New Foundations for Fixpoint Computations: FIX-Hyperdoctrines and the FIX-Logic

Roy L. Crole* Andrew M. Pitts†

Computer Laboratory,
University of Cambridge,
Cambridge CB2 3QG,
England.

*Research supported by a SERC Research Studentship.
†Research supported by the CLICS project (EEC ESPRIT BR nr 3003).

Abstract

This paper introduces a new higher-order typed constructive predicate logic for fixpoint computations, which exploits the categorical semantics of computations introduced by Moggi [Mog 89] and contains a version of Martin Löf's 'iteration type' [MarL 83]. The type system enforces a separation of computations from values. The logic contains a novel form of fixpoint induction and can express partial and total correctness statements about evaluation of computations to values. The constructive nature of the logic is witnessed by strong metalogical properties which are proved using a category-theoretic version of the 'logical relations' method [Plo 85].

1 Introduction

It is well known that primitive recursion at higher types can be given a categorical characterisation in terms of Lawvere's concept of *natural number object* [LamSco 86]. We show that a similar characterisation can be given for general recursion via fixpoint operators of higher types, in terms of a new concept—that of a *fixpoint object* in a suitably structured category. This notion was partly inspired by contemplation of Martin-Löf's non-standard 'iteration type' in his domain theoretic interpretation of type theory [MarL 83]. However, the key ingredient which allows the formulation of the concept of fixpoint object is the treatment of computations using *monads* introduced by Moggi [Mog 89], where there is a distinction between the elements of a type α and computations of elements of that type—the latter being grouped into a new type $T\alpha$. Moggi's computational metalanguage $\lambda\mathrm{ML}_T$ [Mog1 91], contains the following formation rules:

$$\frac{\alpha \text{ type}}{T\alpha \text{ type}}$$

$$\frac{M\colon\alpha}{\mathsf{Val}(M)\colon T\alpha}$$

$$\frac{E\colon T\alpha \qquad \begin{array}{c}[x\colon\alpha]\\ F(x)\colon T\beta\end{array}}{\mathsf{Let}\,(E,F)\colon T\beta}$$

Remark 1.1 These rules, and the others which appear in this paper, are presented in natural deduction style. In this section, we present some rules with discharged hypotheses in square brackets. In later sections, rules will be written using intuitionistic sequents in context, where contexts are lists of typed variables. Since there are several unfamiliar variable binding operations in the syntax, we will also adopt Martin-Löf's theory of expressions and arities. For us this will be a $\alpha\beta\eta$-lambda calculus over ground types TYPE, TERM and PROP, with abstraction denoted $x.e$, application denoted $f(e)$, substitution denoted $e[e'/x]$ and a multiple application such as $(f(e))(e')$ abbreviated to $f(e,e')$; see [NorPetSmi 90], for example. This system will be referred to as the *meta-λ-calculus*. Finally, it should be noted that our syntax is a slight variant of Moggi's.

Intuitively, $\mathsf{Val}(M)$ is the value M regarded as a trivial computation which immediately evaluates to itself; and $\mathsf{Let}\,(E, F)$ denotes the computation which firstly tries to evaluate E to some value $M\colon\alpha$ and then proceeds to evaluate $F(M)$. These intended meanings are captured by three equational axioms:

$$\begin{array}{rcl} \mathsf{Let}\,(\mathsf{Val}(M), F) & = & F(M) \\ \mathsf{Let}\,(E, x.\mathsf{Val}(x)) & = & E \\ \mathsf{Let}\,(\mathsf{Let}\,(E, F), G) & = & \mathsf{Let}\,(E, x.\mathsf{Let}\,(F(x), G)). \end{array}$$

In addition, $\lambda\mathrm{ML}_T$ extends the simply typed lambda calculus: there are function types $\alpha \Rightarrow \beta$ with lambda abstractions $\lambda x\colon\alpha.F(x)$ and applications MN satisfying the usual β and η equalities. The system also contains product types $\alpha\times\beta$ with (surjective) pairing $\langle M, N\rangle$ and projections $\mathsf{Fst}(M)$, $\mathsf{Snd}(M)$; and it contains a type *unit* with unique element $\langle\rangle\colon \mathit{unit}$.

The categorical counterpart of this basic formal system is the notion of a 'cartesian closed category equipped with a strong monad T' [Mog 89, section 2]. We shall refer to such structures as *let cartesian closed categories*, or just *let-ccc's*.

Definition 1.2 A *let-ccc*, C, is a cartesian closed category which enjoys the following properties:

- For each object A in C, there is an object TA,
- for each object A in C, there is a morphism $\eta_A\colon A \to TA$, and
- for each morphism $f : A \times B \to TC$, a morphism

$$\mathit{lift}(f)\colon A \times TB \to TC,$$

such that the following conditions are satisfied:

1. Given $f\colon A \to A'$ and $g\colon A' \times B \to TC$, then

$$\mathit{lift}(g \circ (f \times id_B)) = \mathit{lift}(g) \circ (f \times id_{TB}).$$

2. Given $f\colon A \times B \to TC$, then $\mathit{lift}(f) \circ (id_A \times \eta_B) = f$.
3. $\mathit{lift}(\eta_B \circ \pi_2) = \pi_2\colon A \times TB \to TB$.

4. Given $f\colon A \times B \to TC$, $g\colon A \times C \to TD$, then

$$\mathit{lift}(\mathit{lift}(g) \circ \langle \pi_1, f \rangle) = \mathit{lift}(g) \circ \langle \pi_1, \mathit{lift}(f) \rangle.$$

The above structure is equivalent to specifying a monad (T, η, μ) in the usual sense (see [MacLane 71, Chapter VI]) together with a *strength*, $\alpha_{A,B} : A \times TB \to T(A \times B)$ (namely $\mathit{lift}(\eta_{A \times B})$). It is possible to give a presentation of this variety of category in terms of categorical combinators, extending Curien's ccc combinators for the simply typed lambda calculus [Cur 86]. We will not make direct use of such combinators here, but refer the interested reader to [CroPit 90].

We complete this introduction by discussing the contents of the rest of the paper. In Section 2 we introduce the so called fixpoint type, together with some examples. We describe informally an extension of the system $\lambda\mathrm{ML}_T$ associated with the fixpoint type, and also how the fixpoint type gives rise to the notion of a fixpoint object in a suitably structured category. We describe how the formal system may be used to give denotations to recursively defined programs. In Section 3, we embed the computational lambda calculus, now enriched with fixpoint types and terms, natural numbers and finite coproducts, in a fragment of an intuitionistic predicate calculus with equality. This new logic is tailored for reasoning about evaluations of computations to values, and within it one is able to express certain partial and total correctness statements. We end the section by stating versions of the existence and disjunction properties for full intuitionistic predicate calculus which are adapted to our logic, and formalise the standardness of the natural numbers. In Section 4, we give a categorical semantics for the logical system, and prove the usual categorical-logic correspondence. Finally, in Section 5, we present a particular model of our logical system, and use its internal logic to prove the theorems from Section 3.

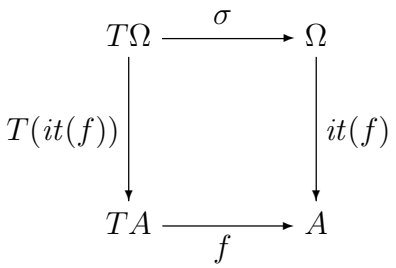
2 The fixpoint type

We begin by discussing the categorical notion of a fixpoint object.

Definition 2.1 In a let-ccc, a *fixpoint object*, is specified by the following data:

- An initial algebra $\sigma : T\Omega \to \Omega$ for the functor T. Thus for any $f : TA \to A$ there is a unique morphism $\mathit{it}(f) : \Omega \to A$ satisfying the

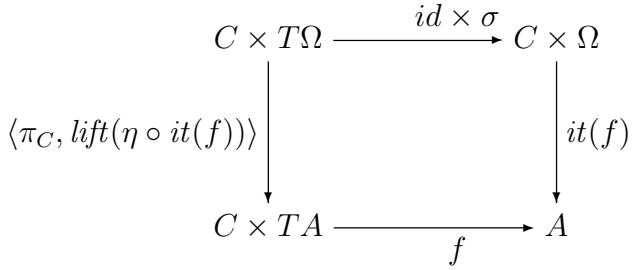
commutative diagram:



- A global element $\omega : 1 \to T\Omega$ which is the equaliser of $\eta\sigma$ and the identity on $T\Omega$. In other words ω is the unique fixed point of $\eta_\Omega\sigma : T\Omega \to T\Omega$; for any $f : A \to T\Omega$, $f = \eta_\Omega\sigma f$ if and only if $f = \omega\,!$ (where $!: A \to 1$ is the unique morphism from A to the terminal object 1).

The usual category-theoretic considerations imply that the structure Ω, σ, ω is determined uniquely up to isomorphism, within the given let-ccc, by the above properties. One should also note that σ, being the structure morphism for the initial algebra of an endofunctor, is itself an isomorphism. A fixpoint object has some characteristics which are reminiscent of a natural numbers object. In particular, if one simply has a category with finite products and a strong monad, the definition of fixpoint object should be strengthened to a parametrised form. This leads to the following lemma:

Lemma 2.2 In a let-ccc, C, the above definition of a fixpoint object is equivalent to the following: There is a morphism $\sigma: T\Omega \to \Omega$, such that given objects A and C in C, and a morphism $f: C \times TA \to A$, then there is a unique morphism $it(f): C \times \Omega \to A$ such that the following diagram commutes:



In addition, there is also a global element $\omega: 1 \to T\Omega$ with the same properties as in Definition 2.1

The basic domain-theoretic example of such a let-ccc with fixpoint object is the category of *predomains*, $\omega\mathcal{C}po$, whose objects are posets possessing joins of countably infinite chains, and whose morphisms are Scott-continuous functions, i.e. monotonic functions preserving joins of countably infinite chains. The objects of $\omega\mathcal{C}po$ are not required to possess a least element; we will refer to them as $\omega cpos$ in this paper. The operation of adjoining a least element to an ωcpo D to give the lifted ωcpo $D_\perp = \{[d] \mid d \in D\} \cup \{\perp\}$ gives a strong monad on $\omega\mathcal{C}po$, called the *lift monad*. There is a fixpoint object in $\omega\mathcal{C}po$ for the lift monad, namely the ωcpo

$$\Omega = \{0 \sqsubset 1 \sqsubset \ldots \sqsubset \top\},$$

with $\sigma : \Omega_\perp \to \Omega$ the continuous function sending $\perp$ to 0, $[n]$ to $n+1$ and $[\top]$ to $\top$; and with $\omega = [\top] \in \Omega_\perp$.

Some other monads on $\omega\mathcal{C}po$ that Moggi [Mog1 91] points out as arising in denotational semantics also possess fixpoint objects. For example the exceptions monad $T(D) = (D + E)_\perp$ (with E some fixed discrete ωcpo of exceptions) and the side-effects monad $T(D) = S \Rightarrow (D \times S)_\perp$ (with S some fixed discrete ωcpo of states) both possess fixpoint objects. This follows from the general theory of solving recursive domain equations in the 'O-category' setting of Wand and Smyth-Plotkin [SmyPlo 82]. For suppose that T is a strong monad on $\omega\mathcal{C}po$ that is locally Scott-continuous (i.e. the action of T on hom ωcpos is continuous) and that maps ωcpos to *pointed* ωcpos (i.e. ωcpos with least elements). To obtain a fixpoint object for such a T, one constructs the initial fixed object for T in the category of pointed ωcpos and embedding-projection pairs by iterating T starting at the one element ωcpo, yielding an isomorphism $\sigma : T(\Omega) \cong \Omega$. Then (Ω, σ) is an initial algebra for $T : \omega\mathcal{C}po \to \omega\mathcal{C}po$, and dually (Ω, σ^{-1}) is a final coalgebra for that functor. The initial algebra property gives us the the first part of the definition of fixpoint object; and Freyd [Freyd 9?] has observed that the second part of Definition 2.1 is implied by the coalgebra property. We record this latter observation as a lemma.

Lemma 2.3 [Freyd] Given a let-ccc, suppose that $\sigma : T\Omega \to \Omega$ is an initial algebra for the functor T (so that in particular, σ is an isomorphism). Suppose further that $\sigma^{-1} : \Omega \to T\Omega$ is a final coalgebra for T. Then there is a global element $\omega : 1 \to T\Omega$ making Ω, σ, ω a fixpoint object for T.

Proof The final coalgebra property means that for any $g : A \to TA$ there is a unique morphism $\check{g} : A \to \Omega$ satisfying $\sigma^{-1}\check{g} = T(\check{g})g$.

Define $\omega : 1 \to T\Omega$ to be $\sigma^{-1}\check{\eta}_1$. From the defining property of $\check{\eta}_1$ and the naturality of η we get

$$\omega = \sigma^{-1}\check{\eta}_1 = T(\check{\eta}_1)\eta_1 = \eta_\Omega\check{\eta}_1 = (\eta_\Omega\sigma)\omega$$

If $f : A \to T\Omega$ is any other morphism satisfying $f = (\eta_\Omega\sigma)f$, we have to see that $f = \omega\,!$. But from $f = (\eta_\Omega\sigma)f$ and the naturality of η one has

$$\sigma^{-1}(\sigma f) = f = \eta_\Omega(\sigma f) = T(\sigma f)\eta_A$$

Hence by the uniqueness part of the coalgebra property, $\sigma f = \check{\eta_A}$ and thus $f = \sigma^{-1}\eta_A$. The same argument applies equally well with $\omega!$ for f. Therefore $f = \sigma^{-1}\eta_A = \omega\,!$. □

Using the correspondence between equational $\lambda\mathrm{ML}_T$-theories and let-ccc's, one can translate the definition of a fixpoint object into a corresponding extension of the system $\lambda\mathrm{ML}_T$. This entails adding a new type *fix*, together with certain term-forming and equality rules, namely:

$$\frac{}{\omega \in T\mathit{fix}} \qquad \frac{E \in T\mathit{fix}}{\sigma(E) \in \mathit{fix}} \qquad \frac{\begin{array}{c}[x \in T\alpha] \\ F(x) \in \alpha \quad N \in \mathit{fix}\end{array}}{\mathsf{It}_\alpha(F, N) \in \alpha}$$

$$\frac{}{\omega = \mathsf{Val}(\sigma(\omega))} \qquad \frac{E = \mathsf{Val}(\sigma(E))}{E = \omega}$$

$$\frac{\begin{array}{c}[x \in T\alpha] \\ F(x) \in \alpha \quad E \in T\mathit{fix}\end{array}}{\mathsf{It}_\alpha(F, \sigma(E)) = F(\mathsf{Let}\,(E, n.\mathsf{Val}(\mathsf{It}_\alpha(F, n))))}$$

$$\frac{\begin{array}{ccc}[x \in T\alpha] & [n \in \mathit{fix}] & [e \in T\mathit{fix}] \\ F(x) \in \alpha & G(n) \in \alpha & G(\sigma(e)) = F(\mathsf{Let}\,(e, n.\mathsf{Val}(G(n)))) \quad N \in \mathit{fix}\end{array}}{G(N) = \mathsf{It}_\alpha(F, N)}$$

(The final rule, expressing the uniqueness of $\mathsf{It}_\alpha(F)$, will be subsumed in the next section in an induction rule for the fixpoint type.)

