$: we show the argument for the former. The cut will look like

$$\dfrac{\begin{array}{cc} \Pi & \begin{array}{c}\Pi' \\ \vdots \\ \{\Gamma'[P]\}_\alpha \vdash Q\end{array} \\ \vdots & \dfrac{}{\Gamma'[P] \vdash \alpha^* Q}\,\alpha^*\mathrm{R} \\ \Gamma \vdash P & \end{array}}{\Gamma'[\Gamma/P] \vdash \alpha^* Q}\,\mathrm{cut}$$

and we can permute the cut upwards:

$$\dfrac{\dfrac{\begin{array}{c}\Pi\\ \vdots\\ \Gamma \vdash P\end{array} \quad \begin{array}{c}\Pi'\\ \vdots\\ \{\Gamma'[P]\}_\alpha \vdash Q\end{array}}{\{\Gamma'[\Gamma/P]\}_\alpha \vdash Q}\,\mathrm{cut}}{\Gamma'[\Gamma/P] \vdash \alpha^* Q}\,\alpha^*\mathrm{R}$$

The argument for $\alpha_!\mathrm{R}$ is very similar. In all of these cases we have

Result Cut depth reduced by 1, cut rank unchanged.

5.1.3 Principal against Principal

Here we deal with the case where the cut is principal on both sides. The intuitionistic connectives and structural rules are handled in the normal manner (we have to use the multicut trick to handle LC): we deal solely with the cases of $\alpha_!$ and α^*. The cases are as follows:

α^* We have

$$\dfrac{\dfrac{\begin{array}{c}\Pi\\ \vdots\\ \{\Gamma\}_\alpha \vdash P\end{array}}{\Gamma \vdash \alpha^* P}\,\alpha^*\mathrm{R} \quad \dfrac{\begin{array}{c}\Pi'\\ \vdots\\ \Gamma'[P] \vdash Q\end{array}}{\Gamma'[\{\alpha^* P\}_\alpha] \vdash P}\,\alpha^*\mathrm{L}}{\Gamma'[\{\Gamma\}_\alpha] \vdash P}\,\mathrm{cut}$$

We replace this with

$$\dfrac{\begin{array}{c}\Pi\\ \vdots\\ \{\Gamma\}_\alpha \vdash P\end{array} \quad \begin{array}{c}\Pi'\\ \vdots\\ \Gamma'[P] \vdash Q\end{array}}{\Gamma'[\{\Gamma\}_\alpha] \vdash P}\,\mathrm{cut}$$

Result Cut depth reduced by two, cut rank reduced by 1.

$\alpha_!$

$$\dfrac{\dfrac{\begin{array}{c}\Pi\\ \vdots\\ \Gamma \vdash P\end{array}}{\{\Gamma\}_\alpha \vdash \alpha_! P}\,\alpha_!\mathrm{L} \quad \dfrac{\begin{array}{c}\Pi'\\ \vdots\\ \Gamma'[\{P\}_\alpha] \vdash Q\end{array}}{\Gamma'[\alpha_! P] \vdash Q}\,\alpha_!\mathrm{L}}{\Gamma[\{\Gamma\}_\alpha] \vdash Q}\,\mathrm{cut}$$

We replace this with

$$\dfrac{\begin{array}{cc}\Pi & \Pi' \\ \vdots & \vdots \\ \Gamma \vdash P & \Gamma'[\{P\}_\alpha] \vdash Q\end{array}}{\Gamma'[\{P\}_\alpha] \vdash Q}\,\text{cut}$$

Result Cut rank reduced by 1, cut depth reduced by 2.

So, finally, we have

Theorem 5. *The cut rule can be eliminated in the intuitionistic system.*

Proof. By the cases above.

5.2 The Classical System

We will make the usual induction over cut rank and cut depth. The classical system needs more care than the intuitionistic system, mainly because of the constraints on the cut rule (that is, that the cutformulae should be similarly nested in both premises). We need a definition of the rank of a formula:

Definition 23. Let A be a formula in the classical language. We define the rank of A, $|A|$, by the clauses

$$\begin{aligned}
|P| &= 1 && \text{for } P \text{ atomic} \\
|\neg P| &= |P| + 1 \\
|\alpha^* P| &= |P| + 1 \\
|\alpha_! P| &= |P| + 1 \\
|\alpha_* P| &= |P| + 1 \\
|P \wedge Q| &= 1 + \max(|P|, |Q|) \\
|P \vee Q| &= 1 + \max(|P|, |Q|) \\
|P \to Q| &= 1 + \max(|P|, |Q|)
\end{aligned}$$

There are, as usual, three cases to consider in the cut elimination process: where either of the premises is an axiom, $\bot$ L, or $\top$R, where the cutformula is non-principal in either of the premises, and were the cutformula is principal in both of the premises.

