A Semantics for Evaluation Logic

E. Moggi*
DISI, Univ. of Genova, Italy
moggi@disi.unige.it

Abstract

This paper proposes a *topos-theoretic* semantic for the modalities and evaluation predicate of Pitts' Evaluation Logic, and introduces several predicate calculi (ranging from Horn sequents to Higher Order Logic), which are sound and complete w.r.t. natural classes of models. It is shown (by examples) that many computational monads satisfy the additional properties required by the proposed semantics.

Introduction

Evaluation logic EL_T is a *typed predicate logic* (see [CP92, Pit91]) based on the metalanguage for computational monads ML_T (a typed calculus introduced in [Mog91]), which permits statements about evaluation of programs to values by the use of *evaluation modalities*. In particular, EL_T might be used for axiomatising computation-related properties of a monad or devising computationally adequate theories (see [Pit91]), and it appears useful when addressing the question of logical principles for reasoning about the behaviour of programs. In the best intentions EL_T should provide a *uniform framework* for presenting programming languages and program logics (as in Scott's and Milner's LCF approach [GMW79], one should view programs as terms and assertions as formulas), which hopefully will support a *modular approach* to their description.
This paper addresses the issue of finding general logical principles for evaluation modalities by following the same methodology used to find the equational axioms for ML_T, i.e. first the categorical semantics, then sound and complete formal systems (see the introduction of [Mog91]). This issue is addressed also in [Pit91], but our approach differs from that by Pitts mainly in the categorical semantics of the evaluation modalities. In fact, our interpretation is uniquely determined by a strong monad T (but it is defined only if T satisfies some additional properties), while Pitts' interpretation depend on some additional structure (which has to be found). However, there are important differences also at the level of logical principles, due to the fact that Pitts allows *non-standard* semantics for formulas (e.g. when formulas over A are interpreted by subsets of $A \times S$, rather than subsets of A), while we want to stick to *standard* semantics. By means of examples, we will show that there is no need for allowing *non-standard* semantics of formulas, and that in our semantics the interpretation of evaluation modalities is "*almost* always the expected one".
The paper is organised as follows. Section 1 explains the intuition about EL_T, presents a set-theoretic semantics with few simple examples, and discusses alternative semantics. Section 2 introduces several formal systems, that will be proved sound and complete w.r.t. suitable classes of categorical models, and establishes some definability results (see Theorem 2.12 and 2.13). Section 3 recalls the hyperdoctrine and topos-theoretic approaches to interpreting typed predicate logics. Section 4 defines our topos-theoretic semantics for (the necessity modality of) EL_T, establishes soundness results for those rules of Section 2 involving necessity (see Theorem 4.8), and presents further examples of our categorical semantics for EL_T (see Example 4.9,e 4.10, 4.11 and 4.12). Section 5 proves several completeness results w.r.t. our topos-theoretic semantics (see Theorem 5.1),

*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).

and in doing so it relates our semantics of EL_T to that in [Pit91] (the three constructions described in this section may be useful to establish similar completeness results). Most of the proofs are given in the appendix.

1 Informal semantics of EL_T

At this stage, we leave vague the language of EL_T (and ML_T), and describe only its key features, but when needed we will be more specific. For applications it is useful to extend EL_T with additional type constructors, operations and logical constants; however, in some cases this might require a careful analysis of the interactions between EL_T and the additional features.

1.1 Key features of ML_T and EL_T

The syntactic categories of EL_T are types, terms and formulas (as in many-sorted first order logic). One may consider also kinds (as in higher order lambda calculus) or proofs, but for our purposes they are unnecessary.

- We write $\vdash \tau$ type for the judgement "τ is a well-formed type". Types are closed w.r.t. the formation rule

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

  $T\tau$ is called a **computational type**, and terms of type $T\tau$ should be thought of as programs which return values of type τ. We do not consider dependent types, since their interaction with computational types is problematic.

  A well-formed context Γ is simply a sequence of the form $x_1{:}\tau_1, \ldots, x_n{:}\tau_n$, where x_i are variables and τ_i are well-formed terms.

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

  $$(\text{lift})\ \frac{\Gamma \vdash e{:}\tau}{\Gamma \vdash [e]_T{:}T\tau} \qquad (\text{let})\ \frac{\Gamma \vdash e_1{:}T\tau_1 \qquad \Gamma, x{:}\tau_1 \vdash e_2{:}T\tau_2}{\Gamma \vdash (\text{let}_T\ x{\Leftarrow}e_1\ \text{in}\ e_2){:}T\tau_2}$$

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

- We write $\Gamma \vdash \phi$ prop for the judgement "ϕ is a well-formed formula in context Γ". Formulas are closed w.r.t. 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.

1.2 A set-theoretic semantics of EL_T

In this section we specialise our categorical semantics for evaluation modalities and evaluation predicate to the category **Set** of sets. In particular, formulas over A are interpreted by subsets of A. For interpreting computational types and terms (lift and let) of ML_T, we must fix a strong monad $(T, \mathrm{t}, \eta, \mu)$ over **Set** (see [Mog91]).

For interpreting necessity, we make the extra assumption that T preserves inclusions, i.e. $A \subseteq B$ implies $TA \subseteq TB$:

$\Gamma, x{:}A \vdash \phi$ prop	$=$	$p \subseteq \Gamma \times A$
$\Gamma \vdash e{:}TA$	$=$	$f{:}\Gamma \to TA$
$\Gamma \vdash [x{\Leftarrow}e]\phi$ prop	$=$	$\langle \mathrm{id}_\Gamma, f\rangle^{-1}(\Box_{\Gamma,A}p) \subseteq \Gamma$

where $\Box_{X,A}{:}\mathcal{P}(X{\times}A) \to \mathcal{P}(X{\times}TA)$ maps a subset $p \subseteq X{\times}A$ into the inverse image $\mathrm{t}^{-1}_{\Gamma,A}(Tp)$ of $Tp \subseteq T(X{\times}A)$ along the tensorial strength $\mathrm{t}_{X,A}{:}X{\times}TA \to T(X{\times}A)$. Actually, to ensure that the substitution lemma continue to holds we must require that necessity *commutes with substitution*, i.e. for every $f{:}X \to Y$

$$\begin{array}{ccc} \mathcal{P}(Y{\times}A) & \xrightarrow{(f\times \mathrm{id}_A)^{-1}} & \mathcal{P}(X{\times}A) \\ \Big\downarrow{\scriptstyle \Box_{Y,A}} & & \Big\downarrow{\scriptstyle \Box_{X,A}} \\ \mathcal{P}(Y{\times}TA) & \xrightarrow[(f\times \mathrm{id}_{TA})^{-1}]{} & \mathcal{P}(X{\times}TA) \end{array}$$

