ISSN 1470-5559

A Type Theory for to Reiter's Regression

Graham White

Queen Mary
University of London

RR-06-02

April 2006

Department of Computer Science

A Type Theory for to Reiter's Regression

Graham White
Department of Computer Science
Queen Mary, University of London
London, UK
graham@dcs.qmul.ac.uk

April 29, 2006

Abstract

We develop a type-theoretic formulation of the approach to the frame problem given by Reiter in his book *Knowledge in Action*.

1 Introduction

Reiter [8] (following the technical leads of Pednault [6, 7] and others) has described an approach to the frame problem in which the central concept is that of *regression*: this is an operation which takes a proposition, asserted of one situation, and maps it to its preconditions, asserted of the previous situation. As Reiter shows, the regression operator has good computational and technical properties: it is, in an appropriate sense, monotonic, it is computationally tractable, and it can be used directly to solve many concrete AI problems.

Conceptually appealing though Reiter's approach is, the mathematical implementation can often be a little ragged. One problem which frequently arises is this: we are working with a single, global language which has terms for situations and for fluents, and we want to be able to say that a given proposition is "about" a given situation. This is not a purely syntactic matter: criteria for this "aboutness" may be difficult, or impossible, to find (see, for example, [8, §9.1.3, p. 210]

This suggests that we should try starting the other way round: that we should use a language in which there is a primitive relation between propositions and the situations that they are about, and see if we can use *that* language to handle the frame problem in Reiter's style. This can be done, and it reveals unexpected connections. The resulting logic is a type theory: situations are types, and a proposition is *about* a particular situation iff it has that type. And, in fact, Reiter's regression operators turn out to be they key ingredient in this construction.

2 The Type Theory

2.1 Situations and Actions

we are given a set Σ of states, and, for each pair of states, a set (which may be empty) of transitions between them. If there is a transition α between states s and t, we can write it thus:

$$\begin{tikzcd} s \arrow[r, bend left, "\alpha"] & t \end{tikzcd}$$

We also assume that, given a compatible pair of transitions (that is, a pair of transitions α and β with the source of one the same as the target of the other) we can compose them, giving a transition $\beta \circ \alpha$:

$$\begin{tikzcd} s \arrow[r, "\alpha"] \arrow[rr, bend left, "\beta\circ\alpha"] & t \arrow[r, "\beta"] & u \end{tikzcd}$$

We can also assume that, for any state s, there is an identity transition $\mathrm{Id} : s \rightsquigarrow s$, which leaves everything unchanged.

Example 1. In the case of traditional AI – for example, [8, § 4.2.2] – the states and transitions form a tree: transitions are sequences of actions and, since we have a distinguished root node, states may also be identified with sequences of actions. We also have additional apparatus: actions have labels, and different transitions may have the same label – for example, we may have, as in [8, p.51],

$$\begin{tikzcd}
& & & \bullet \arrow[dll, "\alpha"'] \arrow[drr, "\beta"] & & & \\
& \bullet \arrow[dl, "\alpha"'] \arrow[dr, "\beta"] & & & & \bullet \arrow[dl, "\alpha"'] \arrow[dr, "\beta"] & \\
\bullet & & \bullet & & \bullet & & \bullet
\end{tikzcd}$$

This labelling corresponds to the the type/token distinction in the philosophy of action [1, 5].

Example 2. In the case of the theory of automata, actions again have labels: however, the states and transitions do not form a tree. We may, for example, have something like

$$\begin{tikzcd}
s \arrow[r, bend left, "\alpha"] \arrow[r, "\beta"'] & t \arrow[d, "\alpha"'] \\
& u \arrow[u, bend right, "\beta"'] \arrow[ul, "\alpha"]
\end{tikzcd}$$

The role of the labelling here is somewhat different – labels stand for observables – and the fact that transitions do not form a tree is important. A tree-like structure of actions and transitions will correspond to the *initial* model of some suitable first-order theory: the theory of automata uses, instead of initial models, final models, and, correspondingly, coinduction replaces induction [10, 9, 4]. Final models are very rarely considered in AI, with the honourable exception of Norman Foo's work. [2].