5.2.1 The Base Cases

The base cases are $\bot$ L, $\top$R, and Axiom. The case of $\bot$ L looks like this:

$$\dfrac{\dfrac{}{\Gamma[\bot] \vdash \Delta[P]}\,\bot\,\text{L} \qquad \begin{array}{c}\vdots \\ \Gamma'[P] \vdash \Delta'\end{array}}{\Gamma[\bot], \Gamma'[\varepsilon] \vdash \Delta[\varepsilon], \Delta'}\,\text{cut}$$

and we replace it with

$$\frac{}{\Gamma[\bot], \Gamma'[\varepsilon] \vdash \Delta[\varepsilon], \Delta'} \bot \mathrm{L}$$

The other base cases are similar.

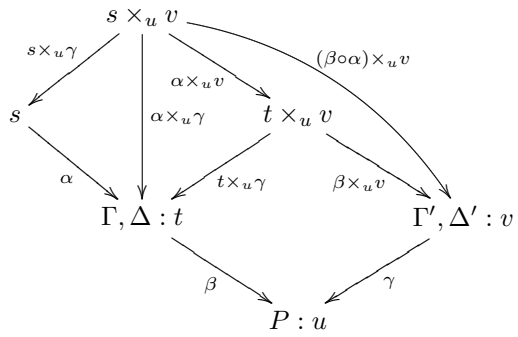
5.2.2 Moving the Cut Upwards

Most of the cases where we move a cut upwards are handled in exactly the same way as in classical logic, and are omitted: however, we do have to take care when the rule that we cut against changes the typing of the entailment. The cases to consider are as follows.

$\{\}^\alpha$ Here the cut application looks like

$$\frac{\dfrac{\begin{array}{c}\Pi\\ \vdots\\ \Gamma \vdash \Delta[P]\end{array}}{\{\Gamma\}^\alpha \vdash \{\Delta[P]\}^\alpha}\{\}^\alpha \qquad \begin{array}{c}\Pi'\\ \vdots\\ \Gamma'[P] \vdash \Delta'\end{array}}{\{\Gamma\}^{\alpha \times_u \gamma}, \{\Gamma'[\varepsilon]\}^{(\beta \circ \alpha) \times_t u} \vdash \{\Delta[\varepsilon]\}^{\alpha \times_u \gamma}, \{\Delta'\}^{(\beta \circ \alpha) \times_t u}} \mathrm{cut}$$

where we have $P : u$, $\Gamma, \Delta : t$, $\Gamma', \Delta' : v$, $\alpha : s \to t$, $\boldsymbol{\nu}(\Delta[]) = \beta : u \to t$, and $\boldsymbol{\nu}(\Gamma'[]) = \gamma : v \to t$. Diagramatically, the situation looks like this.



We replace the above application of cut with

$$\frac{\dfrac{\dfrac{\dfrac{\begin{array}{c}\Pi\\ \vdots\\ \Gamma \vdash \Delta[P]\end{array} \qquad \begin{array}{c}\Pi'\\ \vdots\\ \Gamma'[P] \vdash \Delta'\end{array}}{\{\Gamma\}^{t \times_u \gamma}, \{\Gamma'[\varepsilon]\}^{\beta \times_u v} \vdash \{\Delta[\varepsilon]\}^{t \times_u \gamma}, \{\Delta'\}^{\beta \times_u v}} \mathrm{cut}}{\left\{\{\Gamma\}^{t \times_u \gamma}, \{\Gamma'[\varepsilon]\}^{\beta \times_u v}\right\}^{\alpha \times_u v} \vdash \left\{\{\Delta[\varepsilon]\}^{t \times_u \gamma}, \{\Delta'\}^{\beta \times_u v}\right\}^{\alpha \times_u v}}\{\}^{\alpha \times_u v}}{\left\{\{\Gamma\}^{t \times_u \gamma}\right\}^{\alpha \times_u v}, \left\{\{\Gamma'[\varepsilon]\}^{\beta \times_u v}\right\}^{\alpha \times_u v} \vdash \left\{\{\Delta[\varepsilon]\}^{t \times_u \gamma}\right\}^{\alpha \times_u v}, \left\{\{\Delta'\}^{\beta \times_u v}\right\}^{\alpha \times_u v}} \mathrm{splitL}, \mathrm{splitR}}{\{\Gamma\}^{\alpha \times_u \gamma}, \{\Gamma'[\varepsilon]\}^{(\beta \circ \alpha) \times_u v} \vdash \{\Delta[\varepsilon]\}^{\alpha \times_u \gamma}, \{\Delta'\}^{(\beta \circ \alpha) \times_u v}} \approx$$

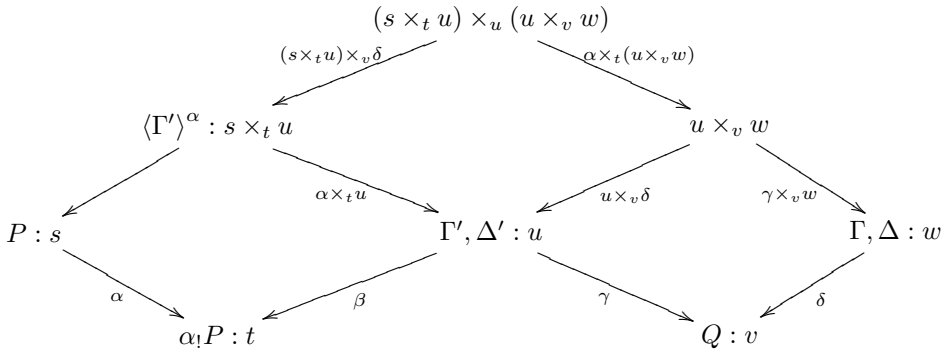
Result Cut rank unchanged: cut depth reduced by 1.