On the other hand, we define evaluation predicate and possibility in terms of necessity (in the setting of HOL) as follows

- $c{:}TA, v{:}A \vdash (c{\Downarrow}v) \stackrel{\Delta}{\equiv} \forall X{:}PA.([x{\Leftarrow}c]x \in_A X) \supset v \in_A X$, and
- $\Gamma, c{:}TA \vdash (\langle x{\Leftarrow}c\rangle\phi) \stackrel{\Delta}{\equiv} \forall w{:}\Omega.([x{\Leftarrow}c](\phi \supset w)) \supset w$, where $\Gamma, x{:}A \vdash \phi$ prop.

In fact, in classical logic possibility can be defined more simply as $\neg([x{\Leftarrow}c]\neg\phi)$.
There are many strong monads over **Set** preserving inclusions (e.g. those corresponding to single-sorted algebraic theories), among them there are most of the computational monads over **Set** (although the situation is more complex in the category of cpos).

Example 1.1 For each computational monad below we give TA and the meaning of $c{\Downarrow}a$, $[a{\Leftarrow}c]p(x,a)$ and $\langle a{\Leftarrow}c\rangle p(x,a)$, where $a{:}A$, $c{:}TA$, $x{:}X$ and p is a predicate over $X{\times}A$.

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

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

  $[a{\Leftarrow}c]p(x,a)$ iff $\forall a{:}A.c{\Downarrow}_A a \supset p(x,a)$

  $\langle a{\Leftarrow}c\rangle p(x,a)$ iff $\exists a{:}A.c{\Downarrow}_A a \wedge p(x,a)$;

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

  $c{\Downarrow}a$ iff $a \in c$

  $[a{\Leftarrow}c]p(x,a)$ and $\langle a{\Leftarrow}c\rangle p(x,a)$ are defined in terms of $c{\Downarrow}a$ as for exceptions;

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

  $c{\Downarrow}a$ iff $\exists s, s'{:}S.\langle a, s'\rangle \in cs$

  $[a{\Leftarrow}c]p(x,a)$ and $\langle a{\Leftarrow}c\rangle p(x,a)$ are defined in terms of $c{\Downarrow}a$ as for exceptions;

- **resumptions** (and non-determinism) $TA = \mu X.\mathcal{P}_{fin}(A + X)$, i.e. the set of finite trees whose leaves are labelled by elements of A

  $c{\Downarrow}a$ iff "at least one leaf of c is labelled by a"

  $[a{\Leftarrow}c]p(x,a)$ and $\langle a{\Leftarrow}c\rangle p(x,a)$ are defined in terms of $c{\Downarrow}a$ as for exceptions;

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

  $c{\Downarrow}a$ iff $\exists k, k'{:}R^A.ck \neq ck' \wedge (\forall a'{:}A.a' = a \vee ka' = k'a')$, i.e. there exist two continuations $k, k'{:}R^A$ that differ only on a and are distinguished by c

$[a{\Leftarrow}c]p(x,a)$ iff $\forall k,k'{:}R^A.(\forall a{:}A.p(x,a) \supset ka = k'a) \supset ck = ck'$

$\langle a{\Leftarrow}c\rangle p(x,a)$ iff $\neg[a{\Leftarrow}c]\neg p(x,a)$, i.e.
$\exists k,k'{:}R^A.ck \neq ck' \wedge (\forall a{:}A.p(x,a) \vee ka = k'a)$.

1.3 Discussion on related semantics

Here we discuss other set-theoretic interpretations of EL_T, and how they relate to the semantics given above.

Example 1.2 [Extensional semantics of [CP92]] This semantics is defined for any strong monad T by taking

- $c{\Downarrow}a$ iff $c = [a]_T$
- $[a{\Leftarrow}c]p(x,a)$ iff $\forall a{:}A.c{\Downarrow}_A a \supset p(x,a)$
- $\langle a{\Leftarrow}c\rangle p(x,a)$ iff $\exists a{:}A.c{\Downarrow}_A a \wedge p(x,a)$.

In this semantics evaluation modalities are definable in first order logic (FOL) over ML_T. In the case of exceptions $TA = A + E$ this semantics coincides with ours. However, for the monad of non-determinism $TA = \mathcal{P}_{fin}(A)$ (and others), the interpretation of the evaluation predicate (i.e. $c{\Downarrow}_A a$ iff $c = \{a\}$) is not what one would expect (i.e. $c{\Downarrow}_A a$ iff $a \in c$).

Example 1.3 [Side-effects of [Pit91]] This semantics is defined for the monad $TA = \mathcal{P}(A{\times}S)^S$ (and others involving side-effects). Its peculiarity is the *non-standard* interpretation of formulas, i.e. predicates over A are interpreted by subsets of $A{\times}S$ (this should be compared with dynamic logic, where propositions are interpreted by subsets of S). We write $s \models p(a)$ for $\langle a,s\rangle \in p$, where $p \subseteq A{\times}S$ is a predicate over A, $a{:}A$ and $s{:}S$.

- $s \models c{\Downarrow}_{ns} a$ iff $\exists s'{:}S.\langle a,s'\rangle \in cs$
- $s \models [a{\Leftarrow}c]_{ns} p(x,a)$ iff $\forall a{:}A, s'{:}S.\langle a,s'\rangle \in cs \supset s' \models p(x,a)$
- $s \models \langle a{\Leftarrow}c\rangle_{ns} p(x,a)$ iff $\exists a{:}A, s'{:}S.\langle a,s'\rangle \in cs \wedge s' \models p(x,a)$

However, we can express this non-standard semantics of EL_T in ours, provided ML_T comes with operations $lookup{:}TS$ and $update{:}S \to T1$, whose intended interpretation (for this particular choice of monad) is $lookup = \lambda s.\{\langle s,s\rangle\}$ and $update(s) = \lambda s'.\{\langle *,s\rangle\}$.