Remark 1. What we have here is a very minimal structure: it does not have to have labelled actions, and it enforces neither initial nor final models. If we add to it identity transitions (that is, a transition from each object to itself which does nothing), then we have a category: identity transitions are implicitly present in the traditional AI picture, since, according to that picture, transitions are finite sequences of actions, and identity transitions are just sequence of length zero. At any event we may assume that we have a category of states and transitions: this should not be contentious, since, as we have argued, it assumes *less* than the standard action descriptions of two well-established disciplines.

2.2 Regression and Progression

The goal of this work is to define operators analogous to Reiter's regression: that is, to assign, to each transition

$$\begin{tikzcd} s \arrow[r, bend left, "\alpha"] & t \end{tikzcd},$$

a map α^*, taking propositions of type t to propositions of type s, which satisfies the "regression theorem" [8, pp. 67f.]:

> If we have a transition α between states s and t, then, if P is a proposition at s and Q is a proposition at t, then
>
> $$P \vdash Q$$
>
> if and only if
>
> $$P \vdash \alpha^* Q$$

That is, $\alpha^* Q$ should be the weakest preconditions of Q.

Having defined regression, a successful theory ought to be able also to define *progression.* Reiter's formulation – slightly adapted to our formalism – is as follows: given a set of sentences Γ_s at a situation s, then, for transition α from s to a situation t, a set of sentences Γ_t is a *progression* of Γ_s to t iff

1. Γ_t is a set of sentences at t, and
2. Γ_s and Γ_t should be equivalent with respect to entailment of any sentences about the future of t.

2.3 The Rules

We can now describe the type theory. This will be given by a sequent calculus: sequents will be labelled by a pair of situations, s and t, and an action which yields the situation t when performed in s. We will write such a sequent like this:

$$\alpha : s \rightsquigarrow t; P_1, \ldots, P_n \vdash Q_1, \ldots, Q_m \tag{1}$$

The P_i are propositions at s, and Q_i are propositions at t. It will be valid (with respect to our transition system) iff, whenever all of the P_i are true at s, one of the Q_j is true at t. As usual, we will use Γ, Δ and so on to stand for sets of propositions: P, Q etc. will stand for propositions.

We define two operators for each transition α: there will be a regression operator, α^*, and a progression operator, $\alpha_!$. It will turn out that α^* commutes both with conjunctions and disjunctions, so that $\alpha^*\Gamma$ can be defined elementwise for sets of propositions both on the left and on the right of sequents. $\alpha_!$, on the other hand, will commute only with disjunctions, so we can apply $\alpha_!$, elementwise, to sets of propositions on the right of sequents, but not on the left: on the left it will be explicitly applied to single propositions.

We have not done as Reiter does, and explicitly defined a primitive notion of progression for sets of propositions: we assume that things are finite enough to be able to work with conjunctions instead of Reiter's sets of propositions. If not, then the required modifications are notationally tedious, but conceptually obvious.

The rules of our calculus are given in Table 1.

2.4 The Key Properties

We show now that the operators α^* and $\alpha_!$ are, in fact, regression and progression operators.

First some trivial technical lemmas.

Lemma 1. *α^* commutes with conjunctions and disjunctions; that is,*

$$\begin{array}{rcl} \alpha^*(P \wedge Q) & \dashv\vdash & \alpha^*P \wedge \alpha^*Q \\ \alpha^*(P \vee Q) & \dashv\vdash & \alpha^*P \vee \alpha^*Q \end{array}$$