Fixpoint objects are so called because they enable one to define fixpoint terms at all types of the form $\alpha \Rightarrow T\beta$. If one views the denotation of a program of type β with input data of type α as a term of type $\alpha \Rightarrow T\beta$, then we have a method for interpreting all recursively defined programs. Indeed, we have

Proposition 2.4 (Definability of fixpoint combinators) In the presence of a fixpoint object, one may define a term

$$Y_{\alpha,\beta}\colon ((\alpha \Rightarrow T\beta) \Rightarrow \alpha \Rightarrow T\beta) \Rightarrow \alpha \Rightarrow T\beta$$

which satisfies $Y_{\alpha,\beta}\ F = F\ Y_{\alpha,\beta}\ F$ for all $F\colon (\alpha \Rightarrow T\beta) \Rightarrow \alpha \Rightarrow T\beta$. Indeed, defining

$$Y_{\alpha,\beta} = \lambda F\colon (\alpha \Rightarrow T\beta) \Rightarrow \alpha \Rightarrow T\beta.\mathsf{It}_{\alpha \Rightarrow T\beta}(\hat{F}, \sigma(\omega))$$

where $\hat{F}$ is $e.(\lambda_\alpha(\mathsf{Let}\,(e, f.Ffx)))$, then

$$\frac{\begin{array}{c}[f\colon \alpha \Rightarrow T\beta, x\colon \alpha] \\ Ffx\colon T\beta\end{array}}{Y_{\alpha,\beta}F\colon \alpha \Rightarrow T\beta}$$

and

$$\frac{\begin{array}{cc}[f\colon \alpha \Rightarrow T\beta, x\colon \alpha] & \\ Ffx\colon T\beta & M\colon \alpha\end{array}}{Y_{\alpha,\beta}F\ M = F(Y_{\alpha,\beta}F)M}$$

are derived rules.

Proof It is easy to see that the first rule is derivable. For the second rule we have

$$\begin{array}{rcl} YFM & = & \mathsf{It}(\hat{F}, \sigma(\omega))M \\ & = & \hat{F}(\mathsf{Let}\,(\omega, n.\mathsf{Val}(\mathsf{It}(\hat{F}, n))))M \\ & = & \mathsf{Let}\,(\mathsf{Let}\,(\omega, n.\mathsf{Val}(\mathsf{It}(\hat{F}, n)))f.FfM) \\ & = & \mathsf{Let}\,(\mathsf{Let}\,(\mathsf{Val}(\sigma(\omega)), n.\mathsf{Val}(\mathsf{It}(\hat{F}, n)))f.FfM) \\ & = & F\mathsf{It}(\hat{F}, \sigma(\omega))M \\ & = & F(YF)M. \end{array}$$

□

The definition of an initial algebra $\sigma : T\Omega \to \Omega$ for a functor T contains both an existence and a uniqueness part. The uniqueness part leads to the *initial T-algebra induction principle* [LehSmy 81, section 5.2], namely to show that a subobject $i : S \hookrightarrow \Omega$ is the whole of Ω, it suffices to show that the composition $\sigma T(i) : TS \to \Omega$ factors through $i : S \hookrightarrow \Omega$.

When the functor T is $(\_) + 1$ on the category of sets, the initial algebra is the natural numbers and the initial T-algebra induction principle is equivalent to the usual principle of mathematical induction. What about when the functor is lifting on $\omega\mathcal{C}po$? Restricting attention to subobjects of domains which are specified by *inclusive* subsets (those subsets of a ωcpo which are closed under taking joins of countable chains), we can use the fact that whenever $i : S \hookrightarrow \Omega$ is an inclusive subset of the ωcpo

$$\Omega = \{0 \sqsubset 1 \sqsubset \ldots \sqsubset \top\},$$

then $(i)_\perp : S_\perp \to \Omega_\perp$ is just the inclusive subset of $\Omega_\perp$ given by

$$\{e \in \Omega_\perp \mid \forall n \in \Omega.[n] = e \supset n \in S\}.$$

Then the initiality property of Ω yields the following form of the induction principle, with $S \subseteq \Omega$ inclusive:

$$\frac{\forall e \in \Omega_\perp.(\forall n \in \Omega.[n] = e \supset n \in S) \supset \sigma(e) \in S}{\forall n \in \Omega.n \in S}$$

Just as least fixed points are definable using the universal property of the initial $(\_)_\perp$-algebra Ω, so is Scott's induction principle for least fixed points [Sco 69] derivable from the above rule.

3 The FIX logical system

How may we enrich the system which we were discussing in Section 2? One obvious approach is to add a fixpoint type, coproduct types $\alpha + \beta$ and a natural number type *nat* to the system $\lambda\mathrm{ML}_T$; we refer to this extension as $\mathrm{FIX}_=$. Then we would arrive at a system which extends Gödel's system T [Gir 89, Chapter 7] but which also admits sound translations of Plotkin's PCF [Plo 77] (with either a call-by-value or a call-by-name operational semantics [Mog 88, section 5]). It is essentially the logical system $\mathrm{FIX}_=$ with which we shall concern ourselves for the rest of the paper. However, we are aiming for a constructive logic which enables us to reason directly about evaluations of computations to values: the logic $\mathrm{FIX}_=$ only captures certain computational intuitions indirectly, by containing equations which model the most basic properties one would expect computations to obey. This does not provide a system which allows on-the-nose reasoning about evaluations of

computations, and we achieve this aim by embedding $\mathrm{FIX}_{=}$ in a fragment of a first order (intuitionistic) predicate calculus with equality. An *intuitionistic* system is defined, rather than a *classical* system, as such a logic captures more closely the behavioural properties of computations. For example, the proposition $\neg\forall(\Phi)$ is classically identified with $\exists(\neg(\Phi))$; however, if we know that a certain proposition $\Phi(x)$ is not always true this does not imply that we can *calculate* when it is not true. Within this new predicate calculus, there are forms of proposition directly tailored for expressing properties of evaluations of computations.

The fragment we consider has conjunction and universal quantification (over elements of a given type), together with certain predicate constructors which implicitly contain forms of implication, disjunction and existential quantification. In order to set up a formal system for our logic, we begin by defining a signature for the types, terms and propositions.

A *FIX signature*, denoted by Sg, is specified by:

- A collection of *types*. The types are built up in the following way. We are given a collection of *basic ground types*, together with the *distinguished ground types unit*, *null*, *nat*, and *fix*. The types are now specified by the following grammar:

$$\alpha ::= \gamma \mid \alpha \times \alpha \mid \alpha + \alpha \mid \alpha \Rightarrow \alpha \mid T\alpha$$

  where γ denotes any ground type.

- A collection of *basic function symbols*, together with the following *distinguished function symbols*: $\langle\rangle$, $\langle \_, \_ \rangle$, Fst, Snd, Inl_α, Inr_β, $\{\}_\alpha$, $\{-,-\}$ λ_α, App, Val, Let , O, Suc, ItNat, ω, σ, It_α.

- A *sorting* for each of the basic function symbols, which is a list of $n+1$ types, and will be written:

$$f\colon \alpha_1, \ldots, \alpha_n \rightarrow \alpha_{n+1}.$$

  In the case that n is zero, we shall write $f\colon \alpha$. We say that f is an *n-ary* basic function symbol when its sorting consists of $n+1$ types.

- A collection of *basic relation symbols*, together with the following *distinguished relation symbols*: $=_\alpha$, true, false, $\wedge$, $\forall_\alpha$, $\Box$, $\Diamond$, $+$.

- A *sorting* for each of the basic relation symbols, which is a list of n types, and will be written:

$$R : \alpha_1, \ldots, \alpha_n.$$

  In the case that n is one, we shall write $R : \alpha$. We say that R is an *n-ary* basic relation symbol when its sorting consists of n types.

We use the signature Sg to define the types, terms and propositions of our logic. Each type is to be regarded as a metaconstant of arity TYPE, each n-ary basic function symbol as a metaconstant of arity $\textsc{term}^n \to \textsc{term}$, and each n-ary basic relation symbol as a metaconstant of arity $\textsc{term}^n \to \textsc{prop}$. The distinguished function and relation symbols are metaconstants; the function symbols representing the simply typed lambda calculus with products and natural numbers have their usual arities, as do the relation symbols representing equality, truth, falsity, conjunction and universal quantification. The remaining metaconstants have the following arities:

1. $\{\}_\alpha$: $\textsc{term} \to \textsc{term}$
2. $\{-,-\}$: $(\textsc{term} \to \textsc{term}) \to (\textsc{term} \to \textsc{term}) \to \textsc{term} \to \textsc{term}$
3. Inl_α: $\textsc{term} \to \textsc{term}$
4. Inr_β: $\textsc{term} \to \textsc{term}$
5. Val: $\textsc{term} \to \textsc{term}$
6. Let : $\textsc{term} \to (\textsc{term} \to \textsc{term}) \to \textsc{term}$
7. ω: $\textsc{term}$
8. σ: $\textsc{term} \to \textsc{term}$
9. It_α: $(\textsc{term} \to \textsc{term}) \to \textsc{term} \to \textsc{term}$
10. $\Box$: $\textsc{term} \to (\textsc{term} \to \textsc{prop}) \to \textsc{prop}$
11. $\Diamond$: $\textsc{term} \to (\textsc{term} \to \textsc{prop}) \to \textsc{prop}$
12. $+$: $(\textsc{term} \to \textsc{prop}) \to (\textsc{term} \to \textsc{prop}) \to \textsc{term} \to \textsc{prop}$

Remark 3.1 We make the following abbreviations: write FM for $\mathsf{App}(F, M)$ and $F^N(M)$ for $\mathsf{ItNat}(F, N, M)$.

Associated with a FIX signature is a collection of *raw terms* and *raw propositions*. In addition to the metaconstants described above, the meta-λ-calculus contains metaconstants which are viewed as object level variables of arity TERM. The raw terms are exactly ($\alpha\beta\eta$ equivalence classes of) the *closed* terms of the meta-λ-calculus which are of arity TERM and the raw propositions are the *closed* terms of the meta-λ-calculus which are of arity PROP.

A *context*, Γ, is a finite list

$$[x_1\colon \alpha_1, \ldots, x_n\colon \alpha_n]$$

where the variables $x_1, \ldots, x_n$ are *distinct*. An *empty* context will be denoted by white space. We will use the (self explanatory) notation $\Gamma,\ x\colon \alpha$, and Γ, Γ' for the concatenation of contexts (where of course x does not occur in Γ). We shall write

$$\Gamma \vdash M\colon \alpha$$

for the judgement that given the context Γ, the raw FIX term M is well formed and has type α. These judgements are generated by the usual type assignment rules for the basic function symbols and the simply typed lambda calculus with products, together with the following rules:

Remark 3.2 It will be assumed in all rule schemes that both the hypothesis and conclusion are well formed. This, together with the use of the theory of expressions and arities, alleviates the need for any side conditions.

Null Type

$$\frac{\Gamma \vdash M\colon \mathit{null}}{\Gamma \vdash \{\}_\alpha(M)\colon \alpha}$$

Binary Coproduct Type

$$\frac{\Gamma \vdash M\colon \alpha}{\Gamma \vdash \mathsf{Inl}_\beta(M)\colon \alpha + \beta} \qquad \frac{\Gamma \vdash N\colon \beta}{\Gamma \vdash \mathsf{Inr}_\alpha(N)\colon \alpha + \beta}$$

$$\frac{\Gamma, x\colon \alpha \vdash F(x)\colon \gamma \quad \Gamma, y\colon \beta \vdash G(y)\colon \gamma \quad \Gamma \vdash C\colon \alpha + \beta}{\Gamma \vdash \{F, G\}(C)\colon \gamma}$$

Computation Type

$$\frac{\Gamma \vdash M : \alpha}{\Gamma \vdash \mathsf{Val}(M) : T\alpha} \qquad \frac{\Gamma \vdash E : T\alpha \quad \Gamma, x : \alpha \vdash F(x) : T\beta}{\Gamma \vdash \mathsf{Let}\,(E, F) : T\beta}$$

Natural Number Type

$$\frac{}{\vdash \mathsf{O} : nat} \qquad \frac{\Gamma \vdash M : nat}{\Gamma \vdash \mathsf{Suc}(M) : nat}$$

$$\frac{\Gamma \vdash M : \alpha \quad \Gamma, x : \alpha \vdash F(x) : \alpha \quad \Gamma \vdash N : nat}{\Gamma \vdash F^N(M) : \alpha}$$

Fix Type

$$\frac{}{\vdash \omega : T\mathit{fix}} \qquad \frac{\Gamma \vdash E : T\mathit{fix}}{\Gamma \vdash \sigma(E) : \mathit{fix}} \qquad \frac{\Gamma, x : T\alpha \vdash F(x) : \alpha \quad \Gamma \vdash N : \mathit{fix}}{\Gamma \vdash \mathsf{It}_\alpha(F, N) : \alpha}$$

We shall write

$$\Gamma \vdash \Phi \; \mathit{prop}$$

for the judgement that given the context Γ, then the raw proposition Φ is well formed. These judgements are generated by the usual rules for intuitionistic predicate calculus with equality *excluding* implication and existential quantification, but augmented by the following rules:

Universal Modality

$$\frac{\Gamma, x : \alpha \vdash \Phi(x)\; \mathit{prop} \quad \Gamma \vdash E : T\alpha}{\Gamma \vdash \Box(E, \Phi)\; \mathit{prop}}$$

Existential Modality

$$\frac{\Gamma, x : \alpha \vdash \Phi(x)\; \mathit{prop} \quad \Gamma \vdash E : T\alpha}{\Gamma \vdash \Diamond(E, \Phi)\; \mathit{prop}}$$