- $s \models c{\Downarrow}_{ns} a$ iff $(update(s);c){\Downarrow}a$
  where $e_1;e_2$ stand for $(\mathrm{let}\, x{\Leftarrow}e_1\, \mathrm{in}\, e_2)$ with $x \notin \mathrm{FV}(e_2)$
- $s \models [a{\Leftarrow}c]_{ns} p(x,a)$ iff
  $[\langle a,s'\rangle{\Leftarrow}\mathrm{let}\, a{\Leftarrow}(update(s);c)\, \mathrm{in}\, (\mathrm{let}\, s'{\Leftarrow}lookup\, \mathrm{in}\, [\langle a,s'\rangle])]p(x,a,s)$
  where the predicate p over $X{\times}A$ of the non-standard semantics should be viewed as a predicate over $X{\times}A{\times}S$ in our semantics
- $s \models \langle a{\Leftarrow}c\rangle_{ns} p(x,a)$ iff
  $\langle\langle a,s'\rangle{\Leftarrow}\mathrm{let}\, a{\Leftarrow}(update(s);c)\, \mathrm{in}\, (\mathrm{let}\, s'{\Leftarrow}lookup\, \mathrm{in}\, [\langle a,s'\rangle])\rangle p(x,a,s)$

This suggests a general way of translating dynamic logic into EL_T with $lookup$ and $update$. In particular, if p is a proposition of dynamic logic and $c{:}T1 \cong \mathcal{P}(S{\times}S)$ is a program, then $s \models [c]p$ iff $[s'{\Leftarrow}(update(s);c;lookup)]p(s')$ is true. For the monad $TA = \mathcal{P}_{fin}(A{\times}S)^S$, there is another possible interpretation for the evaluation predicate, namely $c{\Downarrow}_* a$ iff $\forall s{:}S.\exists s'{:}S.\langle a,s'\rangle \in cs$. However, this can be expressed in our semantics, since $c{\Downarrow}_* a$ iff $[x{\Leftarrow}c](x = a)$.

Example 1.4 [Continuations revisited] There is another interpretation of the evaluation predicate for the monad $TA = R^{(R^A)}$, namely $c{\Downarrow}_* a$ iff $\forall k,k'{:}R^A.ck = ck' \supset ka = k'a$, i.e. any two continuations $k,k'{:}R^A$ that differ on a are distinguished by c. It does not seem possible to express $c{\Downarrow}_* a$ in our semantics, although one can show that $c{\Downarrow}_* a$ implies $c{\Downarrow}a$ (provided R has at least two elements). It is difficult to say which of the two evaluation predicates captures better the operational intuition.

2 Typed predicate logics

The main objective of this paper is to find axioms for necessity (possibility, and the evaluation predicate of EL_T) *justified* by a general and *convincing* semantics. We cannot expect to achieve this without looking at the *interactions* between necessity and other logical constants[1]. Therefore, we consider various typed (calculi and) predicate logics, which provide suitable contexts for necessity. We believe that a good program logic should be built on *standard* logical machinery, even more so if it should serve as a framework for other program logics. In particular, interpretation of formulas should be consistent with that used in predicate calculus. The only concession we make is to use intuitionistic logic, rather than classical logic. This is in line with *Synthetic Domain Theory*, which views of domains as special sets in a constructive universe (see [Hyl91]).

Remark 2.1 We have not reached a satisfactory integration with SDT, yet. In fact, SDT suggests as models for EL_T toposes with a computational monad over the full sub-category of domains, while models for the strongest of our formal systems are toposes with a strong monad over the whole topos.

In presenting typed predicate logics we use three kinds of judgements: formation judgements, equational judgements and entailment judgements. Formation judgements are used for describing the language, we distinguish four of them:

- $\vdash \tau$ type, which means "τ is a well-formed type";
- $\Gamma \vdash$, which means "Γ is a well-formed context", and the only rules for deriving these judgements are:

  (empty) $\emptyset \vdash$ (extend) $\dfrac{\Gamma \vdash \quad \vdash \tau \text{ type}}{\Gamma, x{:}\tau \vdash}\ x \notin \mathrm{DV}(\Gamma)$
- $\Gamma \vdash e{:}\tau$, which means "e is a well-formed term of type τ in context Γ";
- $\Gamma \vdash \phi$ prop, which means "ϕ is a well-formed formula in context Γ".

Equational judgements are used to present the equational calculus underlying a predicate logic (which in general may have no equality predicate):

- $\Gamma \vdash e_1 = e_2{:}\tau$, which means "$e_1$ and e_2 are equal terms of type τ in context Γ" (and it is implicitly assumed that "the equation $e_1 = e_2{:}\tau$ is well-formed in context Γ", i.e. $\Gamma \vdash e_1{:}\tau$ and $\Gamma \vdash e_2{:}\tau$).

For the logics under consideration it is more convenient to use **sequents** $\Phi \Longrightarrow \phi$ (instead of a formulas), where Φ is a finite set/sequence of formulas. We say that "the sequent $\Phi \Longrightarrow \phi$ is well-formed in context Γ", when $\Gamma \vdash \phi'$ prop for every $\phi' \in \Phi \cup \{\phi\}$. Entailment judgements are used for describing which sequents are true:

- $\Gamma \vdash \Phi \Longrightarrow \phi$, which means "$\Phi$ entails ϕ in context Γ" (and it is implicitly assumed that "the sequent $\Phi \Longrightarrow \phi$ is well-formed in context Γ");

  we also write "$\Gamma \vdash \phi_1 \Longleftrightarrow \phi_2$" as a shorthand for $\Gamma \vdash \phi_1 \Longrightarrow \phi_2$ and $\Gamma \vdash \phi_2 \Longrightarrow \phi_1$.

The general format of rules for deriving judgements is

(name) $\dfrac{Premis_1 \ldots Premis_n}{Conclusion}$ side-condition

i.e. if the premisses are true, then the conclusion is true, we may also use **bi-rules** (as a shorthand for $n+1$ rules)

[1] We have not considered falsity $\bot$, disjunction $\vee$ and existential quantification $\exists$, since their interaction with necessity seems marginal. Moreover, when we are able to interpret possibility and the evaluation predicate, i.e. in HOL, $\vee$ and $\exists$ are definable anyway from $\forall$ and $\supset$.