Proof. We have:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\text{Id} : t \rightsquigarrow t; P, Q \vdash P}\,\text{Axiom}}{\text{Id} : s \rightsquigarrow s; P \wedge Q \vdash P}\,\wedge\text{L}}{\alpha : s \rightsquigarrow t; P \wedge Q \vdash \alpha^*P}\,\alpha^*\text{R}}{\text{Id} : s \rightsquigarrow s; \alpha^*(P \wedge Q) \vdash \alpha^*P}\,\alpha^*\text{L} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\text{Id} : t \rightsquigarrow t; P, Q \vdash Q}\,\text{Axiom}}{\text{Id} : s \rightsquigarrow s; P \wedge Q \vdash Q}\,\wedge\text{L}}{\alpha : s \rightsquigarrow t; P \wedge Q \vdash \alpha^*Q}\,\alpha^*\text{R}}{\text{Id} : s \rightsquigarrow s; \alpha^*(P \wedge Q) \vdash \alpha^*Q}\,\alpha^*\text{L}}{\text{Id} : s \rightsquigarrow s; \alpha^*(P \wedge Q) \vdash \alpha^*P \wedge \alpha^*Q}\,\wedge\text{R}$$

The proofs of the other entailments are similar. □

Remark 2. This means that it is sufficient to know α^* only on atoms and the negations of atoms: α^* can then by evaluated on general sentences by writing them in negation normal form. This is an assumption which Reiter [8, pp. 26ff] seems to make implicitly.

Table 1: The Sequent Calculus

$$\frac{}{\mathrm{Id} : s \rightsquigarrow s; \Gamma, P \vdash P, \Delta}\ \text{Axiom}$$

$$\frac{\alpha : s \rightsquigarrow t; \Gamma \vdash \Delta}{\alpha : s \rightsquigarrow t; \Gamma, P \vdash \Delta}\ \text{WL} \qquad \frac{\alpha : s \rightsquigarrow t; \Gamma \vdash \Delta}{\alpha : s \rightsquigarrow t; \Gamma \vdash P, \Delta}\ \text{WR}$$

$$\frac{\alpha : s \rightsquigarrow t; \Gamma, P, P \vdash \Delta}{\alpha : s \rightsquigarrow t; \Gamma, P \vdash \Delta}\ \text{CL} \qquad \frac{\alpha : s \rightsquigarrow t; \Gamma \vdash P, P, \Delta}{\alpha : s \rightsquigarrow t; \Gamma \vdash P, \Delta}\ \text{CR}$$

$$\frac{}{\alpha : s \rightsquigarrow t; \Gamma, \bot \vdash \Delta}\ \bot\,\text{L} \qquad \frac{}{\alpha : s \rightsquigarrow t; \Gamma \vdash \top, \Delta}\ \top\text{R}$$

$$\frac{\mathrm{Id} : s \rightsquigarrow s; \Gamma \vdash P, \alpha^*\Delta}{\alpha : s \rightsquigarrow t; \Gamma, \neg P \vdash \Delta}\ \neg\text{L} \qquad \frac{\alpha : s \rightsquigarrow t; \Gamma, \alpha^* P \vdash \Delta}{\alpha : s \rightsquigarrow t; \Gamma \vdash \neg P, \Delta}\ \neg\text{R}$$

$$\frac{\alpha : s \rightsquigarrow t; \Gamma, P_1 \vdash \Delta \quad \alpha : s \rightsquigarrow t; \Gamma, P_2 \vdash \Delta}{\alpha : s \rightsquigarrow t; \Gamma, P_1 \vee P_2 \vdash \Delta}\ \vee\text{L} \qquad \frac{\alpha : s \rightsquigarrow t; \Gamma \vdash Q_1, Q_2\Delta}{\alpha : s \rightsquigarrow t; \Gamma \vdash Q_1 \vee Q_2, \Delta}\ \vee\text{R}$$

$$\frac{\alpha : s \rightsquigarrow t; \Gamma, P_1, P_2 \vdash \Delta}{\alpha : s \rightsquigarrow t; \Gamma, P_1 \wedge P_2 \vdash \Delta}\ \wedge\text{L} \qquad \frac{\alpha : s \rightsquigarrow t; \Gamma \vdash Q_1, \Delta \quad \alpha : s \rightsquigarrow t; \Gamma \vdash Q_2, \Delta}{\alpha : s \rightsquigarrow t; \Gamma \vdash Q_1 \wedge Q_2, \Delta}\ \wedge\text{R}$$