$\alpha^*\mathbf{L}$, $\alpha^*\mathbf{R}$, $\alpha_!\mathbf{R}$, $\alpha_*\mathbf{L}$ These rules change the nesting of the contexts, but only around the principal formula: since we are considering rule applications in which the cutformula is not principal, this does not affect the cuts that we are considering.

$\alpha_!\mathbf{L}$ The cut application looks like this:

$$\dfrac{\begin{array}{cc}\begin{array}{c}\Pi\\ \vdots\\ \Gamma \vdash \Delta[Q : v] : w\end{array} & \dfrac{\begin{array}{c}\Pi'\\ \vdots\\ \langle \Gamma'[P][Q] \rangle^{\alpha} \vdash \{\Delta'\}^{\alpha \times_t u}\end{array}}{\Gamma'[(\alpha_! P) : t][Q : v] \vdash \Delta' : u}\,\alpha_!\mathrm{L}\end{array}}{\{\Gamma\}^{\gamma \times_v w}, \{\Gamma'[\alpha_! P][\varepsilon]\}^{u \times_v \delta} \vdash \{\Delta[\varepsilon]\}^{\gamma \times_v w}, \{\Delta'\}^{u \times_v \delta}}\,\mathrm{cut}$$

where the category theory, and typing, looks like this:



We replace the above cut with the following (we use Lemma 7 to justify the typing):

$$\dfrac{\begin{array}{cc}\begin{array}{c}\Pi\\ \vdots\\ \Gamma \vdash \Delta[Q]\end{array} & \begin{array}{c}\Pi'\\ \vdots\\ \langle \Gamma'[P][Q] \rangle^{\alpha} \vdash \{\Delta'\}^{\alpha \times_t u}\end{array}\end{array}}{\begin{array}{c}\{\Gamma\}^{(\gamma \times_v w) \circ (\alpha \times_t (u \times_v w))}, \left\{\langle \Gamma'[P][\varepsilon] \rangle^{\alpha}\right\}^{(s \times_t u) \times_v \delta} \vdash \\ \{\Delta[\varepsilon]\}^{(\gamma \times_v w) \circ ((\alpha \times_t (u \times_t w))}, \left\{\{\Delta'\}^{\alpha \times_t u}\right\}^{(s \times_t u) \times_v \delta}\end{array}}\,\mathrm{cut}$$

Now, by Lemma 9,

$$\left\{\langle \Gamma'[P][\varepsilon] \rangle^{\alpha}\right\}^{(s \times_t u) \times_v \delta} = \left\langle \{\Gamma'[P][\varepsilon]\}^{u \times_v \delta} \right\rangle^{\alpha};$$

since these two expressions are typographically equal, we can replace one by another in the proof without any rule application. Furthermore, we can

work on the last expression on the right, using the context equivalence rule $\left\{\{\cdot\}^{\phi}\right\}^{\psi} \approx \{\cdot\}^{\psi\circ\phi}$ and the fact that

$$(\alpha \times_t u) \circ ((s \times_t u) \times_v \delta = (u \times_v s) \circ (\alpha \times_t (u \times_v w)) :$$

so, we can continue the proof as follows:

$$\dfrac{\dfrac{\{\Gamma\}^{(\gamma\times_v w)\circ(\alpha\times_t(u\times_v w)}, \left\langle\{\Gamma'[P][\varepsilon]\}^{u\times_v\delta}\right\rangle^{\alpha} \vdash \{\Delta[\varepsilon]\}^{(\gamma\times_v w)\circ((\alpha\times_t(u\times_t w))}, \{\Delta'\}^{(u\times_v s)\circ(\alpha\times_t(u\times_v w))}}{\left\{\{\Gamma\}^{\gamma\times_v w}\right\}^{\alpha\times_t(u\times_v w)}, \left\langle\{\Gamma'[P][\varepsilon]\}^{u\times_v\delta}\right\rangle^{\alpha} \vdash \left\{\{\Delta\}^{\gamma\times_v w}\right\}^{\alpha\times_t(u\times_v w)}, \left\{\{\Delta'\}^{u\times_v\delta}\right\}^{\alpha\times_t(u\times_v w)}}\approx}{\{\Gamma\}^{\gamma\times_v w}, \{\Gamma'[\alpha_!P][\varepsilon]\}^{u\times_v\delta} \vdash \{\Delta[\varepsilon]\}^{\gamma\times_v w}, \{\Delta'\}^{u\times_v\delta}}\,\alpha_!\mathrm{L}$$

result Cut depth reduced by one, cut rank unchanged.

$\alpha_*\mathbf{L}$ Dual to the above.

