A General Semantics for Evaluation Logic

Eugenio Moggi*
moggi@disi.unige.it

DISI, Univ. of Genova
v.le Benedetto XV 3, 16132 Genova, ITALY

Abstract

The semantics of Evaluation Logic proposed in [14] relies on additional properties of monads. This paper proposes an alternative semantics, which drops all additional requirement on monads, at the expense of stronger assumptions on the underlying category. These assumptions are satisfied by any topos, but not by the category of cpos. However, in the setting of Synthetic Domain Theory (see [7, 23]) it is possible to reconcile the needs of Denotational Semantics with those of Logic.

1 Introduction

Evaluation logic (EL_T) is a *typed predicate logic* originally proposed by [18], which is based on the metalanguage ML_T for computational monads (see [12]) and permits statements about the evaluation of programs to values by the use of *evaluation modalities*: necessity and possibility. In particular, EL_T might be used for axiomatizing computation-related properties of a monad or devising computationally adequate theories (see [18]), and it appears useful when addressing the question of logical principles for reasoning about the behavior of programs. EL_T provides a *framework* for presenting programming languages and program logics in analogy with LCF (see [4]), but with the additional abstraction mechanism given by computational types.

[18] adopts a very general notion of model for EL_T based on indexed posets, where the interpretation of evaluation modalities is not *canonical* (i.e. it is not determined by the underlying monad). In this way, the modalities have to be interpreted on a case by case basis. To overcome this problem, [14] proposes a *canonical* semantics of necessity, under the assumption that the monad satisfies some additional properties. Unfortunately, these properties are not satisfied by some monads used in Denotational Semantics.

This paper proposes a *canonical* semantics of necessity, which makes no assumptions on the monad T (which must be *fibered*), but it requires the underlying category $\mathcal{C}$ to have finite limits and a *stable* and *proper factorization system*. The latter requirement is not an issue in toposes or regular categories, but cannot be satisfied in the category of cpos. However, it is possible to reconcile the needs of Denotational Semantics with those for interpreting EL_T. In fact, Synthetic Domain Theory (SDT) provides several models consisting of a logical universe $\mathcal{C}$ (usually a topos) and a full reflective sub-category $\mathcal{D}$ of $\mathcal{C}$, which is *as good as* the category of cpos for the purposes of Denotational Semantics. Therefore, any monad T over $\mathcal{D}$, used for interpreting some programming language (but not EL_T), may be extended (via the reflection) to the whole of $\mathcal{C}$, and used for interpreting EL_T. Moreover, when the typed predicate logic is sufficiently expressive (i.e. it allows formation of subtypes and has equality and existential quantification), necessity is definable and axioms for it may be proved sound formally, without referring to the categorical semantics.

The rest of the paper is organized as follows: Section 2 gives the syntax of EL_T and exemplifies its new semantics; Section 3 discusses limitations of the semantics proposed in [14]; Section 4 presents the new semantics in full generality and gives further examples; Section 5 explains how, in the setting of SDT, it is possible to define an interpretation of EL_T starting from an interpretation of a programming language; Section 6 relates the new semantics to the previous one; Section 7 shows that necessity is definable in terms of subset types, existential quantification and equality, and gives sound axioms for necessity.

*This work is supported by ESPRIT BRA 6811 (Categorical Logic In Computer Science II) and EC SCIENCE twinning ERBSC1*CT920795 (Progr. Lang. Semantics and Program Logics).

2 Syntax and informal semantics

EL_T is a (conservative) extension of typed predicate logic with equality. ML_T is the term language of EL_T, therefore ML_T is a (conservative) extension of typed equational logic. As typed predicate logic one may take logics such as First Order Logic (FOL), Higher Order Logic (HOL) or LCF ([22, 18]). One may also consider a predicate logic with dependent types. In fact, we need to consider such a logic to be able to express the modalities of EL_T.

2.1 Key features of ML_T and EL_T

The syntactic categories of EL_T are types, terms and formulas (as in first order logic with dependent types).

- We write $\Gamma \vdash \tau$ type for "τ is a well formed type in context Γ". Types are closed under the formation rule

  $$(T)\ \frac{\Gamma \vdash \tau \text{ type}}{\Gamma \vdash T\tau \text{ type}}$$

  $T\tau$ is called a **computational type**, and terms of type $T\tau$ should be thought of as programs which return values of type τ.

  A context Γ is a sequence $x_1{:}\tau_1, \ldots, x_n{:}\tau_n$, where x_i are distinct variables and τ_i are well formed types.

- We write $\Gamma \vdash e{:}\tau$ for "e is a well formed term of type τ in context Γ". Terms are closed under the formation rules

  $$(\text{lift})\ \frac{\Gamma \vdash e{:}\tau}{\Gamma \vdash [e]{:}T\tau}$$

  $$(\text{let})\ \frac{\Gamma \vdash e_1{:}T\tau_1 \qquad \Gamma, x{:}\tau_1 \vdash e_2{:}T\tau_2}{\Gamma \vdash (\text{let}\, x{\Leftarrow}e_1 \,\text{in}\, e_2){:}T\tau_2}\ x \notin \tau_2$$

  Intuitively the program $[e]$ simply returns the value e, while $(\text{let}\, x{\Leftarrow}e_1 \,\text{in}\, e_2)$ first evaluates e_1 and binds the result to x, then evaluates e_2.

- We write $\Gamma \vdash \phi$ prop for "ϕ is a well formed formula in context Γ". Formulas are closed under the formation rules

  $$(\text{necessity})\ \frac{\Gamma \vdash e{:}T\tau \qquad \Gamma, x{:}\tau \vdash \phi \text{ prop}}{\Gamma \vdash [x{\Leftarrow}e]\phi \text{ prop}}$$

  $$(\text{possibility})\ \frac{\Gamma \vdash e{:}T\tau \qquad \Gamma, x{:}\tau \vdash \phi \text{ prop}}{\Gamma \vdash \langle x{\Leftarrow}e\rangle\phi \text{ prop}}$$

  $$(\text{evaluation})\ \frac{\Gamma \vdash e{:}T\tau \qquad \Gamma \vdash v{:}\tau}{\Gamma \vdash e{\Downarrow}v \text{ prop}}$$

  Intuitively the formula $[x{\Leftarrow}e]\phi$ means that every possible result of program e satisfies ϕ, $\langle x{\Leftarrow}e\rangle\phi$ means that some possible result of program e satisfies ϕ, and $e{\Downarrow}v$ means that v is one possible result of program e.

- We write $\Gamma \vdash \Phi \Longrightarrow \phi$ for "Φ entails ϕ".

Remark 2.1 In the formation rule (let) it is unclear what type should be assigned to $(\text{let}\, x{\Leftarrow}e_1 \,\text{in}\, e_2)$, without the side-condition $x \notin \tau_2$ (i.e. x is not among the free variables of τ_2).