Coproduct

$$\frac{\Gamma, x : \alpha \vdash \Phi(x)\; \mathit{prop} \quad \Gamma, y : \beta \vdash \Psi(y)\; \mathit{prop} \quad \Gamma \vdash C : \alpha + \beta}{\Gamma \vdash (\Phi + \Psi)(C)\; \mathit{prop}}$$

Given a context Γ, a *FIX term* (in context Γ) is any raw FIX term M satisfying $\Gamma \vdash M\colon \alpha$ for some (necessarily unique) type α. We refer to α as the *type* of the FIX term M. A *FIX proposition* is defined similarly. Now that we have the syntax for our FIX logical system, we present rules for deducing the validity of the propositions. These rules will be presented in a sequent natural deduction style. We will use an intuitionistic sequent in context as our basic judgement, which will take the form:

$$\Gamma,\ \Lambda \vdash \Phi$$

Here, Λ is a finite list of propositions. The intended meaning of a judgement is that one has a deduction of Φ which involves a certain number of undischarged hypotheses, each of which must occur in the list Λ. In the case that Λ is empty, we simply omit the symbol Λ from the judgement. A *FIX theory*, Th, is specified by a FIX signature, together with a specific collection of sequents in context, which are called the *axioms* of Th. The *theorems* of Th consist of the least collection of sequents in context which contains the axioms of Th, and is closed under the usual rules of equational logic and lambda calculus with surjective pairing (modulo β and η conversion), augmented by the following rules:

Null Type Falsity

$$\frac{}{x\colon \mathit{null} \vdash \mathsf{false}}$$

Null Type Equations

$$\frac{\Gamma, x\colon \mathit{null} \vdash F(x)\colon \alpha \quad \Gamma \vdash M\colon \mathit{null}}{\Gamma \vdash F(M) =_\alpha \{\}_\alpha(M)}$$

Binary Coproduct Type Equations

$$\frac{\Gamma, x\colon \alpha \vdash F(x)\colon \gamma \quad \Gamma, y\colon \beta \vdash G(y)\colon \gamma \quad \Gamma \vdash M\colon \alpha}{\Gamma \vdash \{F, G\}(\mathsf{Inl}_\beta(M)) =_\gamma F(M)}$$

$$\frac{\Gamma, x\colon \alpha \vdash F(x)\colon \gamma \quad \Gamma, y\colon \beta \vdash G(y)\colon \gamma \quad \Gamma \vdash N\colon \beta}{\Gamma \vdash \{F, G\}(\mathsf{Inr}_\alpha(N)) =_\gamma G(N)}$$

$$\frac{\Gamma, z\colon \alpha + \beta \vdash H(z)\colon \gamma \quad \Gamma \vdash C\colon \alpha + \beta}{\Gamma \vdash \{x.H(\mathsf{Inl}_\beta(x)), y.H(\mathsf{Inr}_\alpha(y))\}(C) =_\gamma H(C)}$$

Computation Type Equations

$$\frac{\Gamma \vdash M{:}\,\alpha \quad \Gamma, x{:}\,\alpha \vdash F(x){:}\,T\beta}{\Gamma \vdash \mathsf{Let}\,(\mathsf{Val}(M), F) =_{T\beta} F(M)} \qquad \frac{\Gamma \vdash E{:}\,T\alpha}{\Gamma \vdash \mathsf{Let}\,(E, x.Val(x)) =_{T\alpha} E}$$

$$\frac{\Gamma \vdash E{:}\,T\alpha \quad \Gamma, x{:}\,\alpha \vdash F(x){:}\,T\beta \quad \Gamma, y{:}\,\beta \vdash G(y){:}\,T\gamma}{\Gamma \vdash \mathsf{Let}\,(\mathsf{Let}\,(E, F), G) =_{T\gamma} \mathsf{Let}\,(E, x.\mathsf{Let}\,(F(x), G))}$$

Natural Number Type Equations

$$\frac{\Gamma \vdash M{:}\,\alpha \quad \Gamma, x{:}\,\alpha \vdash F(x){:}\,\alpha}{\Gamma \vdash F^{\mathsf{O}}(M) =_{\alpha} M}$$

$$\frac{\Gamma \vdash M{:}\,\alpha \quad \Gamma, x{:}\,\alpha \vdash F(x){:}\,\alpha \quad \Gamma \vdash N{:}\,nat}{\Gamma \vdash F^{\mathsf{Suc}(N)}(M) =_{\alpha} F(F^{N}(M))}$$

Fix Type Equations

$$\frac{}{\Gamma \vdash \omega =_{Tfix} \mathsf{Val}(\sigma(\omega))} \qquad \frac{\Gamma \vdash E =_{Tfix} \mathsf{Val}(\sigma(E))}{\Gamma \vdash E =_{Tfix} \omega}$$

$$\frac{\Gamma, x{:}\,T\alpha \vdash F(x){:}\,\alpha \quad \Gamma \vdash E{:}\,Tfix}{\Gamma \vdash \mathsf{It}_{\alpha}(F, \sigma(E)) =_{\alpha} F(\mathsf{Let}\, n \Leftarrow E.\mathsf{Val}(\mathsf{It}_{\alpha}(F, n)))}$$

Universal Modality Propositions

$$\frac{\Gamma, x{:}\,\alpha,\ \Lambda, \mathsf{Val}(x) =_{T\alpha} E \vdash \Phi(x)}{\Gamma \mid \Lambda \vdash \Box(E, \Phi)}$$

$$\frac{\Gamma,\ \Lambda \vdash \Box(E, \Phi) \quad \Gamma,\ \Lambda \vdash \mathsf{Val}(M) =_{T\alpha} E}{\Gamma \mid \Lambda \vdash \Phi(M)}$$

Existential Modality Propositions

$$\frac{\Gamma,\ \Lambda \vdash \mathsf{Val}(M) =_{T\alpha} E \quad \Gamma,\ \Lambda \vdash \Phi(M)}{\Gamma,\ \Lambda \vdash \Diamond(E, \Phi)}$$

$$\frac{\Gamma, x{:}\,\alpha,\ \Lambda, \mathsf{Val}(x) =_{T\alpha} E, \Phi(x) \vdash \Psi \quad \Gamma,\ \Lambda \vdash \Diamond(E, \Phi)}{\Gamma,\ \Lambda \vdash \Psi}$$

Coproduct Propositions

$$\frac{\Gamma, \Lambda \vdash (\Phi + \Psi)(C) \quad \Gamma, x{:}\,\alpha,\ \Lambda,\ \mathsf{Inl}_\beta(x) =_{\alpha+\beta} C,\ \Phi(x) \vdash \Theta(F(x)) \quad \Gamma, y{:}\,\beta,\ \Lambda,\ \mathsf{Inr}_\alpha(y) =_{\alpha+\beta} C,\ \Psi(y) \vdash \Theta(G(y))}{\Gamma \mid \Lambda \vdash \Theta(\{F, G\}(C))}$$

$$\frac{\Gamma,\ \Lambda \vdash \Phi(M) \quad \Gamma, y{:}\,\beta \vdash \Psi(y)\ prop}{\Gamma,\ \Lambda \vdash (\Phi + \Psi)(\mathsf{Inl}_\beta(M))} \qquad \frac{\Gamma,\ \Lambda \vdash \Psi(N) \quad \Gamma, x{:}\,\alpha \vdash \Phi(x)\ prop}{\Gamma,\ \Lambda \vdash (\Phi + \Psi)(\mathsf{Inr}_\alpha(N))}$$

Mono Condition

$$\frac{\Gamma,\ \Lambda \vdash \mathsf{Val}(M) =_{T\alpha} \mathsf{Val}(M')}{\Gamma,\ \Lambda \vdash M =_\alpha M'}$$

Disjoint Sum Condition

$$\frac{}{\Gamma, \Lambda, \mathsf{Inl}_\beta(M) =_{\alpha+\beta} \mathsf{Inr}_\alpha(N) \vdash \mathsf{false}}$$

Modality Condition

$$\frac{\Gamma,\ \Lambda \vdash \mathsf{Let}\,(E, F) =_{T\alpha} \mathsf{Val}(M)}{\Gamma \mid \Lambda \vdash \Diamond(E, x.F(x) =_{T\alpha} \mathsf{Val}(M))}$$

Nat Induction

$$\frac{\Gamma,\ \Lambda \vdash \Phi(\mathsf{O}) \quad \Gamma, n{:}\,nat,\ \Lambda, \Phi(n) \vdash \Phi(\mathsf{Suc}(n))}{\Gamma, n{:}\,nat,\ \Lambda \vdash \Phi(n)}$$

Fix Induction

$$\frac{\Gamma, e{:}\,T\mathit{fix},\ \Lambda, \Box(e, \Phi) \vdash \Phi(\sigma(e))}{\Gamma, n{:}\,\mathit{fix},\ \Lambda \vdash \Phi(n)}$$

This completes the rules for deriving sequents; we refer to the system as the *FIX logic*.

Remark 3.3 [Informal Explanation of the FIX Propositions] The FIX logic has many features in common with intuitionistic predicate calculus; for the latter see [Dum 77]. However, it introduces propositions of the form $(\Phi + \Psi)(z)$, $\Box(e, \Phi)$, $\Diamond(e, \Phi)$, and so we shall describe informally the intended meaning of this syntax.

For coproduct propositions, $(\Phi + \Psi)(z)$, the intended meaning is

$$(\exists x\colon \alpha.z = \mathsf{Inl}_\beta(x) \wedge \Phi(x)) \vee (\exists y\colon \beta.z = \mathsf{Inr}_\alpha(y) \wedge \Psi(y)).$$

which we read as 'it is either the case that z is provably equal to $\mathsf{Inl}_\beta(x)$ and that $\Phi(x)$ holds, or it is the case that z is provably equal to $\mathsf{Inr}_\alpha(y)$ and that $\Psi(y)$ holds.'

For the universal modality, $\Box(e, \Phi)$, the intended meaning is

$$\forall x\colon \alpha.(\mathsf{Val}(x) = e \supset \Phi(x)),$$

which we read as 'for all x of type α, if it is the case that e is provably equal to the value of x then necessarily $\Phi(x)$ holds.'

For the existential modality, $\Diamond(e, \Phi)$, the intended meaning is

$$\exists x\colon \alpha.\mathsf{Val}(x) = e \wedge \Phi(x),$$

which we read as 'it is possible that e is provably equal to $\mathsf{Val}(x)$ and that $\Phi(x)$ holds.'

These modalities are special cases of the notion of 'evaluation modality' introduced in [Pit 91]. Here we equate 'evaluation to a value x' with 'equal to $\mathsf{Val}(x)$' and as a result proof-theoretically stronger properties of the modalities are postulated in the FIX logic than are condsidered in *loc. cit.* This enables us to derive the pleasant properties of the FIX logic given in Theorems 3.10, 3.12 and 3.13.

Remark 3.4 Each of the terms $F^N(M)$ and $\mathsf{It}_\alpha(F, N)$ is unique up to provable equality in the FIX logic. If one just considers the computational lambda calculus enriched with fixpoint types and terms, as we did at the beginning of Section 2, then it is necessary to impose a rule making uniqueness explicit. However, in the full FIX logic, this uniqueness is derivable from the rules for Nat and Fix Induction.

The FIX logic can be presented using rules which are closely related to the categorical semantics given in Section 4. The new system is given by substituting the following rules for their counterparts in the FIX logic.

Equality Propositions

$$\frac{\Gamma,\ x{:}\,\alpha,\ x'{:}\,\alpha, \Lambda,\ x =_\alpha x' \vdash \Phi}{\Gamma,\ x{:}\,\alpha,\ \Lambda \vdash \Phi[x\backslash x']}$$

Conjunction Propositions

$$\frac{\Gamma \mid \Lambda \vdash \Phi \quad \Gamma \mid \Lambda \vdash \Psi}{\Gamma \mid \Lambda \vdash \Phi \wedge \Psi}$$

Universal Quantification Propositions

$$\frac{\Gamma,\ x{:}\,\alpha, \Lambda \vdash \Phi(x)}{\Gamma \mid \Lambda \vdash \forall_\alpha(\Phi)}$$

Universal Modality Propositions

$$\frac{\Gamma,\ x{:}\,\alpha, \Lambda[\mathsf{Val}(x)\backslash e] \vdash \Phi(x)}{\Gamma,\ e{:}\,T\alpha, \Lambda \vdash \Box(e, \Phi)}$$

Existential Modality Propositions

$$\frac{\Gamma,\ x{:}\,\alpha, \Lambda[\mathsf{Val}(x)\backslash e],\ \Phi(x) \vdash \Psi[\mathsf{Val}(x)\backslash e]}{\Gamma,\ e{:}\,T\alpha, \Lambda,\ \Diamond(e, \Phi) \vdash \Psi}$$

Coproduct Propositions

$$\frac{\Gamma,\ x{:}\,\alpha, \Lambda[\mathsf{Inl}_\beta(x)\backslash z],\ \Phi(x) \vdash \Theta(F(x)) \quad \Gamma,\ y{:}\,\beta, \Lambda[\mathsf{Inr}_\alpha(y)\backslash z],\ \Psi(y) \vdash \Theta(G(y))}{\Gamma,\ z{:}\,\alpha+\beta, \Lambda,\ (\Phi+\Psi)(z) \vdash \Theta(\{F, G\}(z))}$$

Theorem 3.5 The original FIX logical system and the system defined by the collection of adjoint rules are equivalent.

Proof The proof is routine manipulation of the logical rules. □

Now that we have defined the FIX logical system, we state a proposition which we shall make use of in Section 5.