$$\frac{\alpha : s \rightsquigarrow t; \Gamma, P_2 \vdash \Delta \quad \mathrm{Id} : s \rightsquigarrow s; \Gamma \vdash P_1, \alpha^*\Delta}{\alpha : s \rightsquigarrow t; \Gamma, P_1 \rightarrow P_2 \vdash \Delta}\ \rightarrow\text{L} \qquad \frac{\alpha : s \rightsquigarrow t; \Gamma, \alpha^* Q_1 \vdash Q_2, \Delta}{\alpha : s \rightsquigarrow t; \Gamma \vdash Q_1 \rightarrow Q_2, \Delta}\ \rightarrow\text{R}$$

$$\frac{\alpha : t \rightsquigarrow u; \Gamma \vdash \Delta}{\alpha \circ \beta : s \rightsquigarrow u; \beta^*\Gamma \vdash \Delta}\ \alpha^*\text{L} \qquad \frac{\alpha \circ \beta : s \rightsquigarrow u; \Gamma \vdash \Delta}{\beta : s \rightsquigarrow t; \Gamma \vdash \alpha^*\Delta}\ \alpha^*\text{R}$$

$$\frac{\mathrm{Id} : s \rightsquigarrow s; P \vdash \alpha^*\Delta}{\mathrm{Id} : t \rightsquigarrow t; \alpha_! P \vdash \Delta}\ \alpha_!\text{L} \qquad \frac{\beta : s \rightsquigarrow t; \Gamma \vdash \Delta}{\alpha \circ \beta : s \rightsquigarrow u; \Gamma \vdash \alpha_!\Delta}$$

$$\frac{\beta : s \rightsquigarrow t; \Gamma \vdash Q \quad \alpha : t \rightsquigarrow u; \Gamma', Q \vdash \Delta'}{\alpha \circ \beta : s \rightsquigarrow u; \Gamma, \beta^*\Gamma' \vdash \Delta'}\ \text{cut}$$

Lemma 2. *$\alpha_!$ commutes with disjunctions: that is,*

$$\alpha_!(P \vee Q) \dashv\vdash \alpha_! P \vee \alpha_! Q.$$

Proof.

$$\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots\\ \mathrm{Id} : s \rightsquigarrow s; P \vee Q \vdash P, Q\end{array}}{\alpha : s \rightsquigarrow t; P \vee Q \vdash \alpha_! P, \alpha_! Q}\;\alpha_!\mathrm{R}}{\mathrm{Id} : s \rightsquigarrow s; P \vee Q \vdash \alpha^* \alpha_! P, \alpha^* \alpha_! Q}\;\alpha^*\mathrm{R}}{\mathrm{Id} : t \rightsquigarrow t; \alpha_!(P \vee Q) \vdash \alpha_! P, \alpha_! Q.}\;\alpha_!\mathrm{L}}{\mathrm{Id} : t \rightsquigarrow t; \alpha_!(P \vee Q) \vdash \alpha_! P \vee \alpha_! Q}\;\vee\mathrm{R}$$

and

$$\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots\\ \mathrm{Id} : s \rightsquigarrow s; P \vdash P \vee Q\end{array}}{\alpha : s \rightsquigarrow t; P \vdash \alpha_!(P \vee Q)}}{\mathrm{Id} : s \rightsquigarrow s; P \vdash \alpha^* \alpha_!(P \vee Q)}\;\alpha^*\mathrm{R}}{\mathrm{Id} : t \rightsquigarrow t; \alpha_! P \vdash \alpha_!(P \vee Q)}\;\alpha_!\mathrm{L} \qquad \begin{array}{c}\vdots\\ \mathrm{Id} : t \rightsquigarrow t; \alpha_! Q \vdash \alpha_!(P \vee Q)\end{array}}{\mathrm{Id} : t \rightsquigarrow t; \alpha_! P \vee \alpha_! Q \vdash \alpha_!(P \vee Q)}\;\vee\mathrm{L}$$

□

Now the crucial results:

Proposition 1. *α^* is a regression operator: that is,*

$$\alpha \circ \beta : s \rightsquigarrow u; P \vdash Q \tag{2}$$

holds iff

$$\beta : s \rightsquigarrow t; P \vdash \alpha^* Q \tag{3}$$

does.