2.2 A set-theoretic semantics of EL_T

In this section we specialize the categorical semantics of EL_T to the category **Set** of sets. Since **Set** is a topos, we can take as typed predicate calculus HOL with dependent types:

- a context Γ is interpreted by a set $I \in$ **Set**
- a type $\Gamma \vdash \tau$ type is interpreted by an I-indexed family $X = \langle X_i | i \in I\rangle$ of sets
- a term $\Gamma \vdash e{:}\tau$ is interpreted by an I-indexed family $x = \langle x_i \in X_i | i \in I\rangle$ of elements
- a formula $\Gamma \vdash \phi$ prop is interpreted by a subset of I.

To extend the interpretation of HOL with dependent types to ML_T, one need a fibered monad on **Set** fibered over itself. But in **Set** fibered monads *correspond* to ordinary monads, namely a monad (T, η, μ) induces the fibered monad $T_I(\langle X_i | i \in I\rangle) = \langle T(X_i) | i \in I\rangle$ (and any fibered monad is obtained in this way). Therefore, given a monad (T, η, μ) over **Set** the interpretation can be extended to ML_T as follows (compare with [12]):

$$\frac{[\![\Gamma \vdash \tau \text{ type}]\!]}{[\![\Gamma \vdash T\tau \text{ type}]\!]} \quad \begin{array}{l} = \langle X_i | i \in I\rangle \\ = \langle TX_i | i \in I\rangle \end{array}$$

$$\frac{[\![\Gamma \vdash e{:}\tau]\!]}{[\![\Gamma \vdash [e]{:}T\tau]\!]} \quad \begin{array}{l} = \langle x_i \in X_i | i \in I\rangle \\ = \langle \eta_{X_i}(x_i) \in TX_i | i \in I\rangle \end{array}$$

$$\frac{[\![\Gamma \vdash e_1{:}T\tau_1]\!] \quad [\![\Gamma, x{:}\tau_1 \vdash e_2{:}T\tau_2]\!]}{[\![\Gamma \vdash (\text{let}\, x{\Leftarrow}e_1 \,\text{in}\, e_2){:}T\tau_2]\!]} \quad \begin{array}{l} = \langle c_i \in TX_i | i \in I\rangle \\ = \langle f_i x \in TY_i | i \in I, x \in X_i\rangle \\ = \langle f_i^*(c_i) \in TY_i | i \in I\rangle \end{array}$$

where $f^* = (Tf)\,;\mu_Y{:}TX \to TY$ when $f{:}X \to TY$.

In HOL one can define evaluation predicate and possibility in terms of necessity (see [14]):

- $\Gamma \vdash (e{\Downarrow}v) \stackrel{\Delta}{\equiv} \forall X{:}\Omega^\tau.([x{\Leftarrow}e]X(x)) \supset X(v)$

- $\Gamma \vdash (\langle x \Leftarrow e \rangle \phi) \stackrel{\Delta}{\equiv} \forall w\colon \Omega.([x \Leftarrow e](\phi \supset w)) \supset w$

where Ω is the type of truth values. In classical logic possibility is simply $\neg([x \Leftarrow e]\neg\phi)$.

In FOL over ML_T **with subset types** necessity is definable as follow (see Theorem 7.4):

- $\Gamma \vdash ([x \Leftarrow e]\phi) \stackrel{\Delta}{\equiv} \exists c\colon T(\{x\colon \tau|\phi\}).e =_{T\tau} Tic$, where $\Gamma, x\colon \{x\colon \tau|\phi\} \vdash i(x)\colon \tau$ is the inclusion of $\{x\colon \tau|\phi\}$ into τ and Tic stands for the term let $x \Leftarrow c$ in $[i(x)]$.

Example 2.2 In the following examples of monad (T, η, μ) over **Set** we give TA and the meaning of $c \Downarrow a$, $[x \Leftarrow c]p(i, x)$ and $\langle x \Leftarrow c \rangle p(i, x)$, where $i\colon I$, $a\colon A_i$, $c\colon T(A_i)$ and p is a predicate over $\Sigma i\colon I.A_i$.

- **exceptions** $TA = A + E$, E set of exceptions

  $c \Downarrow a$ iff $c = [a]$, i.e. $c = in_1(a)$

  $[x \Leftarrow c]p(i, x)$ iff $\forall x\colon A_i.c \Downarrow x \supset p(i, x)$

  $\langle x \Leftarrow c \rangle p(i, x)$ iff $\exists x\colon A_i.c \Downarrow x \wedge p(i, x)$;

- **non-determinism** $TA = \mathcal{P}_{fin}(A)$

  $c \Downarrow a$ iff $a \in c$, while the modalities are defined *like* for exceptions;

- **side-effects** $TA = \mathcal{P}_{fin}(A \times S)^S$, S set of states

  $c \Downarrow a$ iff $\exists s, s'\colon S.\langle a, s' \rangle \in cs$, while the modalities are defined *like* for exceptions;

- **resumptions** $TA = \mu X.\mathcal{P}_{fin}(A + X)$, i.e. the set of finite trees with leaves labeled by elements of A

  $c \Downarrow a$ iff "at least one leaf of c is labeled by a", while the modalities are defined *like* for exceptions;

- **continuations** $TA = R^{(R^A)}$, R set of results

  $[x \Leftarrow c]p(i, x)$ iff $\forall k, k'\colon R^{A_i}.(\forall x\colon A_i.p(i, x) \supset kx = k'x) \supset ck = ck'$, i.e. "$ck = ck'$ if $k, k'\colon R^{A_i}$ agree on $\{x\colon A_i|p(i, x)\}$"

  $\langle x \Leftarrow c \rangle p(i, x)$ iff $\neg[x \Leftarrow c]\neg p(i, x)$, i.e. "there exist k and k' that may differ only on $\{x\colon A_i|p(i, x)\}$ and s.t. $ck \neq ck'$"

  $c \Downarrow a$ iff $\langle x \Leftarrow c \rangle(x =_{A_i} a)$, i.e. "there exist k and k' differing only on a and s.t. $ck \neq ck'$"

3 Counter-examples

The semantics of necessity (for EL_T without dependent types) in [14] relies on additional properties of strong monads. If

- $\mathcal{C}$ is a category with finite products, used for interpreting types and terms
- $\mathcal{M}$ is a class of monos stable under pullback, used for interpreting formulas
- (T, t) is a strong endofunctor, used for interpreting computational types, s.t. T preserves monos in $\mathcal{M}$, i.e. $m \in \mathcal{M}$ implies $Tm \in \mathcal{M}$

then the interpretation of necessity is

$$\frac{[\![\Gamma, x\colon \tau \vdash \phi \text{ prop}]\!]}{[\![\Gamma, c\colon T\tau \vdash [x \Leftarrow c]\phi \text{ prop}]\!]} \quad \begin{array}{l} = [m] \in \mathcal{M}[\Gamma \times \tau] \\ = \mathrm{t}^*_{\Gamma,\tau}[Tm] \in \mathcal{M}[\Gamma \times T\tau] \end{array}$$

where $[m]$ is the subobject corresponding to the mono m, and $f^*[m]$ is the inverse image of $[m] \in \mathcal{M}[Y]$ along $f\colon X \to Y$.