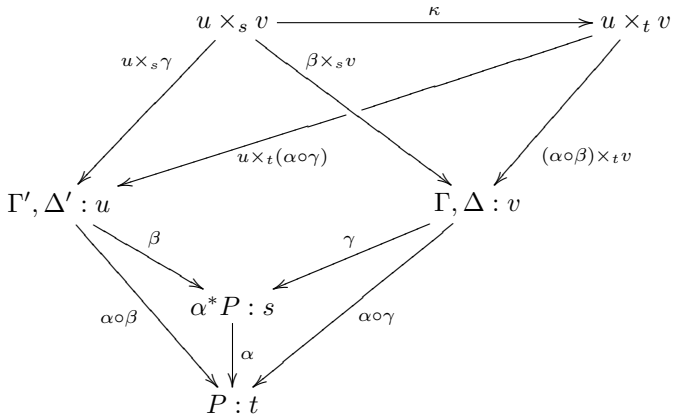
5.2.3 Principal against Principal

The final case is where the cutformula is principal in both sides. The classical connectives give few surprises: we will, then, only consider the functorial connectives.

α^* The cut here will look like

$$\dfrac{\dfrac{\begin{array}{c}\Pi\\ \vdots\\ \Gamma \vdash \Delta[\{P\}^{\alpha}]\end{array}}{\Gamma \vdash \Delta[\alpha^*P]}\,\alpha^*\mathrm{R} \qquad \dfrac{\begin{array}{c}\Pi'\\ \vdots\\ \Gamma'[\{P\}^{\alpha}] \vdash \Delta'\end{array}}{\Gamma'[\alpha^*P] \vdash \Delta'}\,\alpha^*\mathrm{L}}{\{\Gamma\}^{\beta\times_s v}, \{\Gamma'[\varepsilon]\}^{u\times_s\gamma} \vdash \{\Delta[\varepsilon]\}^{\beta\times_s v}, \{\Delta'\}^{u\times_s\gamma}}\,\mathrm{cut}$$

where the category theory, and typing, looks like



Here κ is the canonical morphism $u \times_s v \to u \times_t v$. Note that, because cut is applicable, the two occurrences of $\alpha^* P$ must be typed with the same object, and determinacy means that there can only be one α-typed morphism from that object: consequently, the two occurrences of P must be typed with the same object.

We replace the cut with

$$\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}\Pi \\ \vdots \\ \Gamma \vdash \Delta[\{P\}^\alpha]\end{array} \qquad \begin{array}{c}\Pi' \\ \vdots \\ \Gamma'[\{P\}^\alpha] \vdash \Delta'\end{array}}{\{\Gamma\}^{(\beta\circ\alpha)\times_t v}, \{\Gamma'[\varepsilon]\}^{u\times_t(\gamma\circ\alpha)} \vdash \{\Delta[\varepsilon]\}^{(\beta\circ\alpha)\times_t v}, \{\Delta'\}^{u\times_t(\gamma\circ\alpha)}}\,\text{cut}}{\left\{\{\Gamma\}^{(\beta\circ\alpha)\times_t v}, \{\Gamma'[\varepsilon]\}^{u\times_t(\gamma\circ\alpha)}\right\}^\kappa \vdash \left\{\{\Delta[\varepsilon]\}^{(\beta\circ\alpha)\times_t v}, \{\Delta'\}^{u\times_t(\gamma\circ\alpha)}\right\}^\kappa}\,\{\}^\kappa}{\left\{\{\Gamma\}^{(\beta\circ\alpha)\times_t v}\right\}^\kappa, \left\{\{\Gamma'[\varepsilon]\}^{u\times_t(\gamma\circ\alpha)}\right\}^\kappa \vdash \left\{\{\Delta[\varepsilon]\}^{(\beta\circ\alpha)\times_t v}\right\}^\kappa, \left\{\{\Delta'\}^{u\times_t(\gamma\circ\alpha)}\right\}^\kappa}\,\text{splitL,splitR}}{\{\Gamma\}^{\beta\times_s v}, \{\Gamma'[\varepsilon]\}^{u\times_s\gamma} \vdash \{\Delta[\varepsilon]\}^{\beta\times_s v}, \{\Delta'\}^{u\times_s\gamma}}\,\approx$$

Here the final application of $\approx$ comes from the equalities

$$(u \times_t (\alpha \circ \gamma)) \circ \kappa = u \times_s \gamma$$

and

$$((\alpha \circ \beta)\times_t) \circ \kappa = \beta \times_s v.$$

Result Cut rank reduced by one, cut depth reduced by one.