Proposition 3.6 Within the FIX logical system, the following birules are derivable:

$$\frac{\Gamma \mid \Lambda \vdash \Phi(M)}{\Gamma \mid \Lambda \vdash \Box(\mathsf{Val}(M), \Phi)} \qquad \frac{\Gamma \mid \Lambda \vdash \Phi(M)}{\Gamma \mid \Lambda \vdash \Diamond(\mathsf{Val}(M), \Phi)}$$

$$\frac{\Gamma \mid \Lambda \vdash \Phi(M) \quad \Gamma,\ y{:}\,\beta,\ \Lambda \vdash \Psi(y)\ \mathit{prop}}{\Gamma \mid \Lambda \vdash (\Phi + \Psi)(\mathsf{Inl}_\alpha(M))} \qquad \frac{\Gamma \mid \Lambda \vdash \Psi(N) \quad \Gamma,\ x{:}\,\alpha,\ \Lambda \vdash \Phi(x)\ \mathit{prop}}{\Gamma \mid \Lambda \vdash (\Phi + \Psi)(\mathsf{Inr}_\beta(N))}$$

The proof of this proposition involves simple manipulations of the logical rules and is omitted. It is worth remarking that in fact the first two of the above birules are equivalent to the Mono Condition rule of the FIX logic, modulo the other rules. Similarly, the Modality Condition rule is equivalent modulo the others to each of the birules

$$\frac{\Gamma \mid \Lambda \vdash \Box(E, x.\Box(F(x), \Phi))}{\Gamma \mid \Lambda \vdash \Box(\mathsf{Let}\,(E, F), \Phi)}$$

$$\frac{\Gamma \mid \Lambda \vdash \Diamond(E, x.\Diamond(F(x), \Phi))}{\Gamma \mid \Lambda \vdash \Diamond(\mathsf{Let}\,(E, F), \Phi)}$$

Remark 3.7 The induction rule for *nat* is just the usual principle of mathematical induction. The induction rule for *fix* can be rendered informally as: to prove that a property $\Phi(n)$ holds of all elements n in *fix*, it is sufficient to prove for all computations e of an element of *fix* that $\Phi(\sigma(e))$ holds if whenever e evaluates to a value, that value satisfies Φ. This principle is consistent, (see Theorem 4.7), but only because the FIX propositions have limited forms. In fact, extending the FIX logic with unrestricted intuitionistic negation, implication or existential quantification renders it inconsistent. We call to mind at this point the admissible predicates of LCF [Pau 87]; predicates of LCF which contain implication and existential quantification are not necessarily admissible.

Proposition 3.8 Extending the *FIX* logic with intuitionistic implication renders the system inconsistent.

Proof Since FIX contains falsity (false), adding implication ($\Phi \supset \Psi$) means that one also has negation ($\neg\Phi \equiv (\Phi \supset \mathsf{false})$). So consider the proposition

$$\Phi(n) \equiv \neg(\sigma(\omega) = n)$$

about $n{:}\mathit{fix}$. We sketch the essential details of the proof in an informal fashion.

$\Phi(n)$ satisfies the hypotheses of Fix Induction. For if $\forall n{\Leftarrow}e.\neg(\sigma(\omega) = n)$ holds then $\neg(\omega = e)$, since otherwise we could deduce $\forall n{\Leftarrow}\omega.\neg(\sigma(\omega) = n)$, which is false because $\mathsf{Val}(n) = \omega$ holds for $n = \sigma(\omega)$. However, σ is provably a bijection and so from $\neg(\omega = e)$ we deduce $\neg(\sigma(\omega) = \sigma(e))$, that is $\Phi(\sigma(e))$. So the induction principle for *fix* entails that $\Phi(n)$ holds of all $n \in \mathit{fix}$, and in particular of $\sigma(\omega)$, which is a contradiction. □

Proposition 3.9 Extending the FIX logic with intuitionistic existential quantification renders the system inconsistent.

Proof This proof mimics the ideas which show that the category $\omega\mathcal{C}po$ together with inclusive subsets does not model standard intuitionistic predicate calculus [Dum 77]. Recall that in $\omega\mathcal{C}po$, Beck Chevalley conditions fail for left adjoints to projections; for if this is not the case we can deduce that such left adjoints take inclusive subsets to inclusive subsets by unravelling Beck Chevalley at a global element in $\omega\mathcal{C}po$. Then considering the ωcpo $\mathbb{N} \times \Omega$ and inclusive subset $\{(m, n) \mid m \in \mathbb{N} \wedge n \in \Omega \setminus \{\top\} \wedge n \leq m\}$ we can deduce that $\{n \mid n \in \Omega \setminus \{\top\}\}$ is inclusive in Ω. This is not so.

Consider the term $\bot \stackrel{\text{def}}{=} \mathsf{It}_{T\mathit{fix}}(e.\mathsf{Let}\,(e, x.x), \sigma(\omega)){:}\,T\mathit{fix}$ and set

$$\Phi(n) \stackrel{\text{def}}{=} \exists_{\mathit{nat}}(m.(u.\sigma(\mathsf{Val}(u)))^m(\sigma(\bot))) =_{\mathit{fix}} n.$$

Using the usual rules for intuitionistic existential quantification together with the FIX rules we may deduce $e, n, e = \mathsf{Val}(n), \Box(e, \Phi) \vdash \Phi(n) \vdash \Phi(\sigma(\mathsf{Val}(n)))$ and from (`fixin`) we have $n{:}\,\mathit{fix} \vdash \Phi(n)$. In particular this means that

$$\vdash \exists_{\mathit{nat}}(m.(u.\sigma(\mathsf{Val}(u)))^m(\sigma(\bot))) =_{\mathit{fix}} \sigma(\omega).$$

Using (`mono`) and that σ is an isomorphism we conclude $\vdash \bot = \omega$. □

We will see that the our logic of fixpoint computations is consistent in the next section when we come to consider models. We next state some metatheorems that witness the constructive nature of the FIX logic and suggest its potential as a programming logic.

Theorem 3.10 ['Existence Property'] If E is a closed term of type $T\alpha$ then $\vdash \exists(E, \Phi)$ is derivable in FIX if and only if there is a closed term M of type α for which $\vdash E =_{T\alpha} \mathsf{Val}(M)$ and $\vdash \Phi(M)$ are derivable. (In other words, a formal proof that E evaluates to a value satisfying Φ necessitates the existence of a term denoting that value.)

Remark 3.11 The deduction that $\vdash E = \mathsf{Val}(M)$ captures in an extensional manner the idea that the computation E *evaluates* to the value M. We refer the reader to Section 6 for further comments.

Theorem 3.12 ['Disjunction Property'] If E is a closed term of coproduct type $\alpha+\beta$, Φ and Ψ are properties of α and β and $\vdash (\Phi+\Psi)(E)$ is derivable in FIX, then either $\vdash E =_{\alpha+\beta} \mathsf{Inl}(M)$ and $\vdash \Phi(M)$ are derivable for some closed term M of type α, or $\vdash E =_{\alpha+\beta} \mathsf{Inr}(N)$ and $\vdash \Psi(N)$ are derivable for some closed term N of type β.

The Existence Property enables one to produce closed terms of type *nat* from a computation of a number (i.e. a closed term of type $T\,nat$) together with a proof that the computation converges. There remains the possibility that a closed term of type *nat* is not a value, i.e. a standard numeral. However, this is not so:

Theorem 3.13 [Standardness of *nat*] Every closed term N of type *nat* in the logic FIX is provably equal to a standard numeral $\mathsf{Suc}^n(\mathsf{O})$, that is one may derive $\vdash N =_{nat} \mathsf{Suc}^n(\mathsf{O})$ in *FIX*. (The number n is uniquely determined by N, because the consistency of FIX (Theorem 4.7) implies that $\mathsf{Suc}^n(\mathsf{O})$ and $\mathsf{Suc}^m(\mathsf{O})$ are not provably equal when $n \neq m$.)

These theorems will be proved in Section 5.

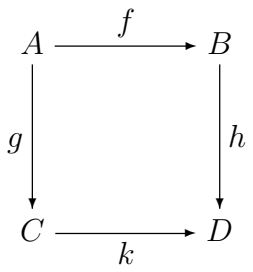
4 Categorical semantics

In proving Theorems 3.10, 3.12, and 3.13, we shall use the fact that FIX theories correspond, in a way to be made precise, to a certain categorical structure. This correspondence is very similar to that between intuitionistic predicate calculus and a particular variety of hyperdoctrine, for example, see [See 83]. Indeed, there is a natural equivalence between FIX theories and FIX hyperdoctrines; thus we begin by giving the definition of this variety of hyperdoctrine.

Definition 4.1 A *FIX category* is a let-ccc with finite coproducts, natural numbers object and fixpoint object, for which each component of the unit of the monad is a monomorphism. A *FIX hyperdoctrine* is specified by a FIX-category $\mathcal{C}$ (referred to as the base category) together with a $\mathcal{C}$-indexed poset, $\mathcal{C}\colon \mathcal{C}^{\mathrm{op}} \to \mathcal{P}oset$ where if $f\colon A \to B$ is a morphism in the base category $\mathcal{C}$, then

we shall denote the corresponding pullback function by $f^* : \mathcal{C}(B) \to \mathcal{C}(A)$, with the fibre at an object A denoted by $\mathcal{C}(A)$.

We shall adopt the following notational convention. If

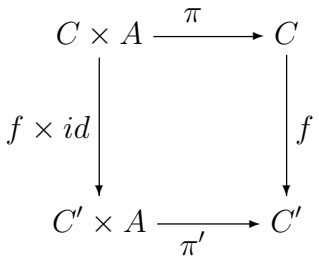


is a commuting square in $\mathcal{C}$ then *right Beck-Chevalley conditions* are said to hold, which will be abbreviated to RBC, if $f^*{:}\mathcal{C}(B) \to \mathcal{C}(A)$ and $k^*{:}\mathcal{C}(D) \to \mathcal{C}(C)$ have right adjoints, which will often be denoted by $\forall f$ and $\forall k$, and these adjoints satisfy the identity $\forall f \circ g^* = h^* \circ \forall k$. We use a dual convention for *left Beck-Chevalley conditions*, LBC.

The indexed poset satisfies the following conditions:

1. The fibres are pointed meet semi-lattices, where in particular the fibre over the initial object of the base category is a singleton. The top element is denoted by $\top$, the bottom element by $\bot$, and the meet of elements $x \in \mathcal{C}(A)$ and $y \in \mathcal{C}(A)$ by $x \wedge y \in \mathcal{C}(A)$. The pullback functions are required to preserve meets, top and bottom elements.

2. RBC holds for all squares of the form



   where the morphisms π and π' are product projections.

3. RBC and LBC hold for all squares of the form

$$\begin{array}{ccc} C \times A & \xrightarrow{id \times \eta} & C \times TA \\ \Big\downarrow{\scriptstyle f \times id} & & \Big\downarrow{\scriptstyle f \times id} \\ C' \times A & \xrightarrow[id \times \eta]{} & C' \times TA \end{array}$$

Also, the hyperdoctrine enjoys a form of Frobenius Reciprocity, namely given $x \in \mathcal{C}(C \times TA)$ and $y \in \mathcal{C}(C \times A)$ we have

$$\exists(id \times \eta)((id \times \eta)^*(x) \wedge y) = x \wedge \exists(id \times \eta)(y).$$

These conditions ensure the soundness of the rules for deducing validity of universal and existential modality propositions in the FIX logic.

4. There is an operation $+$ on fibres

$$+ : \mathcal{C}(C \times A) \times \mathcal{C}(C \times B) \longrightarrow \mathcal{C}(C \times (A + B))$$

which is natural in C. Suppose we are given elements

$$\begin{array}{ll} x \in \mathcal{C}(C \times A) & u \in \mathcal{C}(C \times (A + B)) \\ y \in \mathcal{C}(C \times B) & z \in \mathcal{C}(C \times D) \end{array}$$

and morphisms $f : C \times A \to D$ $g : C \times B \to D$. Then we demand that

$$\frac{(id_C \times i)^*(u) \wedge x \leq \langle \pi_A, f \rangle^*(z) \quad (id_C \times j)^*(u) \wedge y \leq \langle \pi_B, g \rangle^*(z)}{u \wedge (x + y) \leq \langle \pi, \{f \mid g\} \rangle^*(z)}$$

where $i : A \to (A + B)$ $j : B \to (A + B)$ are coproduct insertions,

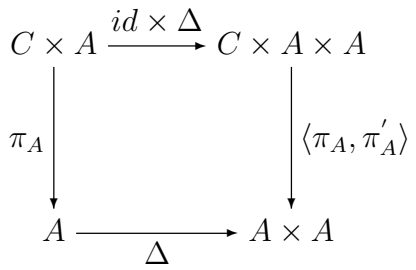
$$(C \times A) + (C \times B) \xrightarrow{\phi} C \times (A + B)$$

is the obvious isomorphism and

$$\pi_A : C \times A \to C \quad \pi_B : C \times B \to C \quad \pi : C \times (A + B) \to C$$

are product projections. Finally, $\{f \mid g\} \stackrel{\text{def}}{=} [f, g] \circ \phi^{-1}$ where $[f, g]$ arises from the coproduct structure of C. Note that if $x + y$ exists, it is determined uniquely. These requirements ensure the soundness of the rules for deducing validity of coproduct propositions.

5. LBC holds for



The left adjoint to $id \times \Delta$ satisfies the following Frobenius Reciprocity condition,

$$\exists(id \times \Delta) \circ (id \times \Delta)^*(x) = x \wedge \exists(id \times \Delta) \circ \pi_C^*(\top)$$

where $x \in \mathcal{C}(C \times A \times A)$ and $\pi_C \colon C \times A \to C$. (Recall that the pullback function π_C^* preserves the top element by definition.) These conditions ensure the soundness of the rules for equality.

6. We demand the inequalities

$$(\eta \times \eta)^* \circ \exists\Delta(\top_{TA}) \leq \exists\Delta(\top_A)$$

and

$$\langle i, j \rangle^* \circ \exists\Delta(\top_{A+B}) = \bot$$

where $\top_A \in \mathcal{C}(A)$, $\top_{TA} \in \mathcal{C}(TA)$ and $\top_{A+B} \in \mathcal{C}(A+B)$ are the top elements of the fibres and $i \colon A \to (A+B)$ $j \colon B \to (A+B)$ are coproduct insertions. This guarantees the soundness of the mono condition and the disjoint sum condition.

7. Given the morphism

$$f \colon C \times TA \times B \times A \to TB$$

then we demand the inequality

$$\langle \mathit{lift}(f), \eta\pi_1 \rangle^* \circ \exists\Delta(\top) \leq \exists(id \times \eta) \circ \langle f, \eta\pi_2 \rangle^* \circ \exists\Delta(\top)$$

where $\top \in \mathcal{C}(TA)$ is the top element of the fibre and

$$\pi_1 \colon C \times TA \times B \times TA \to B \quad \pi_2 \colon C \times TA \times B \times A \to B$$

are product projections. This ensures the soundness of the modality condition.

Finally, to complete the definition of the FIX hyperdoctrine, there are two fibrewise induction conditions and a coherence condition. The induction conditions ensure soundness of the induction rules in the logic and the coherence condition guarantees that semantic equality of terms coincides with derivable equality in the FIX logic.