Remark 3.1 One further assumption on (T, t) is needed to show that the interpretation *commutes with substitution* of variables in Γ (see [14]).

In a category $\mathcal{C}$ with finite limits, there are two obvious choices for $\mathcal{M}$: the class of all monos, and the class of regular monos (i.e. equalizers of parallel pairs) closed under composition. The first choice is *maximal*, the second is *minimal* at least for interpreting equality predicates, and in a topos the two choices coincide. In the category **Cpo** of cpos (i.e. posets with lubs of ω-chains) regular monos correspond to inductive subsets (i.e. subsets of a cpo closed under lubs of ω-chains).

Example 3.2 In **Set** every monad (but not every endofunctor) preserves monos, and for the monads considered in Example 2.2 the two interpretations of necessity agree (see Theorem 6.4).

Most of the strong monads over **Cpo** used in Denotational Semantics satisfy the additional properties requested in [14], when $\mathcal{M}$ is the class of inductive subsets. However, there are two important counter-examples:

- Σ-continuations $TX = \Sigma^{(\Sigma^X)}$, where Σ is the cpo which classifies Scott-open subsets;
- Plotkin's powerdomain $TX = P_p(X_\perp)$, where $P_p(X)$ is the free binary semi-lattice over X.

Remark 3.3 $TX = 2^{(2^X)}$, where 2 is the flat poset with two elements, is already a counter-example (even in the category of posets). But it is not used in Denotational Semantics, as TX does not have a least element. Other counter-examples, based on monads for *dynamic allocation* over a category of parametric functors (see [15, 19]), are too complex to be described here.

Proposition 3.4 *In* **Cpo** *there is a regular mono m s.t. Tm is not monic, when T is the monad of Σ-continuations or Plotkin's powerdomain.*

Proof Let $L \cong (1+L)_\perp$ be the cpo of lazy natural numbers, whose elements are: $s^n(0)$, $s^n(\perp)$ and ∞. The order on L is generated by $s^n(\perp) < s^{n+1}(\perp), s^n(0), \infty$ for every $n \in N$. The mono m corresponds to the inductive subset M of maximal elements $\{s^n(0) | n \in N\} \cup \{\infty\}$. m can be described as the equalizer of $s, s' : L \to L$, where $s(s^n(\perp)) = s^{n+1}(\perp)$, $s(s^n(0)) = s'(s^n(0)) = s^{n+1}(0)$ and are the identity otherwise.

Consider the case $TX = \Sigma^{(\Sigma^X)}$ first. Since Σ^X is isomorphic to the lattice of open subsets of X, then the elements of TX may be thought as open subsets of open subsets of X. Since M is a flat cpo every subset of M is open. Let $O \stackrel{\Delta}{=} \{U \subseteq M | \exists n{:}N.s^n(0) \in U\}$ and $O' \stackrel{\Delta}{=} O \cup \{\{\infty\}\}$, then O and O' are distinct elements of TM. However, Tm maps O and O' to the same element in TX, namely $O'' \stackrel{\Delta}{=} \{V \in \Sigma^L | \exists n{:}N.s^n(0) \in V\}$. In fact, when V is an open subset of L and $\infty \in V$, then $s^n(0) \in V$ for some $n \in N$.

To prove that Tm is not monic, when $TX = P_p(X_\perp)$, one has to use the explicit description of Plotkin's powerdomain available for SFPs. We skip the details. ∎

4 A new semantics of EL_T

In this section we consider a semantics for the necessity modality of EL_T, which is based on different assumptions about $\mathcal{C}$, $\mathcal{M}$ and T. We briefly recall the necessary background about fibrations and factorization systems, and refer to [1, 9] and [2, 5] for more details.

Definition 4.1 (Fibrations) *Given $p: \mathcal{C} \to \mathcal{B}$, we say that*

- *$f \in \mathcal{C}(Y, X)$ is p-**cartesian** $\stackrel{\Delta}{\iff}$ for every $g \in \mathcal{C}(Z, X)$ and $h' \in \mathcal{B}(pZ, pY)$ s.t. $pg = h' ; (pf)$ exists unique $h \in \mathcal{C}(Z, Y)$ s.t. $g = h ; f$ and $h' = ph$*

- *$p: \mathcal{C} \to \mathcal{B}$ is a **fibration** (over $\mathcal{B}$) $\stackrel{\Delta}{\iff}$ for every $X \in \mathcal{C}$ and $f' \in \mathcal{B}(Y', pX)$ exists $f \in \mathcal{C}(Y, X)$ s.t. f is p-cartesian and $f' = pf$.*

Fibrations over $\mathcal{B}$ form a 2-category $Fib(\mathcal{B})$ with fibered functors as 1-cells and vertical natural transformations as 2-cells:

- *given two fibrations $p: \mathcal{C} \to \mathcal{B}$ and $q: \mathcal{D} \to \mathcal{B}$, $F: \mathcal{C} \to \mathcal{D}$ is a **fibered functor** from p to q $\stackrel{\Delta}{\iff}$ $p = F ; q$ and F takes p-cartesian maps to q-cartesian maps*

- *given two fibered functors $F, G: \mathcal{C} \to \mathcal{D}$ from p to q, $\sigma: F \dot{\to} G$ is a **vertical natural transformation** from F to G $\stackrel{\Delta}{\iff}$ $\forall c \in \mathcal{C}. q(\sigma_c) = \mathrm{id}_{pc}$.*

Notation 4.2 Given two fibrations $p: \mathcal{C} \to \mathcal{B}$ and $q: \mathcal{D} \to \mathcal{B}$, fibered functors $F, G: p \to q$, a vertical natural transformation $\sigma: F \dot{\to} G$, and $I \in \mathcal{B}$, we write

- $\mathcal{C}_I$ for the **fiber** of p over I, i.e. the sub-category of $\mathcal{C}$ s.t. $f: X \to Y$ in $\mathcal{C}_I$ iff $pf = \mathrm{id}_I$

- F_I for the functor from $\mathcal{C}_I$ to $\mathcal{D}_I$ s.t. $F_I f = Ff$

- σ_I for the natural transformation from F_I to G_I s.t. $\sigma_{I,c} = \sigma_c$.

Remark 4.3 Since $Fib(\mathcal{B})$ is a 2-category, fibered monads and fibered adjunctions can be defined in the 2-categorical way. If $\mathcal{C}$ is a category with finite limits, then $cod: \mathcal{C}^{\to} \to \mathcal{C}$ is a fibration, called $\mathcal{C}$ **fibered over itself**, and the fiber over I is denoted by $\mathcal{C}/I$. Moreover, a class $\mathcal{M}$ of monos in $\mathcal{C}$ stable under pullback induces a *posetal* fibration $\mathcal{M}$ over $\mathcal{C}$, s.t. the fiber $\mathcal{M}[I]$ over I is the poset of subobjects of I represented by monos in $\mathcal{M}$.