(name) $\dfrac{Premis_1 \ldots Premis_n}{Conclusion}$ side-condition

i.e. the premisses are true iff the conclusion is true.
In general, we impose the following restrictions on rules:

- formation rules for types may have only formation judgements for types as premisses;
- formation rules for terms and formulas may not have equational or entailment judgements in their premisses.

Notation 2.2 We introduce the following notation for referring to sets of rules:

Notation	Set of rules in
ML	section 2.1
HML	section 2.4
T	section 2.2.1
$\Rightarrow$	section 2.2.2
$\top$	section 2.3.1
$\wedge$	section 2.3.2
$\supset$	section 2.3.3
$\forall$	section 2.3.4
$=$	section 2.3.5
$\Box$	section 2.3.7

In some rules side-conditions may refer to a *signature* Σ (see section 2.1), therefore we are actually defining functions from signatures to sets of rules. In this case, we write $R(\Sigma)$ for the set of rules obtained by taking the signature to be Σ. We introduce also some notation for combining sets of rules:

- if $R1$ and $R2$ are sets of rules, then we write $R1, R2$ for $R1 \cup R2$;
- if r is a rule, then we write r for $\{r\}$;
- if R is either ML or HML (see section 2.1 and 2.4), $R1$ is a set of rules for additional types (see section 2.2.1 and 2.2.2) and $R2$ are sets of rules for logical constants (see section 2.3 and 2.5), then we write $R_{R1}[R2]$ for $R \cup R1 \cup R2$.

Given a set R of rules and a set Th of well-formed equational and entailment judgements (w.r.t. R), we say that Th is a **theory** for R, iff it is closed w.r.t. the equational and entailment rules of R.

2.1 The typed predicate logic $ML(\Sigma)$

Definition 2.3 *A* **signature** Σ *is a triple* $\langle \Sigma^t, \Sigma^f, \Sigma^p \rangle$ *s.t.*

- Σ^t *is a set (of type symbols),*
- Σ^f *is a family of sets* $\Sigma^f_{\tau_1,\tau_2}$ *of (unary) function symbols from* τ_1 *to* τ_2*,*
- Σ^p *is a family of sets* Σ^p_τ *of (unary) predicate symbols over* τ*.*

We do not require that τ, τ_1 and τ_2 be well-formed types, since the set of well-formed types depends not only from Σ, but also from a set of rules R (which is not fixed *a priori*), whose side-conditions may refer to Σ. The price to pay for this liberal definition of signature is some extra checks to prevent non well-formed types to get in (see rule (f) below). Alternatively, we could have defined when a signature is well-formed w.r.t. a set of rules R, but this may become rather involved.

General rules

$$(A)\ \vdash A\ \text{type}\ \ A \in \Sigma^t \qquad (x)\ \frac{\Gamma \vdash}{\Gamma \vdash x : \tau}\ \tau = \Gamma(x) \qquad (f)\ \frac{\Gamma \vdash e : \tau_1 \quad \vdash \tau_2\ \text{type}}{\Gamma \vdash f(e) : \tau_2}\ f \in \Sigma^f_{\tau_1,\tau_2}$$

$$(\text{refl})\ \frac{\Gamma \vdash e : \tau}{\Gamma \vdash e = e : \tau} \qquad (\text{symm})\ \frac{\Gamma \vdash e_1 = e_2 : \tau}{\Gamma \vdash e_2 = e_1 : \tau} \qquad (\text{trans})\ \frac{\Gamma \vdash e_1 = e_2 : \tau \quad \Gamma \vdash e_2 = e_3 : \tau}{\Gamma \vdash e_2 = e_3 : \tau}$$

$$(\text{congr})\ \frac{\Gamma \vdash e_i = e_i' : \tau_i \quad (i = 1, \ldots, n)}{\Gamma \vdash [\overline{e}/\overline{x}]e = [\overline{e}/\overline{x}]e' : \tau} \qquad (\text{subst})\ \frac{\Gamma \vdash e_i : \tau_i \quad (i = 1, \ldots, n) \quad x_i : \tau_1, \ldots, x_n : \tau_n \vdash e = e' : \tau}{\Gamma \vdash [\overline{e}/\overline{x}]e = [\overline{e}/\overline{x}]e' : \tau}$$

$$(p)\ \frac{\Gamma \vdash e : \tau}{\Gamma \vdash p(e)\ \text{prop}}\ p \in \Sigma^p_\tau$$

$$(\text{assume})\ \frac{\Gamma \vdash \phi_i\ \text{prop} \quad (i = 1, \ldots, n)}{\Gamma \vdash \phi_1, \ldots, \phi_n \Longrightarrow \phi_i} \qquad (\text{cut})\ \frac{\Gamma \vdash \Phi \Longrightarrow \phi_i \quad (i = 1, \ldots, n) \quad \Gamma \vdash \phi_1, \ldots, \phi_n \Longrightarrow \phi}{\Gamma \vdash \Phi \Longrightarrow \phi}$$

$$(\text{subst})\ \frac{\Gamma \vdash e_i : \tau_i \quad (i = 1, \ldots, n) \quad x_i : \tau_1, \ldots, x_n : \tau_n \vdash \Phi \Longrightarrow \phi}{\Gamma \vdash [\overline{e}/\overline{x}]\Phi \Longrightarrow [\overline{e}/\overline{x}]\phi} \qquad (\text{congr})\ \frac{\Gamma \vdash e_i = e_i' : \tau_i \quad (i = 1, \ldots, n) \quad \Gamma \vdash [\overline{e}/\overline{x}]\Phi \Longrightarrow [\overline{e}/\overline{x}]\phi}{\Gamma \vdash [\overline{e}'/\overline{x}]\Phi \Longrightarrow [\overline{e}'/\overline{x}]\phi}$$

Unit type

$$(1)\ \vdash 1\ \text{type} \qquad (*)\ \frac{\Gamma \vdash}{\Gamma \vdash * : 1} \qquad (*)\ x : 1 \vdash * = x : 1$$