8. Given elements
$$x \in \mathcal{C}(C)\ ,\ y \in \mathcal{C}(C \times N)$$
then we demand that
$$\frac{x \leq \langle id, 0 \circ !\rangle^*(y) \quad \pi^*(x) \wedge y \leq (id \times s)^*(y)}{\pi^*(x) \leq y}$$
where $\pi\colon C \times N \to C$ is a product projection and $0\colon 1 \to N$ $s\colon N \to N$ are given as part of the structure of the natural numbers object in the base category.

9. Given elements
$$x \in \mathcal{C}(C)\ ,\ y \in \mathcal{C}(C \times \Omega)$$
then we demand that
$$\frac{\pi^*(x) \wedge \forall(id \times \eta)(y) \leq (id \times \sigma)^*(y)}{\pi'^*(x) \leq y}$$
where $\pi\colon C \times T\Omega \to C$ $\pi'\colon C \times \Omega \to C$ are product projections and $\sigma\colon T\Omega \to \Omega$ is given as part of the structure of the fixpoint object in the base category.

10. Given morphisms $f, g\colon B \to A$, and the diagonal $\Delta\colon A \to A \times A$, then we ask that
$$\frac{\langle f, g\rangle^* \circ \exists\Delta(\top) = \top}{f = g \ \text{ in } \ \mathcal{C}}$$

This completes Definition 4.1.

A *morphism* of FIX categories is a functor which preserves the categorical structure up to isomorphism. A *morphism* of FIX hyperdoctrines C and $\mathcal{C}'$ is specified by a FIX-category morphism between the base categories, (referred

to as the base functor), say $F\colon \mathcal{C} \to \mathcal{C}'$, together with an indexed collection of monotone functions, called fibre morphisms, $F_A\colon \mathcal{C}(A) \to \mathcal{C}'(FA)$ $A \in ob\mathcal{C}$. These monotone functions are required to preserve the structure of the fibres in a canonical fashion. For example, the pullback functions are preserved by the fibre morphisms in the sense that given a morphism $f\colon A \Rightarrow B$ in $\mathcal{C}$, the following square commutes

$$\begin{array}{ccc} \mathcal{C}(A) & \xrightarrow{F_A} & \mathcal{C}'(FA) \\ {\scriptstyle f^*}\uparrow & & \uparrow{\scriptstyle (F(f))^*} \\ \mathcal{C}(B) & \xrightarrow[F_B]{} & \mathcal{C}'(FB) \end{array}$$

Also, the structure of the fibres is preserved by the fibre morphisms; for example

- Given $\top \in \mathcal{C}(A)$, then $F_A(\top) = \top \in \mathcal{C}'(FA)$,
- given $x, y \in \mathcal{C}(A)$, then $F_A(x \wedge y) = F_A(x) \wedge F_A(y)$,
- given $x \in \mathcal{C}(C \times A)$ and $y \in \mathcal{C}(C \times B)$ then

$$F_{C\times(A+B)}(x + y) = F_{C\times A}(x) + F_{C\times B}(y),$$

The remaining structure of the fibres is preserved in a similar way; we leave the details to the reader.

The definition of a FIX hyperdoctrine is quite involved, and so the first task is to give an example.

Examples 4.2 *The category of predomains, $\omega\mathcal{C}po$, equipped with the lifting monad as described in Section 2, is a FIX category. There is a $\omega\mathcal{C}po$-indexed poset, $\mathcal{I}\colon \omega\mathcal{C}po^{\mathrm{op}} \to \mathcal{P}oset$ where I takes an ωcpo D to the set of inclusive subsets of D, which are ordered by inclusion, and I takes each continuous function $f\colon D \to D'$ to its inverse image function f^{-1} restricted to inclusive subsets. It is trivial to check that $f^{-1}\colon \mathcal{I}(D') \to \mathcal{I}(D)$ is well defined and indeed monotone, and that I is a functor. We define the operations that make $\omega\mathcal{C}po$ a FIX hyperdoctrine, but omit detailed verifications.*

1. *With meet given by intersection of inclusive subsets, it is clear that each fibre is a pointed meet semi-lattice. It is easy to see that each pullback function is a morphism of pointed meet semi-lattices.*

2. *The right adjoint to projection is given by restriction of the dual image functions to inclusive subsets; that RBC holds is trivial. Finally* $\mathcal{I}(\varnothing) = \{\varnothing\}$ *is a singleton.*

3. *The existence of left adjoints is well known, given by restriction of the set theoretic direct image functions to inclusive subsets. The right adjoint to*
$$(id \times \iota)^{-1}\colon \mathcal{I}(C \times D_\perp) \to \mathcal{I}(C \times D)$$
is given by
$$\forall(id \times \iota)\colon \mathcal{I}(C \times D) \to \mathcal{I}(C \times D_\perp)$$
where for $I \in \mathcal{I}(C \times D)$ *we define*
$$\forall(id \times \iota)(I) \stackrel{\text{def}}{=} (id \times \iota)(I) \cup \{(c, \perp) \ : \ \forall c \in C\}$$
It is easy to see that this is a good definition and yields the required adjoint. Checking RBC and LBC is easy; Frobenius Reciprocity is virtually immediate.

4. *Let* $i\colon D \Rightarrow D + D'$ *and* $j\colon D' \to D + D'$ *be coproduct insertions. Given*
$$I \in \mathcal{I}(C \times D) \quad \textit{and} \quad J \in \mathcal{I}(C \times D'),$$
we define
$$+ : \mathcal{I}(C \times D) \times \mathcal{I}(C \times D') \Rightarrow \mathcal{I}(C \times (D + D'))$$
by
$$I + J \stackrel{\text{def}}{=} \exists(id \times i)(I) \cup \exists(id \times j)(J).$$

Note that the fibrewise induction conditions are satisfied because any inclusive subset of an ωcpo *is an* ωcpo.

Remark 4.3 Some other strong monads on $\omega\mathcal{C}po$ (such as those for side-effects and exceptions) were shown in Section 2 to possess fixpoint objects. Thus we get other FIX categories based on ωcpos by changing from lifting

to one of these other monads. However the $\omega\mathcal{C}po$-indexed poset of inclusive subsets will not yield a FIX hyperdoctrine over these FIX categories. This is because the notion of FIX hyperdoctrine is tailored to fit the FIX logic which treats 'evaluation to a value x' as meaning 'equal to $\mathsf{Val}(x)$'. Such a strict interpretation of evaluation is appropriate for a (constructive) treatment of the termination/nontermination aspects of computation; but for other aspects, weaker notions of hyperdoctrine are needed. (For example, [Pit 91] gives a hyperdoctrine over $\omega\mathcal{C}po$ possessing appropriate evaluation modalities for the side-effects monad.)

Next we give the categorical semantics of the FIX logic in a FIX hyperdoctrine C. A *structure*, $\mathbf{M}$, in C for a given FIX-signature Sg is specified by the following data:

- An object $[\![\gamma]\!]$ for each basic ground type γ of Sg, and
- for each basic function symbol $f\colon \alpha_1, \ldots, \alpha_n \to \alpha$, a morphism in C of the form $[\![f]\!]\colon [\![\alpha_1]\!] \times \ldots \times [\![\alpha_n]\!] \to [\![\alpha]\!]$, and
- for each basic relation symbol $R\colon \alpha_1, \ldots, \alpha_n$ an element $[\![R]\!]$ of a fibre of C, where $[\![R]\!] \in \mathcal{C}([\![\alpha_1]\!] \times \ldots \times [\![\alpha_n]\!])$.

For each term in context, we will assign a meaning in the base category C in the following way. The types are interpreted as objects in the category, where the interpretation of a type α is denoted by $[\![\alpha]\!]$. Set

- $[\![\mathit{unit}\,]\!] \stackrel{\text{def}}{=} 1$ where 1 is the terminal object.
- $[\![\mathit{null}]\!] \stackrel{\text{def}}{=} 0$ where 0 is the initial object.
- $[\![\mathit{nat}\,]\!] \stackrel{\text{def}}{=} N$ where N is the natural numbers object.
- $[\![\mathit{fix}\,]\!] \stackrel{\text{def}}{=} \Omega$ where Ω is the fixpoint object.
- $[\![\alpha \times \beta]\!] \stackrel{\text{def}}{=} [\![\alpha]\!] \times [\![\beta]\!]$.
- $[\![\alpha + \beta]\!] \stackrel{\text{def}}{=} [\![\alpha]\!] + [\![\beta]\!]$.
- $[\![\alpha \Rightarrow \beta]\!] \stackrel{\text{def}}{=} [\![\alpha]\!] \Rightarrow [\![\beta]\!]$.
- $[\![T\alpha]\!] \stackrel{\text{def}}{=} T[\![\alpha]\!]$.

Given a context $\Gamma = [x_1 \colon \alpha_1, \ldots, x_n \colon \alpha_n]$, we let $[\![\Gamma]\!] \stackrel{\text{def}}{=} [\![\alpha_1]\!] \times \ldots \times [\![\alpha_n]\!]$. Then for each context Γ, term M and type α for which $\Gamma \vdash M \colon \alpha$ is a valid judgement we give a morphism

$$[\![\Gamma \vdash M \colon \alpha]\!] \colon [\![\Gamma]\!] \to [\![\alpha]\!].$$

Note that when $M \colon \alpha$ is a valid judgement, because the type α is uniquely determined by M and Γ, we will abbreviate $[\![\Gamma \vdash M \colon \alpha]\!]$ to just $[\![\Gamma.M]\!]$. The semantics of terms-in-context is defined by a structural induction on terms:

- $[\![\Gamma, x \colon \alpha, \Gamma'.x]\!] \stackrel{\text{def}}{=} \pi \colon [\![\Gamma]\!] \times [\![\alpha]\!] \times [\![\Gamma']\!] \to [\![\alpha]\!]$
- Let $f \colon \alpha_1, \ldots, \alpha_n \to \alpha$ be a basic function symbol; then
  $[\![\Gamma.f(M_1, \ldots, M_n)]\!] \stackrel{\text{def}}{=} [\![f]\!] \circ \langle [\![\Gamma.M_1]\!], \ldots, [\![\Gamma.M_n]\!] \rangle$
  $\colon [\![\Gamma]\!] \to [\![\alpha_1]\!] \times \ldots \times [\![\alpha_n]\!] \to [\![\alpha]\!]$
- $[\![\Gamma.\langle\rangle]\!] \stackrel{\text{def}}{=} ! \colon [\![\Gamma]\!] \to 1$
- $[\![\Gamma.\{\}_\alpha(M)]\!] \stackrel{\text{def}}{=} ! \circ [\![\Gamma.M]\!] \colon [\![\Gamma]\!] \to 0 \to [\![\alpha]\!]$
- $[\![\Gamma.\langle M, N\rangle]\!] \stackrel{\text{def}}{=} \langle [\![\Gamma.M]\!], [\![\Gamma.N]\!] \rangle \colon [\![\Gamma]\!] \to [\![\alpha]\!] \times [\![\beta]\!]$
- $[\![\Gamma.\mathsf{Fst}(P)]\!] \stackrel{\text{def}}{=} \pi \circ [\![\Gamma.P]\!] \colon [\![\Gamma]\!] \to [\![\alpha]\!] \times [\![\beta]\!] \to [\![\alpha]\!]$
- $[\![\Gamma.\mathsf{Snd}(P)]\!] \stackrel{\text{def}}{=} \pi \circ [\![\Gamma.P]\!] \colon [\![\Gamma]\!] \to [\![\alpha]\!] \times [\![\beta]\!] \to [\![\beta]\!]$
- $[\![\Gamma.\{F, G\}(C)]\!] \stackrel{\text{def}}{=} \{[\![\Gamma, x \colon \alpha.F(x)]\!] \mid [\![\Gamma, y \colon \beta.G(x)]\!]\} \circ \langle id, [\![\Gamma.C]\!] \rangle$
  $\colon [\![\Gamma]\!] \to [\![\Gamma]\!] \times ([\![\alpha]\!] + [\![\beta]\!]) \to [\![\gamma]\!]$
- $[\![\Gamma.\mathsf{Inl}_\beta(M)]\!] \stackrel{\text{def}}{=} i \circ [\![\Gamma.M]\!] \colon [\![\Gamma]\!] \to [\![\alpha]\!] \to [\![\alpha]\!] + [\![\beta]\!]$
- $[\![\Gamma.\mathsf{Inr}_\alpha(N)]\!] \stackrel{\text{def}}{=} j \circ [\![\Gamma.N]\!] \colon [\![\Gamma]\!] \to [\![\beta]\!] \to [\![\alpha]\!] + [\![\beta]\!]$
- $[\![\Gamma.\lambda_\alpha(F)]\!] \stackrel{\text{def}}{=} cur([\![\Gamma, x \colon \alpha.F(x)]\!]) \colon [\![\Gamma]\!] \Rightarrow ([\![\alpha]\!] \Rightarrow [\![\beta]\!])$
- $[\![\Gamma.FM]\!] \stackrel{\text{def}}{=} ap \circ \langle [\![\Gamma.F]\!], [\![\Gamma.M]\!] \rangle \colon [\![\Gamma]\!] \to ([\![\alpha]\!] \Rightarrow [\![\beta]\!]) \times [\![\alpha]\!] \to [\![\beta]\!]$
- $[\![\Gamma.\mathsf{Val}(M)]\!] \stackrel{\text{def}}{=} \eta \circ [\![\Gamma.M]\!] \colon [\![\Gamma]\!] \to [\![\alpha]\!] \to T[\![\alpha]\!]$

- $[\![\Gamma.\mathsf{Let}\,(E,F)]\!] \stackrel{\text{def}}{=} \mathit{lift}([\![\Gamma, x\colon\alpha.F(x)]\!]) \circ \langle id, [\![\Gamma.E]\!]\rangle\colon [\![\Gamma]\!] \to [\![\Gamma]\!] \times T[\![\alpha]\!] \to T[\![\beta]\!]$
- $[\![\Gamma.\mathsf{O}]\!] \stackrel{\text{def}}{=} 0 \circ !\colon [\![\Gamma]\!] \to 1 \to N$
- $[\![\Gamma.\mathsf{Suc}(N)]\!] \stackrel{\text{def}}{=} s \circ [\![\Gamma.N]\!]\colon [\![\Gamma]\!] \to N \to N$
- $[\![\Gamma.F^N(M)]\!] \stackrel{\text{def}}{=} h \circ \langle id, [\![\Gamma.N]\!]\rangle\colon [\![\Gamma]\!] \to [\![\Gamma]\!] \times N \to [\![\alpha]\!]$

  where h is the unique morphism arising from the initiality property of the natural numbers object together with the morphism

  $$[\![\Gamma, x\colon\alpha.F(x)]\!] \circ \langle id, [\![\Gamma.M]\!]\rangle\colon [\![\Gamma]\!] \to [\![\Gamma]\!] \times [\![\alpha]\!] \to [\![\alpha]\!]$$