Remark 4.4 $\mathcal{C}$ and $\mathcal{M}$ as above can be used for interpreting a predicate calculus with dependent types:

- a context Γ is interpreted by an object $I \in \mathcal{C}$

- a type $\Gamma \vdash \tau$ type is interpreted by a $\pi: X \to I$, i.e. an object $\pi \in \mathcal{C}/I$, and $\Gamma, x{:}\tau$ is interpreted by X

- a term $\Gamma \vdash e{:}\tau$ is interpreted by a section x of π, i.e. an $x: I \to X$ in $\mathcal{C}$ s.t. $x ; \pi = \mathrm{id}_I$, or equivalently an element $x: 1 \to \pi$ of π in $\mathcal{C}/I$

- a formula $\Gamma \vdash \phi$ prop is interpreted by a subobject $[m] \in \mathcal{M}[I]$.

There are **subtle problems** in defining such an interpretation, due to the lack of *canonical choices*. A simple way out is to replace the fibration *cod* with a category with attributes $\mathcal{D} \underset{\pi}{\overset{G}{\underset{\Downarrow p}{\rightrightarrows}}} \mathcal{C}$, where π is a discrete split fibration and p is a cartesian natural transformation (see [11]), so that:

- Γ is interpreted by an object $I \in \mathcal{C}$

- $\Gamma \vdash \tau$ type is interpreted by an object $X \in \mathcal{D}_I$, and $\Gamma, x{:}\,\tau$ is interpreted by $G(X)$
- $\Gamma \vdash e{:}\,\tau$ is interpreted by a section x of p_X.

Further *canonical choices* are needed for interpreting subset types $\{x{:}\,\tau|\phi\}$. For the sake of presentation, we ignore canonical choices as far as possible. Nevertheless, all examples of $\mathcal{C}$ we consider admit simple canonical choices.

Definition 4.5 (Factorization systems) *Given a category $\mathcal{C}$ with pullbacks, we say that a class $\mathcal{D}$ of morphisms is* **stable** $\overset{\Delta}{\iff}$ $d \in \mathcal{D}$ *and*

$$\begin{tikzcd} X' \arrow[r, "f"] & X \\ Y' \arrow[u, "d'"] \arrow[r, "f'"'] \arrow[ur, phantom, very near start, "\urcorner"] & Y \arrow[u, "d"'] \end{tikzcd} \quad \textit{implies } d' \in \mathcal{D};$$

$(\mathcal{E}, \mathcal{M})$ is a **factorization system** $\overset{\Delta}{\iff}$

- *$\mathcal{E}$ and $\mathcal{M}$ are classes of morphisms containing all isos and closed under composition,*
- *for every morphism f in $\mathcal{C}$ exist $e \in \mathcal{E}$ and $m \in \mathcal{M}$ s.t. $f = e\,;m$, and the factorization (e, m) is unique up to isomorphism.*

Moreover, $(\mathcal{E}, \mathcal{M})$ is called **stable** *if $\mathcal{E}$ and $\mathcal{M}$ are stable, and* **proper** *if each $e \in \mathcal{E}$ is epi and each $m \in \mathcal{M}$ is mono.*

Notation 4.6 Given a morphism $f{:}\,Y \to X$ in $\mathcal{C}$ with proper factorization (e, m), we write $img_{\mathcal{M}}(f)$ for the subobject $[m] \in \mathcal{M}[X]$.

Remark 4.7 In a factorization system the class $\mathcal{M}$ is always stable. Moreover, $\mathcal{E}$ can be recovered from $\mathcal{M}$ (and conversely). Therefore a class of monos $\mathcal{M}$ identifies at most one (stable and) proper factorization system $(\mathcal{E}, \mathcal{M})$.

The new semantics of EL_T (with dependent types) relies on the following assumptions, if

- $\mathcal{C}$ is a category with finite limits
- $(\mathcal{E}, \mathcal{M})$ is a stable and proper factorization system
- T is a fibered endofunctor over $cod{:}\,\mathcal{C}^{\to} \to \mathcal{C}$

then the interpretation of computational types and necessity is

$$\frac{[\![\Gamma \vdash \tau \text{ type}]\!]}{[\![\Gamma \vdash T\tau \text{ type}]\!]} \quad \begin{array}{l} = (\pi{:}\,\Gamma, \tau \to \Gamma) \in \mathcal{C}/\Gamma \\ = (T_\Gamma \pi{:}\,\Gamma, T\tau \to \Gamma) \in \mathcal{C}/\Gamma \end{array}$$

$$\frac{[\![\Gamma, x{:}\,\tau \vdash \phi \text{ prop}]\!]}{[\![\Gamma, c{:}\,T\tau \vdash [x{\Leftarrow}c]\phi \text{ prop}]\!]} \quad \begin{array}{l} = [m] \in \mathcal{M}[\Gamma, \tau] \\ = img_{\mathcal{M}}(T_\Gamma m) \in \mathcal{M}[\Gamma, T\tau] \end{array}$$

where
$$\begin{tikzcd} \cdot \arrow[r, "T_\Gamma m"] \arrow[dr] & \Gamma, T\tau \arrow[d] \\ & \Gamma \end{tikzcd} \;=\; T_\Gamma \left(\begin{tikzcd} \cdot \arrow[r, "m"] \arrow[dr] & \Gamma, \tau \arrow[d, "\pi"] \\ & \Gamma \end{tikzcd} \right)$$

(One can show that the interpretation *commutes with substitution* of variables in Γ.)

Remark 4.8 The assumptions on $\mathcal{C}$ and $\mathcal{M}$ are stronger than those made in [14]. A fibered endofunctor T over $cod{:}\,\mathcal{C}^{\to} \to \mathcal{C}$ induces a strong endofunctor over $\mathcal{C}$, which may not preserve monos in $\mathcal{M}$.

Example 4.9 In **Set** there is only one stable and proper factorization system: $\mathcal{E}$ = the class of all epis and $\mathcal{M}$ = the class of all monos. In fact, this is true in any other topos.

The fiber $\mathbf{Set}/I$ is equivalent to the category $\mathbf{Set}^I$ of I-indexed families of sets.

Fibered endofunctors over $cod{:}\,\mathbf{Set}^{\to} \to \mathbf{Set}$ correspond to endofunctors over **Set**. In fact, an endofunctor T over **Set** induces the fibered endofunctor s.t. $T_I(\langle X_i | i \in I\rangle) = \langle T X_i | i \in I\rangle$.

Example 4.10 An ω-set $\underline{X}$ is a pair $(X, \Vdash_X)$, where X is a set and $\Vdash_X \subseteq N \times X$ (see [10, 17]). A morphism $f{:}\,\underline{X} \to \underline{Y}$ is a function $f{:}\,X \to Y$ s.t. $\exists e{:}\,N.\forall x{:}\,X.\forall m{:}\,N.m \Vdash_X x \supset e \cdot m \Vdash_Y f(x)$, e is called a **realizer** of f ($e \Vdash f$ for short).

In the category $\omega\mathbf{Set}$ of ω-sets there are two canonical stable and proper factorization systems: either $\mathcal{E}$ = the class of all epis and $\mathcal{M}$ = the class of regular monos, or $\mathcal{E}$ = the class of regular epis and $\mathcal{M}$ = the class of all monos. In fact, this is true in any other quasi-topos (see [24]).