Product types

$$(\times)\ \frac{\vdash \tau_1\ \text{type} \quad \vdash \tau_2\ \text{type}}{\vdash \tau_1 \times \tau_2\ \text{type}} \qquad (\langle , \rangle)\ \frac{\Gamma \vdash e_1 : \tau_1 \quad \Gamma \vdash e_2 : \tau_2}{\Gamma \vdash \langle e_1, e_2 \rangle : \tau_1 \times \tau_2} \qquad (\pi_i)\ \frac{\Gamma \vdash e : \tau_1 \times \tau_2}{\Gamma \vdash \pi_i(e) : \tau_i}$$

$$(\times.\beta)\ \frac{\Gamma \vdash e_1 : \tau_1 \quad \Gamma \vdash e_2 : \tau_2}{\Gamma \vdash \pi_i(\langle e_1, e_2 \rangle) = e_i : \tau_i} \qquad (\times.\eta)\ \frac{\Gamma \vdash e : \tau_1 \times \tau_2}{\Gamma \vdash \langle \pi_1(e), \pi_2(e) \rangle = e : \tau_1 \times \tau_2}$$

2.2 Additional types

2.2.1 Computational types: T

$$(T)\ \frac{\vdash \tau\ \text{type}}{\vdash T\tau\ \text{type}} \qquad (\text{lift})\ \frac{\Gamma \vdash e : \tau}{\Gamma \vdash [e] : T\tau} \qquad (\text{let})\ \frac{\Gamma \vdash e_1 : T\tau_1 \quad \Gamma, x : \tau_1 \vdash e_2 : T\tau_2}{\Gamma \vdash (\text{let}\ x \Leftarrow e_1\ \text{in}\ e_2) : T\tau_2}$$

$$(\text{let}.\xi)\ \frac{\Gamma \vdash e_1 = e_2 : T\tau_1 \quad \Gamma, x : \tau_1 \vdash e_1' = e_2' : T\tau_2}{\Gamma \vdash \text{let}\ x \Leftarrow e_1\ \text{in}\ e_1' = \text{let}\ x \Leftarrow e_2\ \text{in}\ e_2' : T\tau_2}$$

$$(\text{ass})\ \frac{\Gamma \vdash e_1 : T\tau_1 \quad \Gamma, x_1 : \tau_1 \vdash e_2 : T\tau_2 \quad \Gamma, x_2 : \tau_2 \vdash e_3 : T\tau_3}{\Gamma \vdash \text{let}\ x_2 \Leftarrow (\text{let}\ x_1 \Leftarrow e_1\ \text{in}\ e_2)\ \text{in}\ e_3 = \text{let}\ x_1 \Leftarrow c_1\ \text{in}\ (\text{let}\ x_2 \Leftarrow e_2\ \text{in}\ e_3) : T\tau_3}$$

$$(\text{T}.\beta)\ \frac{\Gamma \vdash e_1 : \tau_1 \quad \Gamma, x : \tau_1 \vdash e_2 : T\tau_2}{\Gamma \vdash \text{let}\ x \Leftarrow [e_1]\ \text{in}\ e_2 = [e_1/x]e_2 : T\tau_2} \qquad (\text{T}.\eta)\ \frac{\Gamma \vdash e : T\tau}{\Gamma \vdash \text{let}\ x \Leftarrow e\ \text{in}\ [x] = e : T\tau}$$

2.2.2 Functional types: $\Rightarrow$

$$(\Rightarrow)\ \frac{\vdash \tau_1\ \text{type} \quad \vdash \tau_2\ \text{type}}{\vdash \tau_1 \Rightarrow \tau_2\ \text{type}} \qquad (\text{app})\ \frac{\Gamma \vdash e_1 : \tau_1 \quad \Gamma \vdash e : \tau_1 \Rightarrow \tau_2}{\Gamma \vdash e e_1 : \tau_2} \qquad (\lambda)\ \frac{\Gamma, x : \tau_1 \vdash e : \tau_2}{\Gamma \vdash (\lambda x : \tau_1 . e) : \tau_1 \Rightarrow \tau_2}$$

$$(\lambda.\xi)\ \frac{\Gamma, x : \tau_1 \vdash e_1 = e_2 : \tau_2}{\Gamma \vdash (\lambda x : \tau_1 . e_1) = (\lambda x : \tau_1 . e_2) : \tau_1 \Rightarrow \tau_2}$$

$$(\Rightarrow.\beta)\ \frac{\Gamma \vdash e_1 : \tau_1 \quad \Gamma, x:\tau_1 \vdash e_2 : \tau_2}{\Gamma \vdash (\lambda x:\tau_1.e_2)e_1 = [e_1/x]e_2 : \tau_2}$$

$$(\Rightarrow.\eta)\ \frac{\Gamma \vdash e : \tau_1 \Rightarrow \tau_2}{\Gamma \vdash (\lambda x:\tau_1.ex) = e : \tau_1 \Rightarrow \tau_2}\ x \notin \mathrm{FV}(e)$$

2.3 Logical constants

In presenting the rules for logical constants we follow the adjoint calculus of [Pit89], which is equivalent to (but more compact of) the natural deduction presentation.

2.3.1 True: $\top$

$$(\top)\ \frac{\Gamma \vdash}{\Gamma \vdash \top\ \mathrm{prop}} \qquad (\top)\ \frac{\Gamma \vdash \Phi \Longrightarrow \phi}{\Gamma \vdash \Phi, \top \Longrightarrow \phi}$$

2.3.2 Conjunction: $\wedge$

$$(\wedge)\ \frac{\Gamma \vdash \phi_1\ \mathrm{prop} \quad \Gamma \vdash \phi_2\ \mathrm{prop}}{\Gamma \vdash \phi_1 \wedge \phi_2\ \mathrm{prop}} \qquad (\wedge)\ \frac{\Gamma \vdash \Phi, \phi_1, \phi_2 \Longrightarrow \phi}{\Gamma \vdash \Phi, \phi_1 \wedge \phi_2 \Longrightarrow \phi}$$

2.3.3 Implication: $\supset$

$$(\supset)\ \frac{\Gamma \vdash \phi_1\ \mathrm{prop} \quad \Gamma \vdash \phi_2\ \mathrm{prop}}{\Gamma \vdash \phi_1 \supset \phi_2\ \mathrm{prop}} \qquad (\supset)\ \frac{\Gamma \vdash \Phi, \phi_1 \Longrightarrow \phi_2}{\Gamma \vdash \Phi, \Longrightarrow \phi_1 \supset \phi_2}$$