Proof. Suppose that we have (2); then we can derive (3) by an application of α^*R. Conversely, suppose that we have a proof of (3). Note that we have

$$\dfrac{\dfrac{}{\mathrm{Id} : u \rightsquigarrow u; Q \vdash Q}\;\mathrm{Axiom}}{\alpha : t \rightsquigarrow u; \alpha^* Q \vdash Q}\;\alpha^*\mathrm{L}$$

is valid, so we can simply apply the cut rule and derive (2). □

Remark 3. This generalises Reiter's definition of a regression operator, which restricts to the case where β is the identity transition.

Proposition 2. $\alpha_!$ *is a progression operator; that is, we have*

$$\alpha : t \rightsquigarrow u; \beta_! P \vdash Q \tag{4}$$

iff we have

$$\alpha \circ \beta : s \rightsquigarrow u; P \vdash Q \tag{5}$$

Proof. Suppose first that we have a proof Π of (5): we derive a proof of (4) as follows.

$$\dfrac{\dfrac{\begin{array}{c}\Pi\\ \vdots\\ \alpha \circ \beta : s \rightsquigarrow u; P \vdash Q\end{array}}{\beta : s \rightsquigarrow t; P \vdash \alpha^* Q}}{\alpha : t \rightsquigarrow u; \beta_! P \vdash Q}\;\beta_!\mathrm{L}$$

Conversely, if we have a proof of (4), we can cut with the conclusion of

$$\dfrac{\dfrac{}{\mathrm{Id} : s \rightsquigarrow s; P \vdash P}\;\mathrm{Axiom}}{\beta : s \rightsquigarrow t; P \vdash \beta_! P}\;\beta_!\mathrm{R}$$

thus obtaining (5). □

Remark 4. We could also obtain the above result by first deriving the equivalence

$$\begin{array}{c} P \vdash \alpha^* Q \\ \hline\hline \alpha_! P \vdash Q \end{array}$$

and then deducing Proposition 2 from Proposition 1.

2.5 Making Regression Explicit

The above system is interesting, but does not really allow us to do any of the calculation possible in Reiter's system: it is valid for *any* system of regression operators. It shows, as it were, the properties that regression operators in general must have.

However, we are working with Reiter's system, and for this we have quite concrete descriptions of the regression operators: they are as follows. Since the regression operators commute with conjunctions and disjunctions, we may delay cashing out the operators until we have reduced the formulae to positive or negative atoms: in any case, Reiter only defines them for atoms, and we are working with his definitions.

Reiter's regression operator is defined as follows. Suppose that we are given, for each action α, an set of what are called *effect axioms* [8, pp. 25ff.]:

before	**after**
ϕ_1	ψ_1
$\vdots$	$\vdots$
ϕ_k	ψ_k

where the ϕ_i, ψ_j in $\mathfrak{L}$, and where the ψs are atoms or the negations of atoms. Reiter [8, pp. 26ff] defines, for each action $s \xrightarrow{\alpha} t$ and atom $F \in \mathfrak{L}$, a *successor state axiom* of the form

$$F \text{ is true at } t \text{ iff } \Phi(F, \alpha) \text{ is true at } s, \tag{6}$$

where $\Phi(F, \alpha)$ can be defined purely syntactically from the successor state axioms for α: the definition does not depend on the Ξ_s.