$\alpha_!$ For this case, we will need

Lemma 14. *Any classical marked context of the form* $\langle \Gamma[] \rangle^\alpha$ *is equivalent to*

$$\{\Gamma[]\}^{\alpha \times_t \boldsymbol{\nu}(\Gamma[])}.$$

Proof. We use Lemma 12 to express $\Gamma[]$ in the form $\{[]\}^{\boldsymbol{\nu}(\Gamma[])}, \Gamma''$: by Lemma 5, we have

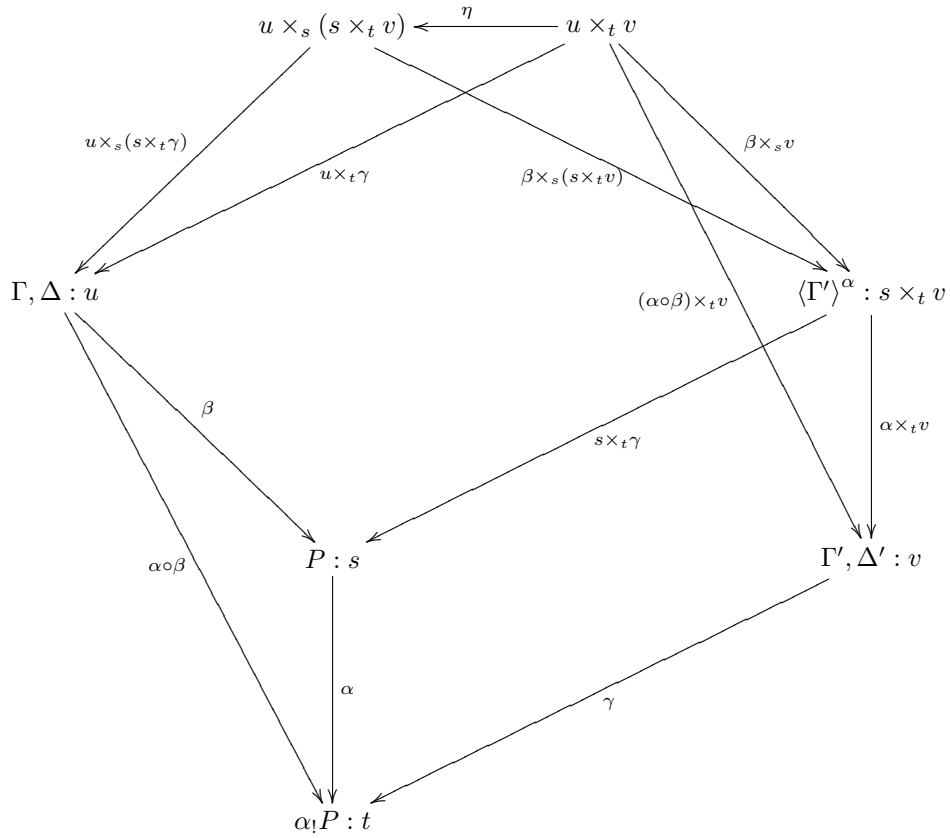
$$\langle \Gamma[] \rangle^\alpha \approx \left\langle \{[]\}^{\boldsymbol{\nu}(\Gamma[P])}, \Gamma'' \right\rangle^\alpha,$$

and the result follows from the definition of $\langle \cdot \rangle^{\cdot}$.

The cut looks like

$$\dfrac{\dfrac{\begin{array}{c}\Pi \\ \vdots \\ \Gamma \vdash \Delta[P]\end{array}}{\Gamma \vdash \Delta[\{\alpha_! P\}^\alpha]}\,\alpha_!\mathrm{R} \qquad \dfrac{\begin{array}{c}\Pi' \\ \vdots \\ \langle \Gamma'[P] \rangle^\alpha \vdash \{\Delta'\}^{\alpha \times_t v}\end{array}}{\Gamma'[\alpha_! P] \vdash \Delta'}}{\{\Gamma\}^{u \times_t \gamma}, \{\Gamma'[\varepsilon]\}^{(\alpha \circ \beta) \times_t v} \vdash \{\Delta[\varepsilon]\}^{u \times_t \gamma}, \{\Delta'\}^{(\alpha \circ \beta) \times_t v}}\,\mathrm{cut}$$

The category theory looks like



Here η is the map $\langle \pi_1, \langle \beta \circ \pi_1, \pi_2 \rangle \rangle$.

We replace the cut with the following:

$$\frac{\Gamma \vdash \Delta[P] \quad \langle \Gamma'[P] \rangle^{\alpha} \vdash \{\Delta'\}^{\alpha \times_t v}}{\{\Gamma\}^{u \times_s (s \times_t \gamma)}, \left\{\langle \Gamma'[\varepsilon] \rangle^{\alpha}\right\}^{\beta \times_s (s \times_t v)} \vdash \{\Delta[\varepsilon]\}^{u \times_s (s \times_t \gamma)}, \left\{\{\Delta'\}^{\alpha \times_t v}\right\}^{\beta \times_s (s \times_t v)}} \text{ cut}$$