2.3.4 Universal quantification: $\forall$

$$(\forall)\ \frac{\Gamma, x;\tau \vdash \phi\ \mathrm{prop}}{\Gamma \vdash \forall x:\tau.\phi\ \mathrm{prop}} \qquad (\forall)\ \frac{\Gamma, x:\tau \vdash \Phi \Longrightarrow \phi}{\Gamma \vdash \Phi \Longrightarrow \forall x:\tau.\phi}\ x \notin \mathrm{FV}(\Phi)$$

2.3.5 Equality: $=$

It is common in predicate logic to view equational judgements as a special form of entailment judgements, by introducing an equality predicate $=_\tau \in \Sigma_{\tau\times\tau}$. On the other hand, hyperdoctrine semantics distinguishes between them, by allowing an *intensional* interpretation of equational judgements.

$$(=)\ \frac{\Gamma \vdash e_1 : \tau \quad \Gamma \vdash e_2 : \tau}{\Gamma \vdash e_1 =_\tau e_2\ \mathrm{prop}} \qquad (=)\ \frac{\Gamma, x:\tau, y:\tau \vdash \Phi, x =_\tau y \Longrightarrow \phi}{\Gamma, x:\tau \vdash [x/y]\Phi \Longrightarrow [x/y]\phi}$$

Congruence for binders are not derivable from (=) and has to be added explicitly. In HOL one can define *Leibniz' equality*, which satisfies already (=).

2.3.6 Additional axioms for $=$

$$(=\text{-}\lambda)\ \frac{\Gamma, x:\tau_1 \vdash \Phi \Longrightarrow e_1 =_{\tau_2} e_2}{\Gamma \vdash \Phi \Longrightarrow (\lambda x:\tau_1.e_1) =_{\tau_1\Rightarrow\tau_2} (\lambda x:\tau_1.e_2)}\ x \notin \mathrm{FV}(\Phi)$$

$$(=\text{let})\ \frac{\Gamma \vdash \Phi \Longrightarrow e_1 =_{T\tau} e_2 \quad \Gamma, x:\tau \vdash \Phi \Longrightarrow e_1' =_{T\tau'} e_2'}{\Gamma \vdash \Phi \Longrightarrow (\mathrm{let}\ x \Leftarrow e_1\ \mathrm{in}\ e_1') =_{T\tau'} (\mathrm{let}\ x \Leftarrow e_2\ \mathrm{in}\ e_2')}$$

Remark 2.4 In $ML_\Rightarrow[=,\forall]$ the rule (=-λ) is equivalent to extensionality for functions, i.e. $f, g:\tau_1\Rightarrow\tau_2 \vdash (\forall x:\tau_1.fx =_{\tau_2} gx) \Longrightarrow f =_{\tau_1\Rightarrow\tau_2} g$. In $HML_\Rightarrow$ the rule (=-λ) is derivable from (Comp-$\Rightarrow$), but they are not equivalent. In $ML_T[=,\Box]$ the rule (=-let) is derivable from ($\Box$-=), but they are not equivalent.

2.3.7 Necessity: $\Box$

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

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

$$(\Box\text{-}\Longrightarrow)\ \frac{\Gamma, x{:}\tau \vdash \Phi, \phi \Longrightarrow \psi}{\Gamma, c{:}T\tau \vdash \Phi, [x{\Leftarrow}c]\phi \Longrightarrow [x{\Leftarrow}c]\psi}\ x \notin \mathrm{FV}(\Phi)$$

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

$$(\Box\text{-}\mu)\ \frac{\Gamma, x{:}\tau \vdash \phi\ \text{prop}}{\Gamma, c{:}T^2\tau \vdash [y{\Leftarrow}c]([x{\Leftarrow}y]\phi) \Longrightarrow [x{\Leftarrow}(\text{let } y{\Leftarrow}c \text{ in } y)]\phi}$$

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

$$(\Box\text{-t}^*)\ \frac{\Gamma, z{:}\tau_1{\times}\tau_2 \vdash \phi\ \text{prop}}{\Gamma, x{:}\tau_1, c{:}T\tau_2 \vdash [y{\Leftarrow}c]([\langle x, y\rangle/z]\phi) \Longleftrightarrow [z{\Leftarrow}(\text{let } y{\Leftarrow}c \text{ in } [\langle x, y\rangle])]\phi}$$

$$(\Box\text{-}\wedge^*)\ \frac{\Gamma, x{:}\tau \vdash \phi_1\ \text{prop} \quad \Gamma, x{:}\tau \vdash \phi_2\ \text{prop}}{\Gamma, c{:}T\tau \vdash [x{\Leftarrow}c](\phi_1 \wedge \phi_2) \Longleftrightarrow ([x{\Leftarrow}c]\phi_1) \wedge ([x{\Leftarrow}c]\phi_2)}$$

$$(\Box\text{-}T^*)\ \frac{\Gamma, x{:}\tau_1 \vdash e{:}\tau_2 \quad \Gamma, y{:}\tau_2 \vdash \phi\ \text{prop}}{\Gamma, c{:}T\tau_1 \vdash [x{\Leftarrow}c]([e/y]\phi) \Longleftrightarrow [y{\Leftarrow}(\text{let } x{\Leftarrow}c \text{ in } [e])]\phi}$$