- $[\![\Gamma.\omega]\!] \stackrel{\text{def}}{=} \omega \circ !\colon [\![\Gamma]\!] \to 1 \to T\Omega$
- $[\![\Gamma.\sigma(E)]\!] \stackrel{\text{def}}{=} \sigma \circ [\![\Gamma.E]\!]\colon [\![\Gamma]\!] \to T\Omega \to \Omega$
- $[\![\Gamma.\mathsf{It}_\alpha(F,N)]\!] \stackrel{\text{def}}{=} f \circ \langle id, [\![\Gamma.N]\!]\rangle\colon [\![\Gamma]\!] \to [\![\Gamma]\!] \times \Omega \to [\![\alpha]\!]$

  where f is the unique morphism arising from the initiality property of the fixpoint object together with the morphism

  $$[\![\Gamma, x\colon T\alpha.F(x)]\!]\colon [\![\Gamma]\!] \times T[\![\alpha]\!] \to [\![\alpha]\!]$$

Finally, for each context Γ and proposition Φ for which we can derive $\Gamma \vdash \Phi\ \mathit{prop}$, we specify an element of a fibre

$$[\![\Gamma.\Phi]\!] \in \mathcal{C}([\![\Gamma]\!]),$$

where we adopt a similar notation to that for terms in context. The semantics of propositions in context is defined using the structure of the propositions:

- $[\![\Gamma.\mathsf{true}]\!] \stackrel{\text{def}}{=} \top \in \mathcal{C}([\![\Gamma]\!])$
- $[\![\Gamma.\mathsf{false}]\!] \stackrel{\text{def}}{=} \bot \in \mathcal{C}([\![\Gamma]\!])$
- $[\![\Gamma.M =_\alpha N]\!] \stackrel{\text{def}}{=} \langle [\![\Gamma.M]\!], [\![\Gamma.N]\!]\rangle^* \circ \exists\Delta(\top)$ where $\top \in \mathcal{C}([\![\alpha]\!])$
- $[\![\Gamma.\Phi \wedge \Psi]\!] \stackrel{\text{def}}{=} [\![\Gamma.\Phi]\!] \wedge [\![\Gamma.\Psi]\!]$

- $[\![\Gamma.\forall_\alpha(\Phi)]\!] \stackrel{\text{def}}{=} \forall\pi_1([\![\Gamma, x{:}\,\alpha.\Phi]\!])$
- $[\![\Gamma.\Box(E,\Phi)]\!] \stackrel{\text{def}}{=} \langle id, [\![\Gamma.E]\!]\rangle^* \circ \forall(id \times \eta)([\![\Gamma, x{:}\,\alpha.\Phi(x)]\!])$
- $[\![\Gamma.\Diamond(E,\Phi)]\!] \stackrel{\text{def}}{=} \langle id, [\![\Gamma.E]\!]\rangle^* \circ \exists(id \times \eta)([\![\Gamma, x{:}\,\alpha.\Phi(x)]\!])$
- $[\![\Gamma.(\Phi+\Psi)(M)]\!] \stackrel{\text{def}}{=} \langle id, [\![\Gamma.M]\!]\rangle^*([\![\Gamma, x{:}\,\alpha.\Phi(x)]\!] + [\![\Gamma, y{:}\,\beta.\Psi(y)]\!])$

The categorical semantics interprets substitution of terms in terms, and terms in propositions in the usual manner. Indeed we have the following two lemmas which make this precise. The first lemma deals with substitution of terms for variables in another term:

Lemma 4.4 The categorical semantics interprets the substitution of a term for a variable in a term via composition in the category. More precisely, if $\Gamma \vdash M_i{:}\,\alpha_i$ for $i = 1, \ldots, n$ and also $\Gamma' \vdash N{:}\,\beta$ where $\Gamma = [x_1{:}\,\alpha_1, \ldots, x_n{:}\,\alpha_n]$, we have

$$[\![\Gamma.N[\vec{M}/\vec{x}]]\!] = [\![\Gamma'.N]\!] \circ \langle [\![\Gamma.M_1]\!], \ldots, [\![\Gamma.M_n]\!]\rangle$$

where the notation $N[\vec{M}/\vec{x}]$ indicates a simultaneous substitution.

The next lemma tells us how substitution of terms for variables in propositions is modelled.

Lemma 4.5 Let $\Gamma' \vdash \Phi$ be a FIX proposition in context, where the context is $\Gamma' = [x_1{:}\,\alpha_1, \ldots, x_n{:}\,\alpha_n]$ and let $\Gamma \vdash M_i{:}\,\alpha_i$ for $i = 1, \ldots, n$ be FIX terms in context. Then

$$[\![\Gamma.\Phi[\vec{M}/\vec{x}]]\!] = \langle [\![\Gamma.M_1]\!], \ldots, [\![\Gamma.M_n]\!]\rangle^*([\![\Gamma'.\Phi]\!]),$$

where the notation $\Phi[\vec{M}/\vec{x}]$ indicates a simultaneous substitution.

If Λ is a finite list of propositions, each of which is well formed in the context Γ, then let

$$[\![\Gamma.\Lambda]\!] \stackrel{\text{def}}{=} \bigwedge_{\Theta\in\Lambda} [\![\Gamma.\Theta]\!].$$

A structure $\mathbf{M}$ in a FIX hyperdoctrine C *satisfies* a sequent in context $\Gamma \mid \Lambda \vdash \Phi$ if

$$[\![\Gamma.\Lambda]\!] \leq [\![\Gamma.\Phi]\!]$$

holds in the fibre $\mathcal{C}([\![\Gamma]\!])$. Given a FIX theory, Th, then $\mathbf{M}$ is called a *model* of the theory if it satisfies all the axioms of Th.

The categorical semantics of the FIX logic is sound; indeed we have:

Proposition 4.6 Let C be a FIX hyperdoctrine, Th a FIX theory, and $\mathbf{M}$ a model of Th in C. Then $\mathbf{M}$ satisfies any sequent in context which is a theorem of Th.

Proof We need to check that the collection of sequents in context which are satisfied by $\mathbf{M}$ is closed under the rules for generating sequents in context. The proof uses lemmas 4.4 and 4.5. □

Taking C to be the FIX hyperdoctrine of inclusive subsets over the FIX category of $\omega\mathcal{C}po$ equipped with the lifting monad, and taking Th to be the empty theory in the above proposition, we have:

Theorem 4.7 The FIX logical system is consistent, in the sense that $\vdash$ false is not provable from the rules given in Section 3 □

Now that the definition of the semantics is complete, we will establish that there is a natural equivalence between the syntax and the semantics. We need some notation.

Proposition 4.8 For each FIX theory Th over some FIX-signature Sg, we may construct a syntactic FIX hyperdoctrine, which we shall denote by $\mathcal{C}(Th)$ or sometimes just $\mathcal{F}$.

Proof The base category is constructed from the types and terms of the FIX theory:

- The objects of $\mathcal{F}$ are the types of the signature Sg,
- and the morphisms are equivalence classes of terms with at most one metavariable, where the equivalence relation is given by provable equality in the FIX logic.

Composition is given by the usual substitution of terms; it is a tedious but straightforward task to check that this does define a FIX category.

We now define an F-indexed functor to the category of meet semi-lattices. We also use F to denote the functor. For each object $\alpha \in \mathcal{F}$, the underlying set consists of equivalence classes of propositions in a single variable context, $x{:}\,\alpha \vdash \Phi(x)$. We shall often omit the context itself; with this convention we define the equivalence relation by

$$\Phi(x) \sim \Psi(y) \quad \text{iff} \quad \Phi(x) \vdash \Psi(x) \quad \text{and} \quad \Psi(x) \vdash \Phi(x).$$

Then referring to equivalence classes by representatives, we order this set by

$$\Phi(x) \leq \Psi(y) \quad \text{iff} \quad \Phi(x) \vdash \Psi(x).$$

Given a morphism $F\colon \alpha \to \beta$, in F, then the pullback function $F^*\colon \mathcal{F}(\beta) \to \mathcal{F}(\alpha)$ is defined by substitution: $F^*(\Phi(y)) \stackrel{\text{def}}{=} \Phi(F)$. The remaining details are routine verifications. □

We also have the following

Proposition 4.9 Given a FIX hyperdoctrine $\mathcal{C}$, then we can define a FIX theory which we denote by $Th(\mathcal{C})$.

Proof The basic ground types are the objects of C, and basic function symbols copies of the morphisms of C. The basic relation symbols are copies of the elements of the hyperdoctrine fibres. This yields a FIX signature, and there is an evident canonical structure for this in C. The axioms of the theory are exactly those sequents in context which are satisfied by the canonical structure. The theorems of $Th(\mathcal{C})$ are generated by the usual rules. □

We state the categorical logic correspondence:

Theorem 4.10 Let C be a FIX hyperdoctrine; then there is an equivalence of hyperdoctrines

$$Eq\colon \mathcal{C}(Th(\mathcal{C})) \simeq \mathcal{C},$$

where Eq is a FIX hyperdoctrine morphism, in the sense that there is a categorical equivalence of base categories, and each fibre morphism

$$Eq_\alpha\colon \mathcal{C}(Th(\mathcal{C}))(\alpha) \to \mathcal{C}(Eq_\alpha(\alpha))$$

is an isomorphism of posets.

5 Categorical logical relations

Now that we have formalised the correspondence between FIX theories and FIX hyperdoctrines, we define a new FIX hyperdoctrine and use it, together with its corresponding logic, to prove the theorems which we stated at the end of Section 3.

Remark 5.1 The hyperdoctrine construction which is detailed below provides a proof of the Existence and Disjunction properties of FIX in much the same way that Freyd's gluing construction may be used to see the existence and disjunction properties of standard intuitionistic predicate calculus. Our construction, in essence, packages the technique of logical relations, as Freyd's glued topos packages the techniques of realizability.

Let $\mathcal{F}$ be the syntactic hyperdoctrine constructed from the pure FIX logic (that is to say the FIX theory with no extralogical axioms) and let $\Gamma : \mathcal{F} \to \omega\mathcal{C}po$ denote the functor which assigns to each object $\alpha \in \mathcal{F}$ its set $\Gamma(\alpha)$ of global elements equipped with the *discrete* partial order. We construct a new FIX hyperdoctrine, denoted by $\mathcal{L}r(\Gamma)$, using a construction that is closely allied to the theory of logical relations. An object of $\mathcal{L}r(\Gamma)$ is a triple $(D, \lhd, \alpha)$, where D is an object of $\omega\mathcal{C}po$, α is an object of $\mathcal{F}$ and $\lhd$ is an inclusive subset of $D \times \Gamma(\alpha)$. A morphism $(D, \lhd, \alpha) \to (D', \lhd', \alpha')$ in $\mathcal{L}r(\Gamma)$ is a pair (f, F), where $f : D \to D'$ in $\omega\mathcal{C}po$, $F : \alpha \to \alpha'$ in $\mathcal{F}$, satisfying the following condition:

$$\forall d \in D. \forall M \in \Gamma(\alpha). d \lhd M \text{ implies } f(d) \lhd' (FM).$$

Finally we need to define a $\mathrm{Lr}(\Gamma)$-indexed poset. We shall denote the fibre at an object $(D, \lhd, \alpha)$ by $\mathcal{L}r(D, \trianglelefteq, \alpha)$. The elements of the fibre consist of all triples $(S, \trianglelefteq, \Phi(x))$, where

1. $S \in \mathcal{I}(D)$, i.e. S runs over the inclusive subsets of the ωcpo D,
2. $\Phi(x) \in \mathcal{F}(\alpha)$, where $\Phi(x)$ is a representative,
3. $\trianglelefteq \in \mathcal{I}(S \times \Gamma_{\Phi}(\alpha) \cap \lhd)$ where

$$\Gamma_{\Phi(x)}(\alpha) \stackrel{\text{def}}{=} \{M \in \Gamma(\alpha) \mid\vdash \Phi(M)\},$$

and the ordering is specified coordinatewise. Given a morphism

$$(f, F) \colon (D', \lhd', \alpha') \to (D, \lhd, \alpha)$$

in $\mathrm{Lr}(\Gamma)$, we define the pullback function

$$(f, F)^* \colon \mathcal{L}r(D, \trianglelefteq, \alpha) \to \mathcal{L}r(D', \trianglelefteq', \alpha')$$

by

$$(f,F)^*(S,\trianglelefteq,\Phi(x)) \stackrel{\text{def}}{=} (f^{-1}(S),\trianglelefteq^*,\Phi(F))$$

where

$$\trianglelefteq^* \stackrel{\text{def}}{=} \{(d,M) \in f^{-1}(S) \times \Gamma_{\Phi(F)}(\alpha') \cap \lhd' |\ f(d) \trianglelefteq FM\}.$$

Proposition 5.2 The above recipe produces another FIX hyperdoctrine $\mathcal{L}r(\Gamma)$.