We now apply Lemma 14, and can continue the proof as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}\{\Gamma\}^{u\times_s(s\times_t\gamma)}, \left\{\langle\Gamma'[\varepsilon]\rangle^{\alpha}\right\}^{\beta\times_s(s\times_t v)} \\ \vdash \{\Delta[\varepsilon]\}^{u\times_s(s\times_t\gamma)}, \left\{\{\Delta'\}^{\alpha\times_t v}\right\}^{\beta\times_s(s\times_t v)} \\ \vdots (\approx)^n \\ \{\Gamma\}^{u\times_s(s\times_t\gamma)}, \left\{\{\Gamma'[\varepsilon]\}^{\alpha\times_t v}\right\}^{\beta\times_s(s\times_t v)} \\ \vdash \{\Delta[\varepsilon]\}^{u\times_s(s\times_t\gamma)}, \left\{\{\Delta'\}^{\alpha\times_t v}\right\}^{\beta\times_s(s\times_t v)}\end{array}}{\left\{\{\Gamma\}^{u\times_s(s\times_t\gamma)}, \left\{\{\Gamma'[\varepsilon]\}^{\alpha\times_t v}\right\}^{\beta\times_s(s\times_t v)}\right\}^{\eta} \vdash \left\{\{\Delta[]\}^{u\times_s(s\times_t\gamma)}, \left\{\{\Delta'\}^{\alpha\times_t v}\right\}^{\beta\times_s(s\times_t v)}\right\}^{\eta}}\,\{\}^{\eta}}{\{\Gamma\}^{(u\times_s(s\times_t\gamma))\circ\eta}, \{\Gamma'[\varepsilon]\}^{(\alpha\times_t v)\circ(\beta\times_s(s\times_t v))\circ\eta} \vdash \{\Delta[\varepsilon]\}^{(u\times_s(s\times_t\gamma))\circ\eta}, \{\Delta'\}^{(\alpha\times_t v)\circ(\beta\times_s(s\times_t v))\circ\eta}}\,\approx}{\{\Gamma\}^{u\times_t\gamma}, \{\Gamma'[\varepsilon]\}^{(\alpha\circ\beta)\times_t v} \vdash \{\Delta[\varepsilon]\}^{u\times_t\gamma}, \{\Delta'\}^{(\alpha\circ\beta)\times_t v}}\,\text{splitL}$$

Result Cut rank decreased by one: cut depth decreased by one.

α_* Dual to the above.

So, finally, we have

Theorem 6. *The classical system satisfies cut elimination.*

Proof. We make an induction on cut rank, with an inner induction on cut depth. Consider an uppermost cut in a proof Pi: by induction, we can decrease the depth of the cut first, until we have either a cut against an axiom or a cut where both formulae are principal: in that case, we can decrease the cut rank. We can thus eliminate all of the topmost cuts, and so, by induction, we decrease all of the cuts.

6 Interpreting the Classical System in Reiter's Calculus

We now show how our classical system is related to Reiter's calculus. We first face a merely technical problem: Reiter defines progression and regression operators between any situation and the initial situation, whereas our calculus will require such operators between *any* pair of situations $\alpha \sqsubset \beta$. This can be dealt with, however, by realising that any situation is the initial situation of its own future. More technically, we make definitions as follows.

Suppose that we have a fixed sequence of actions $\boldsymbol{\alpha}$, then we can define a map of the ground situations to themselves by prefixing with α.

Definition 24. Define a map $(\boldsymbol{\alpha};\cdot) : \mathit{groundSit} \to \mathit{groundSit}$ by

$$\begin{aligned}\boldsymbol{\alpha};\mathsf{S}_0 &= \mathsf{S}_{\boldsymbol{\alpha}}\\ \boldsymbol{\alpha};\mathsf{do}(\beta,s) &= \mathsf{do}(\beta,\boldsymbol{\alpha};s)\end{aligned}$$

Definition 25. Suppose that $\mathsf{S}_{\boldsymbol{\alpha}} \sqsubseteq \mathsf{S}_{\boldsymbol{\beta}}$: we must then have $\beta = \alpha;\beta'$ for some β'. Define the interpretation

$$\iota_{\boldsymbol{\beta'},\boldsymbol{\beta}} : \mathfrak{L}_{\mathsf{S}_{\boldsymbol{\beta'}}} \to \mathfrak{L}_{\mathsf{S}_{\boldsymbol{\beta}}}$$

by $(\boldsymbol{\alpha};\cdot)$ on terms of sort *situation*, and by the identity on all other terms, predicates and functions; if $\boldsymbol{\alpha}$ is the empty sequence (i.e. if $\mathsf{S}_{\boldsymbol{\alpha}} = \mathsf{S}_0$) then let

$$\iota_{\boldsymbol{\beta}} : \mathfrak{L}_{\mathsf{S}_0} \to \mathfrak{L}_{\mathsf{S}_{\boldsymbol{\beta}}}$$

be the similarly defined interpretation with domain $\mathfrak{L}_{\mathsf{S}_0}$.

The following is immediate:

Proposition 23. *1. $\iota_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ is an isomorphism, for any $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.*

2. $\iota_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ restricts to the identity on $\mathcal{D}_{una}$.