2.3.8 Additional axioms for $\Box$

$$(\Box\text{-}\eta^*)\ \frac{\Gamma, x{:}\tau \vdash \phi\ \text{prop}}{\Gamma, x{:}\tau \vdash \phi \Longleftrightarrow [x{\Leftarrow}[x]]\phi}$$

$$(\Box\text{-}\mu^*)\ \frac{\Gamma, x{:}\tau \vdash \phi\ \text{prop}}{\Gamma, c{:}T^2\tau \vdash [y{\Leftarrow}c]([x{\Leftarrow}y]\phi) \Longleftrightarrow [x{\Leftarrow}(\text{let } y{\Leftarrow}c \text{ in } y)]\phi}$$

$$(\Box\text{-}\supset^*)\ \frac{\Gamma \vdash \phi_1\ \text{prop} \quad \Gamma, x{:}\tau \vdash \phi_2\ \text{prop}}{\Gamma, c{:}T\tau \vdash [x{\Leftarrow}c](\phi_1 \supset \phi_2) \Longleftrightarrow \phi_1 \supset ([x{\Leftarrow}c]\phi_2)}$$

$$(\Box\text{-}\forall^*)\ \frac{\Gamma, x{:}\tau_1, y{:}\tau_2 \vdash \phi\ \text{prop}}{\Gamma, c{:}T\tau_1 \vdash [x{\Leftarrow}c](\forall y{:}\tau_2.\phi) \Longleftrightarrow \forall y{:}\tau_2.([x{\Leftarrow}c]\phi)}$$

$$(\Box\text{-}=)\ \frac{\Gamma, x{:}\tau_1 \vdash e_1{:}\tau_2 \quad \Gamma, x{:}\tau_1 \vdash e_2{:}\tau_2}{\Gamma, c{:}T\tau_1 \vdash ([x{\Leftarrow}c]e_1 =_{\tau_2} e_2) \Longrightarrow (\text{let } x{\Leftarrow}c \text{ in } [e_1]) =_{T\tau_2} (\text{let } x{\Leftarrow}c \text{ in } [e_2])}$$

Remark 2.5 All axioms proposed in [Pit91] for necessity are derivable in $ML_T[\top,\wedge,\Box,\Box\text{-}\eta^*,\Box\text{-}\mu^*]$, and conversely all rules in $\Box,\Box\text{-}\eta^*,\Box\text{-}\mu^*$ are derivable in Pitts' Evaluation Logic, with the exception of $(\Box\text{-}\Longrightarrow)$. In fact, Pitts allows only a restricted form of $(\Box\text{-}\Longrightarrow)$, where Φ is empty, which (unlike the more general form) is sound for his non-standard interpretation of formulas (see Example 1.3).

2.4 Higher Order Logic: HML

HML is an extension of $ML[\top,\wedge,\supset,\forall]$ where formulas are *represented* as terms of a type Ω of truth values. In particular, logical constants and predicate symbols are replaced by function symbols, namely: $p \in \Sigma^p_\tau$ is replaced by $p \in \Sigma^f_{\tau,\Omega}$, and the (formation rules for) logical constants are replaced by $\top \in \Sigma^f_{1,\Omega}$, $\wedge, \supset \in \Sigma^f_{\Omega\times\Omega,\Omega}$, $\forall_\tau \in \Sigma^f_{P\tau,\Omega}$, $=_\tau \in \Sigma^f_{\tau\times\tau,\Omega}$, and $\Box_\tau \in \Sigma^f_{T\tau\times P\tau,\Omega}$.

$$(\Omega)\ \ \Omega \vdash \text{type} \qquad (P)\ \frac{\vdash \tau\ \text{type}}{\vdash P\tau\ \text{type}} \qquad ()\ \frac{\Gamma \vdash \phi{:}\Omega}{\Gamma \vdash \phi\ \text{prop}}$$

$$(\{|\})\ \frac{\Gamma, x{:}\tau \vdash \phi{:}\Omega}{\Gamma \vdash \{x{:}\tau|\phi\}{:}P\tau} \qquad (\in)\ \frac{\Gamma \vdash E{:}P\tau \quad \Gamma \vdash e{:}\tau}{\Gamma \vdash e \in_\tau E{:}\Omega}$$

$$(\{|\}.\xi)\ \frac{\Gamma, x{:}\tau \vdash \phi_1 = \phi_2 : \Omega}{\Gamma \vdash \{x{:}\tau|\phi_1\} = \{x{:}\tau|\phi_2\} : P\tau} \qquad (P.\beta)\ \frac{\Gamma \vdash e{:}\tau \quad \Gamma, x{:}\tau \vdash \phi : \Omega}{\Gamma \vdash [e/x]\phi = e \in_\tau \{x{:}\tau|\phi\} : \Omega}$$

$$(P.\eta)\ \frac{\Gamma \vdash E : P\tau}{\Gamma \vdash E = \{x{:}\tau | x \in_\tau E\} : P\tau}\ x \notin \mathrm{DV}(\Gamma)$$

Sometimes, we may write Ω^τ for $P\tau$, $E(e)$ for $e \in_\tau E$ and $\lambda x{:}\tau.\phi$ for $\{x{:}\tau|\phi\}$.
The term representation of formulas is more *intensional*, i.e. $\Gamma \vdash \phi_1 = \phi_2 : \Omega$ implies $\Gamma \vdash \phi_1 \iff \phi_2$, while the converse may not be true. In HML all logical constants (except necessity) could be expressed in terms of $\supset$ and $\forall$

- $\bot \triangleq \forall X{:}P1.X(*)$
- $\phi_1 \vee \phi_2 \triangleq \forall X{:}P1.((\phi_1 \supset X(*)) \wedge (\phi_2 \supset X(*))) \supset X(*)$, where $X \notin \mathrm{FV}(\phi)$
- $x \simeq_\tau y \triangleq \forall X{:}P\tau.X(x) \supset X(y)$, called **Leibniz' equality**
- $\exists x{:}\tau.\phi \triangleq \forall X{:}P1.(\forall x{:}\tau.\phi \supset X(*)) \supset X(*)$, where $X \notin \mathrm{FV}(\phi)$
- $\exists! x{:}\tau.\phi \triangleq (\exists y{:}\tau.\forall x{:}\tau.\phi \leftrightarrow x \simeq_\tau y)$, where $y \notin \mathrm{FV}(\phi)$

so that the following rules are derivable

$$(\bot)\ \frac{\Gamma \vdash \phi\ \mathrm{prop}}{\Gamma \vdash \bot \Longrightarrow \phi} \qquad (\vee)\ \frac{\Gamma \vdash \Phi, \phi_1 \Longrightarrow \phi \quad \Gamma \vdash \Phi, \phi_2 \Longrightarrow \phi}{\Gamma \vdash \Phi, (\phi_1 \vee \phi_2) \Longrightarrow \phi}$$

$$(=)\ \frac{\Gamma, x{:}\tau, y{:}\tau \vdash \Phi, x \simeq_\tau y \Longrightarrow \phi}{\Gamma, x{:}\tau \vdash [x/y]\Phi \Longrightarrow [x/y]\phi} \qquad (\exists)\ \frac{\Gamma, x{:}\tau \vdash \Phi, \psi \Longrightarrow \phi}{\Gamma \vdash \Phi, (\exists x{:}\tau.\psi) \Longrightarrow \phi}\ x \notin \mathrm{FV}(\Phi, \phi)$$