Proof We write just Lr for $\mathcal{L}r(\Gamma)$. We check that we have defined a base FIX category; most of the details are simple calculations, once it is clear how one defines the various categorical constructs. The terminal object is

$$(*, \lhd_{unit}, unit),$$

where $* \lhd_{unit} id_{unit}$. The binary product is given by

$$(D,\lhd,\alpha) \times (D',\lhd',\alpha') \stackrel{\text{def}}{=} (D \times D', \lhd_\times, \alpha \times \alpha'),$$

where (with the obvious notation),

$$(d,d') \lhd_\times N \text{ iff } d \lhd \mathsf{Fst}(N) \text{ and } d' \lhd' \mathsf{Snd}(N).$$

It is clear that $\lhd_\times$ is inclusive, and easy to check the remaining details. Exponentials of objects are defined by

$$(D',\lhd',\alpha') \Rightarrow (D,\lhd,\alpha) \stackrel{\text{def}}{=} (D' \Rightarrow D, \lhd_{ap}, \alpha' \Rightarrow \alpha),$$

where

$$f \lhd_{ap} M \quad \text{iff} \quad \forall d' \in D'. \forall L' \in \Gamma(\alpha'). d' \lhd' L' \text{ implies } f(d') \lhd ap \circ \langle M, L' \rangle$$

and ap is the evaluation morphism in F. The transpose rule is given by

$$\frac{(f,F)\colon (D \times D', \lhd_\times, \alpha \times \alpha') \to (D'', \lhd'', \alpha'')}{(cur(f), cur(F))\colon (D,\lhd,\alpha) \to (D' \Rightarrow D'', \lhd_{ap}, \alpha' \Rightarrow \alpha'')}$$

and the evaluation morphism is (ap, ap). Finite coproducts are also defined in the same (hopefully now familiar) coordinatewise/logical relations manner. The natural numbers object of Lr is specified by

$$(\mathbb{N}, \lhd_{nat}, nat),$$

where

$$n \lhd_{nat} N \text{ iff } N = \mathsf{Suc}^n(\mathsf{O}),$$

and the zero and successor morphisms are the expected coordinatewise ones. We now show that for a particular choice of monad, our category Lr does indeed become a FIX category. The action of the monad on objects is specified by

$$T(D, \lhd, \alpha) \stackrel{\text{def}}{=} (D_\perp, \lhd_T, T\alpha),$$

where

$$e \lhd_T E \quad \text{iff} \quad \forall d \in D.[d] = e \text{ implies } \exists M \in \Gamma(\alpha).d \lhd M \text{ and } \eta_\alpha M = E,$$

and $\eta \stackrel{\text{def}}{=} (\iota, \eta_\alpha) \colon (D, \lhd, \alpha) \to (D_\perp, \lhd, T\alpha)$, with $\iota \colon D \to D_\perp$ the canonical inclusion. Finally the lifting rule is

$$\frac{(f, F) \colon (D \times D', \lhd_\times, \alpha \times \alpha') \to (D''_\perp, \lhd''_T, T\alpha'')}{(f_\perp, \mathit{lift}(F)) \colon (D \times D'_\perp, \lhd'_T, \alpha \times T\alpha') \to (D''_\perp, \lhd''_T, T\alpha'')}$$

where $f_\perp(d, [d']) \stackrel{\text{def}}{=} f(d, d')$ and $f_\perp(d, \perp) \stackrel{\text{def}}{=} \perp$.

Now we show that Lr does indeed possess a fixpoint object. This will be determined up to isomorphism; thus as for the previous constructs we exhibit a candidate and show that it satisfies the required properties. The expected candidate for the fixpoint object would be $(\Omega, \lhd_{\mathit{fix}}, \mathit{fix})$, with structure morphism (σ, σ). By definition of the action of the monad on objects, in the relation $\lhd_T^{\mathit{fix}}$ one has $\perp \lhd_T^{\mathit{fix}} M$, for any $M \in \Gamma(T\mathit{fix})$. As (σ, σ) must preserve the relation, then $0 \lhd_{\mathit{fix}} \sigma M$ must hold, and the action of the monad yields $[0] \lhd_T^{\mathit{fix}} \eta\sigma M$. Once again (σ, σ) preserves this, so we must have $1 \lhd_{\mathit{fix}} \sigma\eta\sigma M$. In general we are forced to have $n \lhd_{\mathit{fix}} (\sigma\eta)^n \sigma M$. Finally, considering that the relation $\lhd_{\mathit{fix}}$ has to be a certain inclusive subset, we are led to the following definition:

$(\Omega, \lhd_{\mathit{fix}}, \mathit{fix})$ is a fixpoint object for T over Lr, where

- $n \lhd_{\mathit{fix}} N$ iff $\exists M \in \Gamma(\mathit{fix}).N = (\sigma\eta)^n M$, and
- $\top \lhd_{\mathit{fix}} N$ iff $\forall n \in \Omega \setminus \{\top\}.n \lhd_{\mathit{fix}} N$.

We check that the relation $\lhd_{\mathit{fix}}$ is inclusive. Let $\theta \colon \mathbb{N} \to \Omega$ be a function satisfying $\theta(r) \leq \theta(r+1)$. Set $n_r \equiv \theta(r)$, so a chain in $\lhd_{\mathit{fix}}$ is given by a sequence $n_r \lhd_{\mathit{fix}} N$ where $N \in \Gamma(\mathit{fix})$. We need to check that $\bigvee_{r \in \mathbb{N}} n_r \lhd_{\mathit{fix}} N$.

If $\bigvee n_r$ is not $\top$ we are done. Otherwise, given any $n \in \Omega \setminus \{\top\}$, we can choose $r \in \mathbb{N}$ such that $n_r \geq n$. As $n_r \lhd_{\mathit{fix}} N$, we get $N = (\sigma\eta)^{n_r} M = (\sigma\eta)^n (\sigma\eta)^{n_r - n} M$, and so $n \lhd_{\mathit{fix}} N$. As n was arbitrary, we are done. Now we check that (σ, σ) is a morphism in Lr, where $(\sigma, \sigma) \colon (\Omega_\perp, \lhd_T^{\mathit{fix}}, T\mathit{fix}) \to (\Omega, \lhd_{\mathit{fix}}, \mathit{fix})$. We have three cases to cover.

1. If $\perp \lhd_T^{\mathit{fix}} N$ then $\sigma(\perp) = 0 \lhd_{\mathit{fix}} \sigma N$.
2. Suppose that $[\top] \lhd_T^{\mathit{fix}} \eta N$. Then $\top \lhd_{\mathit{fix}} N$ and hence $\forall n \in \Omega \setminus \{\top\}. n \lhd_{\mathit{fix}} N$. In particular, we have $n - 1 \lhd_{\mathit{fix}} N$, and so there is some $M \in \Gamma(\mathit{fix})$ for which $N = (\sigma\eta)^{n-1} M$, giving $\sigma\eta N = (\sigma\eta)^n M$. So we have $\forall n \in \Omega \setminus \{\top\}. n \lhd_{\mathit{fix}} \sigma\eta N$, that is $\top \lhd_{\mathit{fix}} \sigma\eta N$.
3. This is immediate from the definition of $\lhd_{\mathit{fix}}$.

Finally, we have to verify that our definition yields an initial T-algebra in Lr. Take

$$(f, F) \colon (D_\perp, \lhd_T, T\alpha) \to (D, \lhd, \alpha).$$

The unique mediating morphism for (σ, σ) has to be $(\tilde{f}, \tilde{F}) \stackrel{\text{def}}{=} (\mathit{it}(f), \mathit{it}(F))$ whose coordinates are the mediating morphisms in $\omega\mathcal{C}po$ and F. Firstly we check that it is a morphism in Lr. Suppose that $n \lhd_{\mathit{fix}} N$. Then for some M we get $N = (\sigma\eta)^n M$. From the definition of the $\lhd_T$ relation, we get

$$\perp \lhd_T \mathit{lift}(\eta\tilde{F})\sigma^{-1} M$$

and so

$$f(\perp) \lhd F\mathit{lift}(\eta\tilde{F})\sigma^{-1} M = \tilde{F} M.$$

Now suppose that $f^r(\perp) \lhd \tilde{F}(\sigma\eta)^{r-1} M$, where $r \leq n - 1$. Clearly

$$[f^r(\perp)] \lhd_T \eta\tilde{F}(\sigma\eta)^{r-1} M,$$

and so

$$f^{r+1}(\perp) \lhd \tilde{F}(\sigma\eta)^r M.$$

Inductively we have

$$f^{n+1}(\perp) \lhd \tilde{F}(\sigma\eta)^n M = \tilde{F} N,$$

which is what we had to prove. Lastly, if $\top \lhd_{\mathit{fix}} N$ we need

$$\bigvee f^n(\perp) \lhd \tilde{F} N,$$

which follows from inclusivity of $\lhd$. We leave the reader to verify that the morphisms

$$(\top, \omega)\colon (*, \lhd_{unit}, unit) \to (\Omega_\perp, \lhd_T^{fix}, Tfix),$$

and

$$(\sigma, \sigma)\colon (\Omega_\perp, \lhd_T^{fix}, Tfix) \to (\Omega, \lhd_{fix}, fix)$$

constitute a fixpoint object in Lr.

Now it is time to verify that all the conditions required of the Lr-indexed poset hold. We give the constructions of the adjoints and operations which appear in the definition of a FIX hyperdoctrine, but omit the verification of the conditions which the adjoints and operations satisfy. Firstly we define the right adjoint to

$$(\pi, \mathsf{Fst}(z))^*\colon \mathcal{L}r(C, \lhd', \gamma) \to \mathcal{L}r(C \times D, \lhd' \times \lhd, \gamma \times \alpha)$$

which we write as

$$\forall\pi\colon \mathcal{L}r(C \times D, \lhd_\times, \gamma \times \alpha) \to \mathcal{L}r(C, \lhd', \gamma)$$

(with $\lhd_\times$ an abbreviation for $\lhd' \times \lhd$) where we define

$$\forall\pi(S, \trianglelefteq_\times, \Phi(z)) \stackrel{\text{def}}{=} (\forall\pi(S), \trianglelefteq_\times^\circ, \forall\pi(\Phi(z)))$$

with

$$\begin{aligned} \trianglelefteq_\times^\circ \stackrel{\text{def}}{=} \; & \{(c, N) \in \forall\pi(S) \times \Gamma_{\forall\pi(\Phi(z))}(\gamma) \cap \lhd' | \\ & \forall d \in D. \forall M \in \Gamma(\alpha). d \lhd M \;\text{ implies }\; (c, d) \lhd_\times \langle N, M\rangle\}. \end{aligned}$$

The right adjoint to

$$(id \times \iota, id \times \eta)^*\colon \mathcal{L}r(C \times D_\perp, \lhd' \times \lhd_T, \gamma \times T\alpha) \to \mathcal{L}r(C \times D, \lhd' \times \lhd, \gamma \times \alpha)$$

which we shall write as

$$\Box\colon \mathcal{L}r(C \times D, \lhd_\times, \gamma \times \alpha) \to \mathcal{L}r(C \times D_\perp, \lhd_\times^T, \gamma \times T\alpha)$$

(with $\lhd_\times$ and $\lhd_\times^T$ abbreviations for $\lhd' \times \lhd$ and $\lhd' \times \lhd_T$) is defined by

$$\Box(S, \trianglelefteq_\times, \Phi(u)) \stackrel{\text{def}}{=} (\Box(S), \trianglelefteq_\times^\circ, \Box(\Phi(u))),$$

where

$$\trianglelefteq_\times^\circ \stackrel{\text{def}}{=} \{((c,e),\langle N,E\rangle) \in \Box(S)\times\Gamma_{\Box(\Phi(u))}(\gamma\times T\alpha)\cap \lhd_\times^T \mid$$
$$\forall d\in D.\forall M\in\Gamma(\alpha).d\lhd M, e=[d], E=\mathsf{Val}(M) \text{ implies } (c,d)\trianglelefteq_\times \langle N,M\rangle\}.$$

We now define the left adjoint to

$$(id\times\iota, id\times\eta)^*\colon \mathcal{L}r(C\times D_\perp, \lhd'\times\lhd_T, \gamma\times T\alpha)\to\mathcal{L}r(C\times D,\lhd'\times\lhd,\gamma\times\alpha)$$

which we shall write as

$$\Diamond\colon \mathcal{L}r(C\times D,\lhd_\times,\gamma\times\alpha)\to\mathcal{L}r(C\times D_\perp,\lhd_\times^T,\gamma\times T\alpha)$$

where we define

$$\Diamond(S,\trianglelefteq_\times,\Phi(u)) \stackrel{\text{def}}{=} (\Diamond(S),\trianglelefteq_\times^\circ,\Diamond(\Phi(u)))$$

with

$$\trianglelefteq_\times^\circ \stackrel{\text{def}}{=} \{((c,e),\langle N,E\rangle)\in\Diamond(S)\times\Gamma_{\Diamond(\Phi(u))}(\gamma\times T\alpha)\cap\lhd_\times^T\mid$$
$$\exists d\in D.\exists M\in\Gamma(\alpha).e=[d],\ E=\mathsf{Val}(M),\ (d,c)\trianglelefteq_\times\langle N,M\rangle)\}.$$

We define the $+$ operation, where

$$+\colon \mathcal{L}r(C\times D,\lhd_C\times\lhd,\gamma\times\alpha)\times\mathcal{L}r(C\times D',\lhd_C\times\lhd',\gamma\times\alpha')\to$$
$$\mathcal{L}r(C\times(D+D'),\lhd_C\times(\lhd+\lhd'),\gamma\times(\alpha+\alpha'))$$

and

$$(I,\trianglelefteq_1,\Phi(v)) \in \mathcal{L}r(C\times D,\lhd_C\times\lhd,\gamma\times\alpha)$$
$$(J,\trianglelefteq_2,\Psi(w)) \in \mathcal{L}r(C\times D',\lhd_C\times\lhd',\gamma\times\alpha')$$

by taking the sum of these elements to be

$$(I+J,\trianglelefteq_3,\Phi(v)+\Psi(w)),$$

where

$$\trianglelefteq_3 \stackrel{\text{def}}{=} \{((c,e),\langle N,E\rangle)\in(I+J)\times\Gamma_{\Phi(v)+\Psi(w)}(\gamma\times(\alpha+\alpha'))\cap\lhd_C\times(\lhd+\lhd')\mid$$
$$\exists d\in D.\exists M\in\Gamma(\alpha).e=i(d), E=\mathsf{Inl}(M),(c,d)\trianglelefteq_1\langle N,M\rangle$$
$$\text{or}$$
$$\exists d'\in D'.\exists M'\in\Gamma(\alpha').e=j(d'),E=\mathsf{Inr}(M'),(c,d')\trianglelefteq_2\langle N,M'\rangle\}.$$

The left adjoint to

$$(id \times \Delta, id \times \Delta)^* : \mathcal{L}r(C \times D \times D, \lhd'_\times, \gamma \times \alpha \times \alpha) \to \mathcal{L}r(C \times D, \lhd_\times, \gamma \times \alpha)$$

written as

$$\exists(id \times \Delta) : \mathcal{L}r(C \times D, \lhd_\times, \gamma \times \alpha) \to \mathcal{L}r(C \times D \times D, \lhd'_\times, \gamma \times \alpha \times \alpha)$$

is defined by

$$\exists(id \times \Delta)(S, \trianglelefteq_\times, \Phi(u)) \stackrel{\text{def}}{=} (\exists(id \times \Delta)(S), \trianglelefteq^\circ_\times, \exists(id \times \Delta)(\Phi(u))),$$

where

$$\begin{aligned} \trianglelefteq^\circ_\times \stackrel{\text{def}}{=} \{((c,e), \langle N, E\rangle) \in \exists(id \times \Delta)(S) \times \Gamma_{\exists(id\times\Delta)(\Phi(u))}(\gamma \times \alpha \times \alpha) \cap \lhd'_\times \mid \\ \exists d \in D. \exists M \in \Gamma(\alpha). e = \Delta(d), E = \Delta(M), (c,d) \trianglelefteq_\times \langle N, M\rangle\}. \end{aligned}$$

□

We are now in a position to prove the theorems stated on Page 21. We need to make one further observation, namely

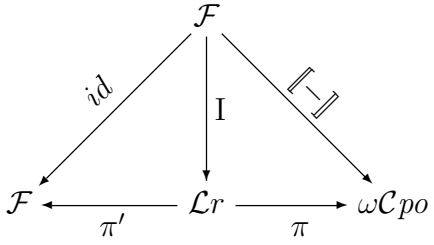
Proposition 5.3 The FIX hyperdoctrine F, arising from the pure FIX logic, is initial amongst all FIX hyperdoctrines.