We can now define suitable regression and progression maps R and η, between pairs of situations $\mathsf{S}_{\boldsymbol{\alpha}} \sqsubseteq \mathsf{S}_{\boldsymbol{\beta}}$, by transport of structure:

Definition 26. Given situations as above, define

$$\begin{aligned}R_{\boldsymbol{\beta},\boldsymbol{\alpha}} : \mathfrak{L}_{\mathsf{S}_{\boldsymbol{\beta}}} &\to \mathfrak{L}_{\mathsf{S}_{\boldsymbol{\alpha}}}\\ P &\mapsto \iota_{\boldsymbol{\alpha}}(R(\iota^{-1}_{\boldsymbol{\beta'},\boldsymbol{\beta}}(P)))\\ \eta_{\boldsymbol{\alpha},\boldsymbol{\beta}} : \mathfrak{L}_{\mathsf{S}_{\boldsymbol{\alpha}}} &\to \mathfrak{L}_{\mathsf{S}_{\boldsymbol{\beta}}}\\ P &\mapsto \iota_{\boldsymbol{\beta'},\boldsymbol{\beta}}(\eta_{\boldsymbol{\beta'}}(\iota^{-1}_{\boldsymbol{\alpha}}(P)))\end{aligned}$$

where, again, $\boldsymbol{\beta} = \boldsymbol{\alpha};\boldsymbol{\beta'}$.

Diagrammatically, this looks as follows:

$$\begin{array}{ccc}
 & \overset{R_{\boldsymbol{\beta},\boldsymbol{\alpha}}}{\leftarrow} & \\
\mathfrak{L}_{\mathsf{S}_{\boldsymbol{\alpha}}} & \underset{\eta_{\boldsymbol{\alpha},\boldsymbol{\beta}}}{\rightarrow} & \mathfrak{L}_{\mathsf{S}_{\boldsymbol{\beta}}} \\
\uparrow{\scriptstyle \iota_{\boldsymbol{\alpha}}} & & \uparrow{\scriptstyle \iota_{\boldsymbol{\beta'},\boldsymbol{\beta}}} \\
 & \overset{R}{\leftarrow} & \\
\mathfrak{L}_{\mathsf{S}_0} & \underset{\eta_{\boldsymbol{\beta'}}}{\rightarrow} & \mathfrak{L}_{\mathsf{S}_{\boldsymbol{\beta'}}}
\end{array} \tag{41}$$

We have

Proposition 24. *1. For $P \in \mathfrak{L}_{\mathsf{S}_{\boldsymbol{\alpha}}}$, $R(P) = R_{\boldsymbol{\alpha},[]}$*

2. $R_{\boldsymbol{\beta},\boldsymbol{\gamma}} \circ R_{\boldsymbol{\alpha},\boldsymbol{\beta}} = R_{\boldsymbol{\alpha},\boldsymbol{\gamma}}$, for $\mathsf{S}_{\boldsymbol{\gamma}} \sqsubset \mathsf{S}_{\boldsymbol{\beta}} \sqsubset \mathsf{S}_{\boldsymbol{\alpha}}$

3. $\eta_{\boldsymbol{\beta},\boldsymbol{\alpha}} \circ \eta_{\boldsymbol{\gamma},\boldsymbol{\beta}} = \eta_{\mathsf{S}_{\boldsymbol{\gamma}},\mathsf{S}_{\boldsymbol{\alpha}}}$, for $\mathsf{S}_{\boldsymbol{\gamma}} \sqsubset \mathsf{S}_{\boldsymbol{\beta}} \sqsubset \mathsf{S}_{\boldsymbol{\alpha}}$

Proof. For the first part, we use the fact that prefixing with the empty sequence is the identity: the second and third parts uses the fact that prefixing is associative, together with pasting of diagrams like (41).

Lemma 15. *For $\Gamma, \Delta \subseteq \mathfrak{L}_{\mathsf{S}_{\boldsymbol{\beta'}}}$,*

$$\Gamma \vdash \Delta$$

iff

$$\iota_{\beta',\beta}\Gamma \vdash \iota_{\beta',\beta}\Delta$$

Proof. By the interpolation theorem, we can restrict the proofs in the premise and conclusion to the languages $\mathfrak{L}_\alpha$ and $\mathfrak{L}_\beta$. Now, since $\iota_{\alpha,\beta}$ is an isomorphism of languages, it has an inverse: both it and its inverse must preserve derivability in the languages $\mathfrak{L}_\alpha$ and $\mathfrak{L}_\beta$.

We have

Proposition 25. *For $\mathsf{S}_{\boldsymbol{\alpha}} \sqsubset \mathsf{S}_{\boldsymbol{\beta}}$, $\eta_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ is left adjoint to $R_{\boldsymbol{\beta},\boldsymbol{\alpha}}$: that is, for $P \in \mathfrak{L}_{\mathsf{S}_{\boldsymbol{\alpha}}}$ and $Q \in \mathfrak{L}_{\mathsf{S}_{\boldsymbol{\beta}}}$, with $\alpha \sqsubseteq \beta$,*

$$\mathcal{D}_{una}, P \vdash R_{\boldsymbol{\beta},\boldsymbol{\alpha}}Q$$

iff

$$\mathcal{D}_{una}, \eta_{\boldsymbol{\alpha},\boldsymbol{\beta}}P \vdash Q$$

Proof.