The fiber $\omega\mathbf{Set}/\underline{I}$, where $\underline{I} = (I, \Vdash_I)$, is equivalent to the sub-category of $\omega\mathbf{Set}^I$ whose morphisms are realizable families, where $\langle f_i{:}\,\underline{X}_i \to \underline{Y}_i | i \in I\rangle$ is **realizable** $\overset{\Delta}{\iff}$ $\exists e{:}\,N.\forall i{:}\,I.\forall m{:}\,N.m \Vdash_I i \supset e \cdot m \Vdash f_i$.

Fibered endofunctors over $cod{:}\,\omega\mathbf{Set}^{\to} \to \omega\mathbf{Set}$ correspond to realizable endofunctors over $\omega\mathbf{Set}$, where T is **realizable** iff there exists $e \in N$ s.t. $e \cdot n \Vdash T f$

when $n \Vdash f$ (see [3]). The fibered endofunctor corresponding to a realizable endofunctor T over $\omega\mathbf{Set}$ is defined by analogy with **Set**.

Remark 4.11 In **Cpo** there is a proper factorization system: $\mathcal{E}$ = the class of all epis and $\mathcal{M}$ = the class of regular monos. However, it is not stable.

5 Synthetic Domain Theory setting

The proposed semantics for EL_T is drastically at odds with classical Domain Theory. Anyway, when looking for an expressive logic (including FOL) we are forced to leave the category of cpos, as only a fragment of FOL may be interpreted in **Cpo**. Fortunately, one may reconcile the needs of Denotational Semantics with an expressive logic by moving to the SDT setting (see [7, 23]).

Definition 5.1 *Given two fibrations $p: \mathcal{C} \to \mathcal{B}$ and $q: \mathcal{D} \to \mathcal{B}$, we say that q is a* **full reflective subfibration** *of $p \overset{\Delta}{\iff}$ there is a full and faithful fibered functor $i: q \to p$ with a fibered left adjoint $r \dashv i$.*

For our purposes the main result of SDT is:

> **there are quasi-toposes** $\mathcal{C}$ (more generally left-exact categories with a stable and proper factorization system) **with a full reflective subfibration** $q: \mathcal{D} \to \mathcal{C}$ **of** $cod: \mathcal{C}^{\to} \to \mathcal{C}$**, which has all the desiderata for Denotational Semantics**, in particular the objects of $\mathcal{D}$ *look like* (indexed families of) cpos.

We are left to show that, given a semantics for a programming language PL in the fiber $\mathcal{D}_1$ over 1, we can *construct* a fibered monad T over *cod* (so that one can interpret EL_T).

- Usually the semantics of PL can be given via translation in a typed metalanguage ML for Denotational Semantics, which has an *intended* interpretation in any *reasonable* category for Denotational Semantics, in particular in $\mathcal{D}_1$.
- According to the monadic approach (see [12]) the translation from PL to ML should factor through a metalanguage $ML_T(\Sigma)$ for computational monads. Therefore, one has a monad over $\mathcal{D}_1$, which is expressible in ML.
- Under reasonable assumptions about ML and q, a syntactic description of a monad in ML induces a fibered monad T over q.
- Finally, any fibered monad T over q extends, via the fibered adjunction $r \dashv i$, to a fibered monad $r\,;T\,;i$ over the fibration *cod*.

In conclusion, we have argued that it is possible to satisfy the assumptions for the new semantics of EL_T, whenever one starts from a semantics of PL in $\mathcal{D}_1$ given via translation into a metalanguage for Denotational Semantics.

Remark 5.2 In the literature there are several ways of constructing $\mathcal{D}$ from $\mathcal{C}$: the category of complete extensional PERs of [3, 21]), the category of complete Σ-spaces of [16], the category of Σ-replete objects (see [7, 23]).

Example 5.3 We show that the category ExP of complete extensional PERs can be turned into a full reflective sub-fibration of $\omega\mathbf{Set}$ fibered over itself.

- Consider the small category PER of partial equivalence relations (PERs) and its full reflective subcategory ExP of complete extensional PERs defined in [3].
- PER and ExP can be turned into internal categories in $\omega\mathbf{Set}$ (indeed they are full internal subcategories of $\omega\mathbf{Set}$).
- The reflection $r \dashv i$ defined in [3] can be made internal, i.e. the relevant functors and natural transformations are realizable ([3] provides all the necessary information to prove this). Therefore, ExP is a full reflective sub-category of PER in the 2-category $\mathbf{Cat}(\omega\mathbf{Set})$ of internal categories in $\omega\mathbf{Set}$.
- The externalization 2-functor (from $\mathbf{Cat}(\omega\mathbf{Set})$ to the 2-category of fibrations over $\omega\mathbf{Set}$) turns PER into a (split) fibration over $\omega\mathbf{Set}$ and ExP into a full reflective sub-fibration of PER. Since PER is a full reflective sub-fibration of $\omega\mathbf{Set}$ fibered over itself (see [8]), so is ExP.

The result cannot be improved by replacing $\omega\mathbf{Set}$ with the Effective Topos Eff, because PER is not a full reflective sub-fibration of Eff fibered over itself.

6 Relation to previous semantics

In this section we give sufficient conditions to ensure that the new semantics of necessity agrees with that proposed in [14]. But first we need some auxiliary results.

Proposition 6.1 *If $\mathcal{C}$ is a category with finite limits, then $\mathcal{C}/I$ has finite limits and any stable class $\mathcal{M}$ (of monos) on $\mathcal{C}$ induces a stable class $\mathcal{M}/I$ (of monos) on $\mathcal{C}/I$ given by*

$$\begin{tikzcd} X' \arrow[r, "m"] \arrow[dr] & X \arrow[d] \\ & I \end{tikzcd} \in \mathcal{M}/I \overset{\Delta}{\iff} m \in \mathcal{M}$$

Proposition 6.2 *If $\mathcal{C}$ is a category with finite limits and T is a fibered endofunctor over cod: $\mathcal{C}^{\rightarrow} \rightarrow \mathcal{C}$, then T_I has a tensorial strength t_I.*

Proof Given $(\alpha: A \rightarrow I)$ and $(\beta: B \rightarrow I)$ objects of $\mathcal{C}/I$, the tensorial strength $\mathrm{t}_{I,\alpha,\beta}$ is

$$\begin{tikzcd} \cdot \arrow[r, "\mathrm{t}_{I,\alpha,\beta}"] \arrow[dr, "\alpha\times T_I(\beta)"'] & \cdot \arrow[d, "T_I(\alpha\times\beta)"] \\ & I \end{tikzcd} \triangleq T\left(\begin{tikzcd} \cdot \arrow[r, "\mathrm{id}"] \arrow[d, "\pi_1"'] & \cdot \arrow[d, "\alpha\times\beta"] \\ A \arrow[r, "\alpha"'] & I \end{tikzcd}\right)$$

where the product of α and β in $\mathcal{C}/I$ is given by

$$\begin{tikzcd} \cdot \arrow[r, "\pi_2"] \arrow[d, "\pi_1"'] \arrow[dr, "\alpha\times\beta"] \arrow[dr, phantom, very near start, "\lrcorner"] & B \arrow[d, "\beta"] \\ A \arrow[r, "\alpha"'] & I \end{tikzcd}$$

∎

Remark 6.3 Since $\mathcal{C}$ is isomorphic to the fiber $\mathcal{C}/1$, then a fibered endofunctor T on *cod* induced a strong endofunctor over $\mathcal{C}$ corresponding to (T_1, t_1) .