Consider the successor state axioms: they are, as we have described, of the form "if ϕ holds at s, then ψ holds at t", where we are assuming that there is a single-stage transition α between s and t. In these axioms, the ψs are always (positive or negative) atoms: so we can, for each single-stage action α and each relational symbol F in the language, collect the antecedents of the corresponding successor state axioms together into two formulae: $\gamma^+(\alpha, F)$ will be the preconditions at s for F to become true at t as a result of α, whereas $\gamma^-(\alpha, F)$ will be the preconditions at s for F to become false at t as a result of α.

Now F will be true at t if either it becomes true as a result of α, or if it is true at s and not forced to become false as a result of α. Consequently,

$$\alpha^* F = \gamma^+(\alpha, F) \vee (F \wedge \neg\gamma^-(\alpha, F)) \tag{7}$$

and, similarly,

$$\alpha^* \neg F = \gamma^-(\alpha, F) \vee (\neg F \wedge \neg\gamma^+(\alpha, F)) \tag{8}$$

So we may write the following two rules for regression. Suppose that we have composable actions α and β:

$$\begin{tikzcd} s \arrow[r, "\beta"] \arrow[rr, bend left, "\alpha\circ\beta"] & t \arrow[r, "\alpha"] & u \end{tikzcd}$$

Suppose also that α is atomic. Then we have the following pair of right rules:

$$\frac{\beta : s \rightsquigarrow t; \Gamma \vdash \gamma^+(\alpha, F)}{\beta : s \rightsquigarrow t; \Gamma \vdash \alpha^* F}\ \alpha^*\text{R1}$$

$$\frac{\beta : s \rightsquigarrow t; P_1, \Gamma \vdash F \wedge \neg\gamma^-(\alpha, F)}{\beta : s \rightsquigarrow u; \Gamma \vdash \alpha^* F}\ \alpha^*\text{R2}$$

There are, of course, corresponding rules with the signs reversed for dealing with $\neg F$.

There is also a left rule. For simplicity, we give in the case where we have just one formula on the left: corresponding, but more complex, rules can be given for more general antecedents. Again we assume that we have composable actions

$$\begin{tikzcd} s \arrow[r, "\beta"] \arrow[rr, bend left, "\alpha\circ\beta"] & t \arrow[r, "\alpha"] & u \end{tikzcd}$$

but, this time, that β is atomic.

$$\frac{\alpha \circ \beta : s \rightsquigarrow u; \gamma^{+}(\beta, F) \vdash Q \quad \alpha \circ \beta : s \rightsquigarrow u; F, \neg\gamma^{-}(\beta, F) \vdash Q}{\alpha \circ \beta : s \rightsquigarrow u; \alpha^{*} F \vdash Q}\ \beta^{*}\mathrm{L}$$

and, of course, there will be a corresponding rule with the signs reversed for $\neg F$.

2.5.1 The Initial State

It only remains for us to show how we specify the initial state. If we have a collection of axioms – say $\{\eta_i\}_{i \in I}$ – which are supposed to hold in the initial state, then we can represent them in the calculus as

$$\frac{}{\mathrm{Id} : S_0 \rightsquigarrow S_0; \vdash \eta_i}\ \text{Axiom}$$

for each $i \in I$.

3 Semantics

At this stage, one might justifiably ask for a semantics; we need, after all, to show that these constructions are non-vacuous. A suitable semantics will be given by the following collection of objects.

1. a category of states and transitions: this will be called the *base category*, and will be the same as the category of states and transitions which we have been using for labelling our sequents with.

2. for each object s of the base category, a boolean algebra A_s: this algebra will be called the *propositions at s*

3. for each morphism $\alpha : s \rightsquigarrow t$ of the base category, a lattice homomorphism

$$\alpha^{*} : A_t \to A_s,$$

4. for each morphism $\alpha : s \rightsquigarrow t$ of the base category, an $\vee$-semilattice homomorphism

$$\alpha_{!} : A_s \to A_t$$

left adjoint to α^{*} – that is, such that

$$\alpha_{!} P \leq Q$$

in A_t iff

$$P \leq \alpha^{*} Q$$

in A_t.