$$\mathcal{D}_{una}, P \vdash R_{\boldsymbol{\beta},\boldsymbol{\alpha}}Q$$

iff

$$\mathcal{D}_{una}, \iota_\alpha^{-1}P \vdash \iota_\alpha^{-1}R_{\boldsymbol{\beta},\boldsymbol{\alpha}}Q \qquad \text{since } \mathcal{D}_{una} \text{ is invariant under } \iota_\alpha$$

iff

$$\mathcal{D}_{una}, \iota_\alpha^{-1}P \vdash R(\iota_{\beta',\beta}^{-1}Q) \qquad \text{by definition of } R_{\beta,\alpha}$$

iff

$$\mathcal{D}_{una}, \eta_{\beta'}\iota_\alpha^{-1}P \vdash \iota_{\beta',\beta}^{-1}Q \qquad \text{by Proposition 1, p. 15}$$

iff

$$\mathcal{D}_{una}, \iota^{-1}_{\beta',\beta}(\eta_{\alpha,\beta})P \vdash \iota^{-1}_{\beta',\beta}Q \qquad \text{by definition of } \eta_{\alpha,\beta}$$

iff

$$\mathcal{D}_{una}, \eta_{\alpha,\beta}P \vdash Q \qquad \text{by Lemma 15}$$

Now we can define our key concept.

Definition 27. Let $\mathcal{D}$ be a basic action theory. Define the *associated classical Reiter category* as follows:

1. The base is the tree of situations;
2. The fibre over a situation $\mathsf{S}_{\boldsymbol{\alpha}}$ is the partial order associated to the preorder on $\mathfrak{L}_{\mathsf{S}_{\boldsymbol{\alpha}}}$, where the ordering is induced by the entailment relation given by $\mathcal{D}_{una}$;
3. The reindexing functors are given by the regression operators $R_{\boldsymbol{\beta},\boldsymbol{\alpha}}$;
4. The left adjoints are given by the progression operators, $R_{\boldsymbol{\alpha},\boldsymbol{\beta}}$
5. The right adjoints are the de Morgan duals to the progression operators.

Finally we have

Theorem 7. *The associated Reiter category to $\mathcal{D}$ is a classical Reiter category.*

Proof. The base category is a tree, and hence has fibre products. The fibres are Boolean algebras, by construction: the reindexing morphisms, by construction, preserve both meets and joins, and are hence Boolean algebra morphisms. The Beck-Chevalley condition is trivial when the category is a tree.

References

[1] Jean Bénabou. Fibred categories and the foundations of naïve category theory. *Journal of Symbolic Logic*, 50:10–37, 1985.

[2] Edward Craig, editor. *Routledge Encyclopaedia of Philosophy*. Routledge, 1998.

[3] Donald Davidson. *Essays on Actions and Events*. Oxford University Press, 1980.

[4] Donald Davidson. Events as particulars. In *Essays on Actions and Events* [3].

[5] Donald Davidson. The individuation of events. In *Essays on Actions and Events* [3], pages 163–180. Originally published in [17, 216–34].

[6] Jennifer Hornsby. Action. In Craig [2], pages 37–41.

[7] Jennifer Hornsby. Anomalousness in action. In Lewis Edwin Hahn, editor, *The Philosophy of Donald Davidson*, volume XXVII of *The Library of Living Philosophers*, pages 623–635. Open Court, Chicago and La Salle, Illinois, 1999.

[8] B. Jacobs. *Categorical Logic and Type Theory*. Number 141 in Studies in Logic and the Foundations of Mathematics. North Holland, Amsterdam, 1999.

[9] Bart Jacobs and Jan Rutten. A tutorial on (co)algebras and (co)induction. *EATCS Bulletin*, 62:229–259, 1997.

[10] Peter T. Johnstone. *Stone Spaces*. Number 3 in Cambridge Studies in Advanced Mathematics. Cambridge University Press, 1982.

[11] John McCarthy and P.J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence*, 4:463–502, 1969.

[12] Robin Milner. *Communication and Concurrency*. Prentice-Hall, 1989.

[13] Edwin P. Pednault. *Toward a Mathematical Theory of Plan Synthesis*. PhD thesis, Department of Electrical Engineering, Stanford University, Stanford CA, 1986.

[14] Edwin P. Pednault. Synthesising plans that contain actions with context-dependent effects. *Computational Intelligence*, (4):357–372, 1988.

[15] Raymond Reiter. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In V.I. Lifschitz, editor, *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honour of John McCarthy*, pages 418–420. Academic Press, San Diego CA, 1991.

[16] Raymond Reiter. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. MIT Press, Cambridge MA, 2001.

[17] Nicholas Rescher. *Essays in Honor of Carl G. Hempel*. Reidel, Dordrecht, 1969.

[18] Jan Rutten. Universal coalgebra: A theory of systems. *Theoretical Computer Science*, (249):3–80, 2000.

[19] Davide Sangiorgi. On the bisimulation proof method. *Mathematical Structures in Computer Science*, 8(5):447–479, October 1998.

[20] A.S. Troelstra and H. Schwichtenberg. *Basic Proof Theory*. Number 43 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, Cambridge, 1996.

[21] Linda Wetzel. Type/token distinction. In Craig [2], pages 509–511.