Theorem 6.4 *If the assumptions required by the new semantics of necessity are satisfied, and moreover for every $I \in \mathcal{C}$ the functor T_I preserves pullbacks of monos in $\mathcal{M}/I$, i.e. in $\mathcal{C}/I$*

$$\begin{tikzcd} \cdot \arrow[r, "f"] & \cdot \\ \cdot \arrow[u, "n"] \arrow[r, "f'"'] \arrow[ur, phantom, very near start, "\urcorner"] & \cdot \arrow[u, "m \in \mathcal{M}/I"'] \end{tikzcd} \Longrightarrow \begin{tikzcd} \cdot \arrow[r, "T_I f"] & \cdot \\ \cdot \arrow[u, "T_I n"] \arrow[r, "T_I f'"'] \arrow[ur, phantom, very near start, "\urcorner"] & \cdot \arrow[u, "T_I m \in \mathcal{M}/I"'] \end{tikzcd}$$

then, the semantics of necessity according to [14] is defined and agrees with the new semantics.

7 Definability and sound axioms

In this section we give axioms for necessity sound w.r.t. the new semantics. These axioms are a subset of those considered in [14]. The simplest way to validate them is to express necessity in terms of standard logical constants, and then use the familiar rules satisfied by these logical constants.

Proposition 7.1 *If $\mathcal{C}$ has finite limits and $\mathcal{M}$ is a stable class of monos, then one can interpret subset types*

$$(\{|\})\ \frac{\Gamma, x{:}\,\tau \vdash \phi \text{ prop}}{\Gamma \vdash \{x{:}\,\tau|\phi\} \text{ type}} \qquad (i)\ \frac{\Gamma \vdash e{:}\,\{x{:}\,\tau|\phi\}}{\Gamma \vdash i_{x:\tau.\phi}(e){:}\,\tau}$$

$$(j)\ \frac{\Gamma \vdash e{:}\,\tau \qquad \Gamma, x{:}\,\tau \vdash \phi \text{ prop} \qquad \Gamma \vdash [e/x]\phi}{\Gamma \vdash j_{x:\tau.\phi}(e){:}\,\{x{:}\,\tau|\phi\}}$$

Proof Given $\Gamma \vdash \tau$ type and $\Gamma, x{:}\,\tau \vdash \phi$ prop, interpreted by $(\pi{:}\,\Gamma, \tau \rightarrow \Gamma) \in \mathcal{C}/\Gamma$ and $[m{:}\,\phi \hookrightarrow \Gamma, \tau] \in \mathcal{M}[\Gamma, \tau]$ respectively, then

- the interpretation of $\Gamma \vdash \{x{:}\,\tau|\phi\}$ type is given by $(m\,;\pi{:}\,\phi \rightarrow \Gamma) \in \mathcal{C}/\Gamma$
- the interpretation of $\Gamma \vdash i(e){:}\,\tau$ is given by $f\,;m$, when $\Gamma \vdash e{:}\,\{x{:}\,\tau|\phi\}$ is interpreted by $f{:}\,\Gamma \rightarrow \phi$ s.t. $f\,;(m\,;\pi) = \mathrm{id}$.
- the interpretation of $\Gamma \vdash j(e){:}\,\{x{:}\,\tau|\phi\}$ is given by the unique f' s.t. $f = f'\,;m$, when $\Gamma \vdash e{:}\,\tau$ is interpreted by f s.t. $f\,;\pi = \mathrm{id}$ and $f^*[m] = [\mathrm{id}_\Gamma]$.

∎

Remark 7.2 The term constructors $i_{x:\tau.\phi}$ and $j_{x:\tau.\phi}$ play the role of explicit coercions, and it is often convenient to leave them implicit.

A correct interpretation of subset types relies on *canonical choices* of representative for each subobject $[m] \in \mathcal{M}[I]$. More precisely, we need a category with attributes $\begin{tikzcd} \mathcal{D} \arrow[r, shift left=2, "G"] \arrow[r, shift right=2, "\pi"'] \arrow[r, phantom, "\Downarrow p"] & \mathcal{C} \end{tikzcd}$, (see Remark 4.4) with units and sums (see [11]) and a fibered embedding $\iota{:}\,|\mathcal{M}| \rightarrow \pi$ s.t. $[m] = [p_{\iota([m])}]$. Intuitively, ι maps the interpretation of ϕ to the interpretation of the subset $\{*{:}\,1|\phi\}$ of the unit type 1. In the examples based on sets and ω-sets, ι is easy to define.

Proposition 7.3 *If $\mathcal{C}$ has finite limits and $(\mathcal{E}, \mathcal{M})$ is a stable and proper factorization system, then one can* *interpret equality and existential quantification*

$$(=)\ \frac{\begin{array}{c}\Gamma \vdash \tau \text{ type}\\ \Gamma \vdash e_i{:}\,\tau\end{array}}{\Gamma \vdash e_1 =_\tau e_2 \text{ prop}} \quad (\exists)\ \frac{\Gamma, x{:}\,\tau \vdash \phi \text{ prop}}{\Gamma \vdash \exists x{:}\,\tau.\phi \text{ prop}}$$

Proof Given a type $\Gamma \vdash \tau$ type whose interpretation is $(\pi{:}\,\Gamma,\tau \to \Gamma) \in \mathcal{C}/\Gamma$, the interpretation of equality over τ (according to Categorical Logic) is the diagonal $\Delta{:}\,\pi \hookrightarrow \pi \times_\Gamma \pi$ in $\mathcal{C}/\Gamma$. We need to show that $\Delta \in \mathcal{M}$:

- Δ is a regular mono in $\mathcal{C}$, e.g. is the equalizer of the two projections $\pi_1, \pi_2{:}\,\pi\times_\Gamma\pi \to \pi$ in $\mathcal{C}/\Gamma$

$$\begin{tikzcd}
\Gamma,\tau \arrow[dr, "\Delta"] \arrow[drr, "\mathrm{id}"] \arrow[ddr, "\mathrm{id}"'] & & \\
& \Gamma,\tau\times\tau \arrow[r, "\pi_1"] \arrow[d, "\pi_2"] \arrow[dr, phantom, "\lrcorner", very near start] & \Gamma,\tau \arrow[d, "\pi"] \\
& \Gamma,\tau \arrow[r, "\pi"'] & \Gamma
\end{tikzcd}$$

- every regular mono is in $\mathcal{M}$. More precisely, if m is the equalizer of f and g, then $[m] = img_\mathcal{M}(m)$.