Interpretations will be given in the usual way: for each object s of our transition system, we give a boolean algebra homomorphism from the propositions at s to A_s. If we are specifying regression explicitly, *a la* Reiter, we would also stipulate that the regression operators had the desired effects. An entailment will then be valid in a model if, for all interpretations, whenever all elements of the antecedent are $\top$ in A_s, then at least one element of the consequent is $\top$ in A_t.

We can, then, prove soundness and completeness. Soundness is, as usual, easy but tedious. Completeness needs more work, and also needs assumptions on the base category: if we are working in the standard AI case, where the base category is a *tree*, then we can do this by first proving cut elimination for the type theory, and then defining a canonical model by a Lindebaum construction: elements of A_s will be equivalence classes of propositions at s under the equivalence relation generated by

$$\mathrm{Id} : s \rightsquigarrow s; P \vdash Q.$$

We need cut elimination to show that the Boolean algebras thus generated are non-trivial.

3.1 The Basis for Our Semantics

The semantics that we have given uses a particular mathematical structure: it is what is known as a *fibred category.* Intuitively, this can be regarded as a family of categories, parametrised by another category: here the parametrising category – what is usually referred to as the *base category* – is the category of states and transitions, whereas the parametrised categories – what are called the *fibres* – are the boolean algebras A_s.

One of the crucial points in the definition of a fibred category is this: that we must ensure that isomorphic objects in the base correspond to equivalent fibre categories. More generally, we must ensure that, if there is a morphism in the base, the corresponding fibres are related by a suitable functor: this is usually called a *pullback*, but it is also what we described as a *regression operator.* So, in terms of fibred categories, Reiter's operators are not something that has to be added: they are a basic part of the definition.

3.1.1 The System

Now there is not much point in describing mathematical apparatus unless it has a use, and fibred categories have an important use: they give the semantics of type theories. Types correspond to objects in the parametrising category: propositions of those types correspond to objects in the parametrised categories: and morphisms are entailments.

Remark 5. Note for the cognoscenti: this corresponds to the propositions-as-objects rather than the propositions-as-types paradigm [3, p. 138].

So, corresponding to our fibred category, we have the type theory that we have just described: and the process of discovery was – as often occurs – opposite to the process of exposition: the fibred category was discovered first, and then it was made concrete by translating it into type-theoretic terms.

4 Conclusion

We have shown that Reiter's treatment of regression has hidden category-theoretic structure of a well-known sort. However, the story does not end there: we can, on the basis of the category theory, construct a type theory, and this type theory can be used to perform the sort of regression arguments that Reiter uses. From this perspective, Reiter's insights are, in fact, similar to techniques used in other research communities – the main similarities are with Grothendieck's work on fibred categories, and with Lawvere's work on the semantics of the quantifiers. What is surprising is that these other approaches have quite different motivations: Grothendieck was working on geometry, whereas Lawvere was working on quantification; neither of these contexts is directly comparable with what Reiter was doing. Many problems in artificial intelligence thus seem to be closer to mainstream mathematics than either AI practitioners, or mathematicians, realise.

References

[1] Donald Davidson. The individuation of events. In *Essays on Actions and Events*, pages 163–180. Oxford University Press, 1980. Originally published in [**?**, 216–34].

[2] Norman Y. Foo, Abhaya Nayak, Maurice Pagnucco, and Dongmo Zhang. State minimisation revisited. In Markus Stumptner, Dan Corbett, and Michael J. Brooks, editors, *Australian Joint Conference on Artificial Intelligence*, volume 2256 of *Lecture Notes in Computer Science*, pages 153–164. Springer, 2001.

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

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

[5] Jaegwon Kim. *Supervience and Mind: Selected Philosophical Essays.* Cambridge Studies in Philosophy. Cambridge University Press, Cambridge, 1993.

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

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

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

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

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