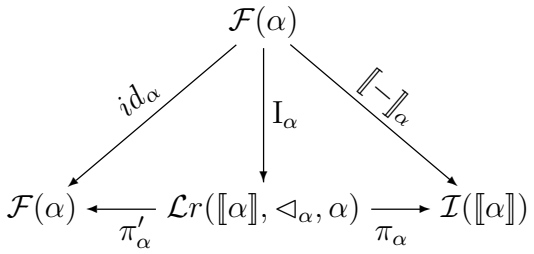
Proof This is immediate from the definition of FIX hyperdoctrine morphism. □

Using the initiality of F, we see that there are FIX hyperdoctrine morphisms $[\![-]\!]$ and I; and in addition there are obvious projections π and π' where

$$\begin{aligned} [\![-]\!] &: \mathcal{F} \to \omega\mathcal{C}po \\ \mathrm{I} &: \mathcal{F} \to \mathcal{L}r \\ \pi &: \mathcal{L}r \to \omega\mathcal{C}po \\ \pi' &: \mathcal{L}r \to \mathcal{F}. \end{aligned}$$

These FIX hyperdoctrine morphisms satisfy the following commutative diagrams:





Let us now prove the Disjunction Property for the FIX logical system. First note that the closed term $E\colon \alpha+\beta$ corresponds to a morphism $E\colon unit \to \alpha+\beta$ in F. The action of the base functor part of $\mathrm{I}\colon \mathcal{F} \to \mathcal{L}r$ on this morphism, using the above commutative diagrams, is

$$\mathrm{I}(E) = ([\![E]\!], E)\colon (*, \lhd_{unit}, unit) \to ([\![\alpha]\!] + [\![\beta]\!], \lhd_{\alpha+\beta}, \alpha + \beta).$$

Also, the following square commutes:

$$\begin{array}{ccc} \mathcal{F}(\alpha+\beta) & \xrightarrow{\mathrm{I}_{\alpha+\beta}} & \mathcal{L}r([\![\alpha]\!] + [\![\beta]\!], \lhd_{\alpha+\beta}, \alpha+\beta) \\ \Big\downarrow{\scriptstyle E^*} & & \Big\downarrow{\scriptstyle ([\![E]\!], E)^*} \\ \mathcal{F}(unit) & \xrightarrow[\mathrm{I}_{unit}]{} & \mathcal{L}r(1, \lhd_{unit}, unit) \end{array}$$

The theorem follows by observing the action of the two possible routes of the square. Let z be a variable of type $\alpha+\beta$, and consider $(\Phi+\Psi)(z) \in \mathcal{F}(\alpha+\beta)$. Then we have

$$\begin{array}{rcl} \mathrm{I}_{unit}(E^*((\Phi+\Psi)(z)) & = & \mathrm{I}((\Phi+\Psi)(E)) \\ & & ([\![(\Phi+\Psi)(E)]\!], \trianglelefteq_{unit}, (\Phi+\Psi)(E)) \end{array}$$

where because $\vdash (\Phi+\Psi)(E)$ by hypothesis, and I_{unit} preserves greatest elements, the relation $\trianglelefteq_{unit}$ must be non-empty. Also, we have

$$\begin{array}{rcl} ([\![E]\!], E)^*(\mathrm{I}_{\alpha+\beta}((\Phi+\Psi)(z))) & = & ([\![E]\!], E)^*([\![(\Phi+\Psi)(z)]\!], \trianglelefteq_{\alpha+\beta}, (\Phi+\Psi)(z)) \\ & = & ([\![E]\!]^{-1}([\![(\Phi+\Psi)(z)]\!]), \trianglelefteq^*_{\alpha+\beta}, (\Phi+\Psi)(E)) \end{array}$$

where

$$\begin{array}{rl} \trianglelefteq^*_{\alpha+\beta} = \{(*, id_{unit}) \in [\![E]\!]^{-1}([\![(\Phi+\Psi)(z)]\!]) \times \Gamma_{(\Phi+\Psi)(E)}(unit) \cap \lhd_{unit} \mid & \\ [\![E]\!](*) \trianglelefteq_{\alpha+\beta} E \circ id\}. & \end{array}$$

But this relation is exactly $\trianglelefteq_{unit}$, hence is *non-empty*, yielding

$$[\![E]\!](*) \trianglelefteq_{\alpha+\beta} E,$$

that is

$$[\![E]\!](*) \triangleleft_{\alpha+\beta} E.$$

By definition of the relation $\triangleleft_{\alpha+\beta}$ this means without loss of generality there is a global element $M \in \Gamma(\alpha)$, that is a closed term M, for which

$$\vdash E =_{\alpha+\beta} \mathsf{Inl}(M),$$

and from this we may derive $\vdash \Phi(M)$ using Proposition 3.6.

The proof of the Existence Property is very similar. We take a proposition $\Diamond(e, \Phi) \in \mathcal{F}(T\alpha)$ and use the square

$$\begin{array}{ccc} \mathcal{F}(T\alpha) & \xrightarrow{\mathrm{I}_{T\alpha}} & \mathcal{L}r([\![\alpha]\!]_\perp, \triangleleft_{T\alpha}, T\alpha) \\ \Big\downarrow{\scriptstyle E^*} & & \Big\downarrow{\scriptstyle ([\![E]\!], E)^*} \\ \mathcal{F}(unit) & \xrightarrow[\mathrm{I}_{unit}]{} & \mathcal{L}r(1, \triangleleft_{unit}, unit) \end{array}$$

As above, this yields

$$\begin{array}{rcl} \trianglelefteq_{unit} & = & \trianglelefteq^*_{T\alpha} \\ & = & \{(*, id) \in [\![E]\!]^{-1}([\![\Diamond(e, \Phi)]\!]) \times \Gamma_{\Diamond(E,\Phi)}(unit) \cap \triangleleft_{unit} |\ [\![E]\!](*) \trianglelefteq_{T\alpha} E\}. \end{array}$$

This has to be non empty and so

$$[\![E]\!](*) \trianglelefteq_{T\alpha} E,$$

that is

$$[\![E]\!](*) \triangleleft_{T\alpha} E.$$

Hence there is some closed M for which $\vdash E =_{T\alpha} \mathsf{Val}(M)$ and using Proposition 3.6 we have $\vdash \Phi(M)$.

Finally we prove the Standardness of nat. Let N be a closed term of type nat. Using the square

$$\begin{array}{ccc} \mathcal{F}(nat) & \xrightarrow{\mathrm{I}_{nat}} & \mathcal{L}r(\mathbb{N}, \lhd_{nat}, nat) \\ \Big\downarrow{\scriptstyle N^*} & & \Big\downarrow{\scriptstyle ([\![N]\!], N)^*} \\ \mathcal{F}(unit) & \xrightarrow[\mathrm{I}_{unit}]{} & \mathcal{L}r(1, \lhd_{unit}, unit) \end{array}$$

and arguing in the same way as for the previous two theorems, we conclude that

$$[\![N]\!](*) \trianglelefteq_{nat} N,$$

and from this we deduce that

$$\vdash N =_{nat} \mathsf{Suc}^n(0),$$

using the definition of the $\lhd_{nat}$ relation in the natural numbers object of Lr. This completes the proofs of the theorems from section 3.

We finish this section by remarking that the Existence Property expresses a *formal adequacy* of the FIX logic for the metalanguage. Indeed, we have the following

Corollary 5.4 Given a closed term E of type $T\alpha$, it is provably equal to a value $\mathsf{Val}(M)$, where M is a closed term of type α, if and only if the $\omega\mathcal{C}po$ interpretation

$$[\![E]\!] \in [\![T(\alpha)]\!] = [\![\alpha]\!]_\perp$$

is not $\perp$.

6 Concluding remarks

The predicate $x \Leftarrow e\ [x \in \alpha, e \in T(\alpha)]$ of *evaluation* is implicit in the FIX logic, but as was pointed out in Remark 3.3, it is treated here in a very 'extensional' way as equivalent to $\mathsf{Val}(x) = e$. It is possible to envisage a weaker logic than FIX (and a corresponding kind of categorical structure) in which $x \Leftarrow e\ [x \in \alpha, e \in T(\alpha)]$ is an atomic predicate satisfying

$$\frac{}{M \Leftarrow \mathsf{Val}(M)} \qquad \frac{M \Leftarrow E \quad N \Leftarrow F(M)}{N \Leftarrow (\mathsf{Let}\,(E, F))}$$

and in which there are modified rules for the bounded quantifiers. For some recent progress in this area we refer the reader to [Pit 91]. There are strong connections between FIX and the traditional 'axiomatic domain theory' of LCF [Pau 87] and to Plotkin's approach to denotational semantics using partial continuous functions [Plo 85]. Our logic appears inherently more constructive, since it is based on the notion of *evaluation* of a (possibly non-terminating) computation to a value, rather than on *non-termination* and on *information* ordering between (possibly partial) computations. However, the precise relationship between the FIX logic and 'axiomatic domain theory' has yet to be clarified.

FIX establishes a novel approach to fixed point equations at the level of functions. This technique can be extended to the level of types by using a dependently typed logic which contains a universal type and a fixpoint type. The fundamental idea is that a recursive type (domain equation) induces a recursive function on the universal type. The fixpoint type yields a fixpoint for this function; the type coded by this fixpoint corresponds to the solution of the domain equation. For details see [Cro 91].

FIX is not an 'integrated' logic—proofs of propositions are external to the system. Undoubtedly something to aim for is a system combining features of FIX with those of the Calculus of Constructions [CoqHu 88], obtaining both the 'terms-as-computations' and 'terms-as-proofs' paradigms in a single (consistent!) system.

Acknowledgements

The authors would like to thank the referees for their comments, and the members (M. Hyland, E. Moggi, V. de Paiva, W. Phoa, E. Ritter) of the CLICS Club in Cambridge for numerous discussions about this work.

References

[CoqHu 88] Coquand T. and Huet G. (1988), The Calculus of Constructions, *Inform. and Computation* 76, 95–120.

[CroPit 90] Crole R. L. and Pitts A. M. (1990), New Foundations For Fixpoint Computations, *in* "Proceedings, 5th Annual Symposium

on Logic In Computer Science", IEEE Computer Society Press, Washington, pp. 489–497.

[Cro 91] Crole R. L. (1991), "Programming Metalogics with a Fixpoint Type", Ph.D. Thesis, University of Cambridge, U.K.

[Cur 86] Curien P.-L. (1986), "Categorical Combinators, Sequential Algorithms and Functional Programming", Pitman, London.

[Dum 77] Dummett M. (1977), "Elements of Intuitionism", Oxford University Press.

[Gir 89] Girard J.-Y. (1989), "Proofs and Types" (translated and with appendices by P. Taylor and Y. Lafont), Cambridge University Press.

[Freyd 9?] Freyd P. J. (199?), Algebraically complete categories, *in* "Proceedings, Category Theory '90, Como", Lecture Notes in Math., Springer-Verlag, Berlin, *to appear*.

[Laf 88] Lafont Y. (1988), "Logiques, Catégories et Machines", Ph.D. thesis, Univ. Paris VII.

[LamSco 86] Lambek J. and Scott P. J. (1986), "Introduction to Higher Order Categorical Logic", Cambridge Studies in Advanced Mathematics 7, Cambridge University Press.

[LehSmy 81] Lehmann D. J. and Smyth M. B. (1981), Algebraic specification of data types: a synthetic approach, *Math. Systems Theory* 14, 97–139.

[MacLane 71] MacLane S. (1971), "Categories for the Working Mathematician", Springer-Verlag, Berlin.

[Mog 88] Moggi E. (1988), Computational lambda-calculus and monads, LFCS Technical Report 88-66, University of Edinburgh.

[Mog 89] Moggi E. (1989), Computational lambda-calculus and monads, *in* "Proceedings, 4th Annual Symposium on Logic in Computer Science", IEEE Computer Society Press, Washington, pp. 14–23.

[Mog1 91] Moggi E. (1991), Notions of computations and monads, *Inform. and Computation* 93, 55–92.

[MarL 83] Martin-Löf P. (1983), Notes on the domain theoretic interpretation of type theory, *in* "Proceedings, Workshop on Semantics of Programming Languages", Chalmers Univ.

[NorPetSmi 90] Nordström B., Petersson K. and Smith J. M. (1990), "Programming in Martin-Löf's Type Theory", Oxford University Press.

[Pau 87] Paulson L. C. (1987), "Logic and Computation", Cambridge University Press.

[Pit 91] Pitts A. M. (1991), Evaluation logic, *in* "IV Higher Order Workshop, Banff 1990" (G. Birtwistle, Ed.), Workshops in Computing, Springer-Verlag, Berlin, *to appear*.

[Plo 77] Plotkin G. D. (1977), LCF considered as a programming language, *Theoretical Computer Science* 5, 223–255.

[Plo 85] Plotkin G. D. (1985), Denotational semantics with partial functions, unpublished lecture notes from CSLI Summer School.

[Sco 69] Scott D. S. (1969), A type-theoretic alternative to CUCH, ISWIM, OWHY, unpublished manuscript, University of Oxford.

[See 83] Seely R. A. G. (1983), Hyperdoctrines, natural deduction and the Beck condition, *Zeitschr. f. math. Logik und Grundlagen d. Math.* 29, 505–542.

[SmyPlo 82] Smyth M. B. and Plotkin G. D. (1982), The category-theoretic solution of recursive domain equations, *SIAM J. Comput.* 11, 761–783.