  Let (e, n) be the $(\mathcal{E},\mathcal{M})$-factorization of m, then m factors through n (as $m = e\,;n$), while n factors through m because n equalizes f and g (as $e\,;n\,;f = m\,;f = m\,;g = e\,;n\,;g$ and e is epi).

The interpretation of existential quantification along $\pi{:}\,\Gamma,\tau \to \Gamma$ is given by the left adjoint $\exists_\pi$ to $\pi^*{:}\,\mathcal{M}[\Gamma] \to \mathcal{M}[\Gamma,\tau]$, which must also satisfy the Beck-Chevalley. It is easy to show that:

- $\exists_\pi([m]) = img_\mathcal{M}(m\,;\pi)$
- the Beck-Chevalley condition holds, since the factorization system $(\mathcal{E},\mathcal{M})$ is stable.

∎

Theorem 7.4 *If the assumptions required for interpreting necessity according to the new semantics are satisfied, then*

$$(\Box)\ \frac{\Gamma \vdash e{:}\,T\tau \qquad \Gamma, x{:}\,\tau \vdash \phi \text{ prop}}{\Gamma \vdash [x{\Leftarrow}e]\phi \iff \exists c'{:}\,T(\{x{:}\,\tau|\phi\}).e =_{T\tau} Tic'}$$

where Tic' stands for the term $\mathrm{let}\ x'{\Leftarrow}c'\ \mathrm{in}\ [i(x')]$.

Proof Let $[\![\Gamma \vdash \tau \text{ type}]\!] = \pi{:}\,\Gamma,\tau \to \Gamma$ and $[\![\Gamma, x{:}\,\tau \vdash \phi \text{ prop}]\!] = [m{:}\,\phi \hookrightarrow \Gamma,\tau]$, then

- $[\![\Gamma \vdash \{x{:}\,\tau|\phi\} \text{ type}]\!] = (m\,;\pi{:}\,\phi \to \Gamma)$
- $[\![\Gamma, c{:}\,T\tau \vdash [x{\Leftarrow}c]\phi \text{ prop}]\!] = img_\mathcal{M}(T_\Gamma m)$
- $[\![\Gamma, x{:}\,\{x{:}\,\tau|\phi\} \vdash i(x){:}\,\tau]\!] = i$, where

$$\begin{tikzcd}
\phi \arrow[dr, "i"] \arrow[drr, "\mathrm{id}"] \arrow[ddr, "m"'] & & \\
& \cdot \arrow[r, "p_1"] \arrow[d, "p_2"] \arrow[dr, phantom, "\lrcorner", very near start] & \phi \arrow[d, "m\,;\pi"] \\
& \Gamma,\tau \arrow[r, "\pi"'] & \Gamma
\end{tikzcd}$$

- $[\![\Gamma, c{:}\,T\tau, c'{:}\,T(\{x{:}\,\tau|\phi\}) \vdash c = Tic' \text{ prop}]\!] = [n]$, where

$$\begin{tikzcd}
\cdot \arrow[dr, "n"] \arrow[drr, "\mathrm{id}"] \arrow[ddr, "T_\Gamma m"'] & & \\
& \cdot \arrow[r, "q_1"] \arrow[d, "q_2"] \arrow[dr, phantom, "\lrcorner", very near start] & \cdot \arrow[d, "T_\Gamma(m\,;\pi)"] \\
& \Gamma,T\tau \arrow[r, "T_\Gamma\pi"'] & \Gamma
\end{tikzcd}$$

- $[\![\Gamma, c{:}\,T\tau \vdash \exists c'{:}\,T(\{x{:}\,\tau|\phi\}).c = Tic' \text{ prop}]\!] = \exists_{q_2}([n])$

Therefore, we must prove that $\exists_{q_2}([n]) = img_\mathcal{M}(T_\Gamma m)$:

- $\exists_{q_2}([n]) =$ by definition of $\exists_f$ (see Proposition 7.3)
- $img_\mathcal{M}(n\,;q_2) =$ by definition of n
- $img_\mathcal{M}(T_\Gamma m)$

∎

With the characterization of necessity given above, it is quite easy to derive rules for necessity from those for existential quantification, equality predicate, subset and computational types.

Corollary 7.5 *The following rules are sound w.r.t. the new semantics of EL_T:*

- $(\Box\text{-}\top^*)\ \dfrac{\Gamma \vdash \tau \text{ type}}{\Gamma, c{:}\,T\tau \vdash \top \Longrightarrow [x{\Leftarrow}c]\top}$

- $(\Box\text{-}\Longrightarrow)$ $\dfrac{\Gamma \vdash e{:}\tau \quad \Gamma, x{:}\tau \vdash \Phi, \phi \Longrightarrow \psi}{\Gamma \vdash \Phi, [x{\Leftarrow}e]\phi \Longrightarrow [x{\Leftarrow}e]\psi}\; x \notin \Phi$

- $(\Box\text{-}T)$ $\dfrac{\Gamma, x{:}\tau_1 \vdash e{:}\tau_2 \qquad \Gamma, x{:}\tau_2 \vdash \phi \text{ prop} \quad \Gamma, c{:}T\tau_1 \vdash \Phi \Longrightarrow [x{\Leftarrow}c]([e/x]\phi)}{\Gamma, c{:}T\tau_1 \vdash \Phi \Longrightarrow [x{\Leftarrow}(\text{let}\, x{\Leftarrow}c\, \text{in}\, [e])]\phi}$

- $(\Box\text{-}=)$ $\dfrac{\Gamma \vdash e{:}T\tau_1 \qquad \Gamma, x{:}\tau_1 \vdash e_i{:}\tau_2 \quad \Gamma \vdash \Phi \Longrightarrow ([x{\Leftarrow}e]e_1 =_{\tau_2} e_2)}{\Gamma \vdash \Phi \Longrightarrow T(x.e_1)e =_{T\tau_2} T(x.e_2)e}\; x \notin \tau_2$

  where $T(x.e_i)e$ *stand for the term* $\text{let}\, x{\Leftarrow}e\, \text{in}\, [e_i]$

- $(\Box\text{-}\eta)$ $\dfrac{\Gamma, x{:}\tau \vdash \phi \text{ prop}}{\Gamma, x{:}\tau \vdash \phi \Longrightarrow [x{\Leftarrow}[x]]\phi}$

Proof As a example we derive the rule $(\Box\text{-}\Longrightarrow)$. First, we restrict to the case $\Phi = \emptyset$, by applying the bi-rule

$$(\vdash)\ \frac{\Gamma \vdash \phi, \Phi \Longrightarrow \psi}{\Gamma, x{:}\{x{:}1|\phi\} \vdash \Phi \Longrightarrow \psi}\; x \notin \Gamma$$

where 1 is the unit type.

Therefore, from $\Gamma, x{:}\tau \vdash \phi_1 \Longrightarrow \phi_2$ we must derive $\Gamma \vdash \exists c_1{:}T\tau_1.e = Ti_1c_1 \Longrightarrow \exists c_2{:}T\tau_2.e = Ti_2c_2$, where τ_i and i_i stand for $\{x{:}\tau|\phi_i\}$ and $i_{x:\tau.\phi_i}$. For simplicity, let us assume that $\Gamma = \emptyset$:

- $x{:}\tau_1 \vdash \phi_1(i_1(x))$, by rules for i

- $x{:}\tau_1 \vdash \phi_2(i_1(x))$, by the assumption

- $x{:}\tau_1 \vdash i_1(x) = i_2(j_2(x))$, where j_2 stand for $j_{x:\tau.\phi_2}$, by rules for j

- $c_1{:}T\tau_1 \vdash\Longrightarrow Ti_1c_1 = Ti_2(T(x{:}\tau_1.j_2(x))c_1)$, by equational rules for computational types

- $c_1{:}T\tau_1 \vdash e = Ti_1c_1 \Longrightarrow e = Ti_2(T(x{:}\tau_1.j_2(x))c_1)$

- $c_1{:}T\tau_1 \vdash e = Ti_1c_1 \Longrightarrow \exists c_2{:}T\tau_2.e = Ti_2c_2$

- $\vdash \exists c_1{:}T\tau_1.e = Ti_1c_1 \Longrightarrow \exists c_2{:}T\tau_2.e = Ti_2c_2$

∎

Remark 7.6 Commutativity of necessity with conjunction $[x{\Leftarrow}c](\phi_1 \wedge \phi_2) \Longleftrightarrow ([x{\Leftarrow}c]\phi_1) \wedge ([x{\Leftarrow}c]\phi_2)$ fails in the new semantics.

We don't know any model which does not satisfy $[y{\Leftarrow}c][x{\Leftarrow}y]\phi \Longrightarrow [x{\Leftarrow}(\text{let}\, y{\Leftarrow}c\, \text{in}\, y)]\phi$.

8 Conclusions

Most program logics are tide up to specific programming languages, look rather ad hoc, and *standard* logical axioms may be unsound. This state of affairs is not very satisfactory, since it is time-consuming (even for logicians) to get acquainted with new logics and develop good proof strategies. It would be nice if *standard* predicate logic were enough for reasoning about programs (including non-functional ones), and if proofs could be kept *simple* with a clever choice of language and axioms.

Synthetic Domain Theory has given an important contribution towards this ideal situation, by showing that the slogan "domains are sets" makes sense. Based on this, we show that a *reasonable* semantics of a programming language PL in a model $\mathcal{C}$ of SDT induces an interpretation of computational types over the whole $\mathcal{C}$, and therefore an interpretation of EL_T (definable in *standard* predicate logic).

In a companion paper (see [13]) we give evidence of the expressiveness of EL_T, by showing that the program logic $VTLoE$ (a Dynamic Predicate Logic for an untyped call-by-value functional language with references introduced in [6]) has a simple translation into EL_T and most of its axioms are formally derivable in EL_T. At a category-theoretic level, there seems to be close relations between the new semantics of EL_T and the topos-theoretic semantics of Modal Logics proposed by Reyes (see [20]).

Acknowledgements

I would like to thank A. Kock for suggesting to use indexed monads, M. Hyland, P. Rosolini, T. Streicher, P. Taylor for discussions on SDT related issues, B. Jacobs and J. Power for advice on fibrations, and P. Cenciarelli for comments.

References

[1] J. Benabou. Fibred categories and the foundation of naive category theory. *Journal of Symbolic Logic*, 50, 1985.

[2] P. Freyd and G.M. Kelly. Categories of continuous functors, i. *Journal of Pure and Applied Algebra*, 2, 1972.

[3] P. Freyd, P. Mulry, G. Rosolini, and D. Scott. Extensional pers. In *5th LICS Conf.* IEEE, 1990.

[4] M.J.C. Gordon, R. Milner, and C.P. Wadsworth. *Edinburgh LCF: A Mechanized Logic of Computation*, volume 78 of *LNCS*. Springer Verlag, 1979.

[5] H. Herrlich and E. Strecker. *Category Theory*. Heldermann Verlag, 1979.

[6] F. Honsell, I.A. Mason, S.F. Smith, and C. Talcott. A variable typed logic of effects. *Information and Computation*, to appear.

[7] J.M.E. Hyland. First steps in synthetic domain theory. In A. Carboni, C. Pedicchio, and G. Rosolini, editors, *Conference on Category Theory '90*, volume 1488 of *LNM*. Springer Verlag, 1991.

[8] J.M.E. Hyland, E.P. Robinson, and G. Rosolini. The discrete objects in the effective topos. *Proc. London Math. Soc.*, 60, 1990.

[9] B. Jacobs. *Categorical Type Theory*. PhD thesis, University of Nijmegen, 1991.

[10] G. Longo and E. Moggi. Constructive natural deduction and its modest interpretation. *Mathematical Structures in Computer Science*, 1, 1991.

[11] E. Moggi. A category-theoretic account of program modules. *Mathematical Structures in Computer Science*, 1, 1991.

[12] E. Moggi. Notions of computation and monads. *Information and Computation*, 93(1), 1991.

[13] E. Moggi. Representing $VTLoE$ in evaluation logic. Available via FTP from theory.doc.ic.ac.uk as /theory/papers/Moggi/elt-vtloe.dvi.Z, 1994.

[14] E. Moggi. A semantics for evaluation logic. *Fundamenta Informaticae*, 22(1/2), 1994.

[15] P.W. O'Hearn and R.D. Tennent. Relational parametricity and local variables. In *20th POPL*. ACM, 1993.

[16] W. Phoa. Effective domains and intrinsic structure. In *5th LICS Conf.* IEEE, 1990.

[17] W. Phoa. An introduction to fibrations, topos theory, the effective topos and modest sets. Technical Report ECS-LFCS-92-208, Edinburgh Univ., Dept. of Comp. Sci., 1992.

[18] A.M. Pitts. Evaluation logic. In G. Birtwistle, editor, *IVth Higher Order Workshop, Banff 1990*, volume 283 of *Workshops in Computing*. Springer Verlag, 1991.

[19] A.M. Pitts and I.D.B. Stark. Observable properties of higher order functions that dynamically create local names, or: What's *new*? In *Math. Found. of Comp. Sci. '93*, volume 711 of *LNCS*. Springer Verlag, 1993.

[20] G.E. Reyes and H. Zolfaghari. Topos-theoretic approaches to modalities. In A. Carboni, C. Pedicchio, and G. Rosolini, editors, *Conference on Category Theory '90*, volume 1488 of *LNM*. Springer Verlag, 1991.

[21] G. Rosolini. An exper model for quest. In S. Brookes, editor, *Mathematical Foundations of Programming Semantics*, LNCS. Springer-Verlag, 1992.

[22] D.S. Scott. A type-theoretic alternative to CUCH, ISWIM, OWHY. Oxford notes, 1969.

[23] P. Taylor. The fixed point property in synthetic domain theory. In *6th LICS Conf.* IEEE, 1991.

[24] O. Wyler. Are there topoi in topology? In *Categorical Topology*, volume 540 of *LNM*. Springer Verlag, 1976.