2.5 Additional axioms for HML

2.5.1 Comprehension for powersets: Comp-P

$$(\text{Comp-}P)\ \frac{\Gamma, x{:}\tau \vdash \phi\ \mathrm{prop}}{\Gamma \vdash \emptyset \Longrightarrow \exists! X{:}P\tau.(\forall x{:}\tau.\phi \leftrightarrow x \in_\tau X)}\ X \notin \mathrm{FV}(\phi)$$

Remark 2.6 In HML the rule (Comp-P) is equivalent to extensionality for sets, i.e. $X, Y{:}P\tau \vdash (\forall x{:}\tau.x \in_\tau X \leftrightarrow x \in_\tau Y) \Longrightarrow X \simeq_{P\tau} Y$.

2.5.2 Comprehension for functional types: Comp-$\Rightarrow$

$$(\text{Comp-}\Rightarrow)\ \frac{\Gamma, x{:}\tau_1, y{:}\tau_2 \vdash \phi\ \mathrm{prop}}{\Gamma \vdash (\forall x{:}\tau_1.\exists! y{:}\tau_2.\phi(x,y)) \Longrightarrow (\exists! f{:}\tau_1 \Rightarrow \tau_2.\forall x{:}\tau_1.\phi(x, fx))}\ f \notin \mathrm{FV}(\phi)$$

(Comp-$\Rightarrow$) says that there is a one-one correspondence between functions and functional relations.

2.5.3 Comprehension for computational types: Comp-T

$$(\text{Comp-}T)\ \frac{\vdash \tau\ \mathrm{type}}{C{:}T(P\tau) \vdash \begin{array}{l}([X \Leftarrow C]\exists! x{:}\tau.x \in_\tau X) \iff \\ \exists! c{:}T\tau.C \simeq_{T(P\tau)} (\mathrm{let}\ x \Leftarrow c\ \mathrm{in}\ [\{x\}])\end{array}}$$

where $\{x\} \triangleq \{y | y \simeq_\tau x\}$.

2.6 Formal consequences

This section gives few useful axioms and rules derivable in some of the formal systems introduced previously. Of special significance are the definability results in Theorem 2.12 and 2.13, whose main consequences are: necessity is expressible in HML_T, necessity and possibility are definable from the evaluation predicate, when necessity commutes with $\supset$ and $\forall$.

Lemma 2.7 *In* $ML_T[\top, \wedge, \Box\text{-}\top^*, \Box\text{-}\Longrightarrow, \Box\text{-}T](\Sigma)$ *the following are derivable:*

1. ($\Box$-D1) $\Gamma, c{:}T\tau \vdash \phi \Longrightarrow [x{\Leftarrow}c]\phi \;\; x \notin \mathrm{FV}(\phi)$
2. ($\Box$-$\wedge$) $\Gamma, c{:}T\tau \vdash [x{\Leftarrow}c](\phi_1 \wedge \phi_2) \Longrightarrow ([x{\Leftarrow}c]\phi_1) \wedge ([x{\Leftarrow}c]\phi_2)$
3. ($\Box$-$\supset$) $\Gamma, c{:}T\tau \vdash [x{\Leftarrow}c](\phi_1 \supset \phi_2) \Longrightarrow \phi_1 \supset ([x{\Leftarrow}c]\phi_2) \;\; x \notin \mathrm{FV}(\phi_1)$
4. ($\Box$-$\forall$) $\Gamma, c{:}T\tau_1 \vdash [x{\Leftarrow}c](\forall y{:}\tau_2.\phi) \Longrightarrow \forall y{:}\tau_2.([x{\Leftarrow}c]\phi)$
5. ($\Box$-iso) $\dfrac{\Gamma, x{:}\tau_1 \vdash e{:}\tau_2 \text{ iso}}{\Gamma, c{:}T\tau_1 \vdash [x{\Leftarrow}c]([e/y]\phi) \Longleftrightarrow [y{\Leftarrow}(\text{let } x{\Leftarrow}c \text{ in } [e])]\phi} \;\; x \notin \mathrm{FV}(\phi)$

   where $\Gamma, x{:}\tau_1 \vdash e{:}\tau_2$ iso *means that for some* e' *the equational judgements* $\Gamma, x{:}\tau_1 \vdash x = [e/y]e'{:}\tau_1$ *and* $\Gamma, y{:}\tau_2 \vdash y = [e'/x]e{:}\tau_2$ *are derivable*
6. ($\Box$-let) $\Gamma, c{:}T\tau_1 \vdash [x{\Leftarrow}c]([y{\Leftarrow}e]\phi) \Longrightarrow [y{\Leftarrow}(\text{let } x{\Leftarrow}c \text{ in } e)]\phi \;\; x \notin \mathrm{FV}(\phi)$

   using also ($\Box$-μ).

Proof see Appendix ∎

Lemma 2.8 *In* $ML_T[\top, \wedge, \Box\text{-}\top^*, \Box\text{-}\Longrightarrow, \Box\text{-}T^*, \Box\text{-}\wedge^*](\Sigma)$ *the following are derivable:*

1. ($\Box$-t*)
2. ($\Box$-let*) $\Gamma, c{:}T\tau_1 \vdash [x{\Leftarrow}c]([y{\Leftarrow}e]\phi) \Longleftrightarrow [y{\Leftarrow}(\text{let } x{\Leftarrow}c \text{ in } e)]\phi \;\; x \notin \mathrm{FV}(\phi)$

   using also ($\Box$-μ*).

Proof see Appendix ∎

Lemma 2.9 *In* $ML_T[\top, \wedge, =, \Box\text{-}\top^*, \Box\text{-}\Longrightarrow, \Box\text{-}T, \Box\text{-}=](\Sigma)$ *the following are derivable:*

1. ($\Box$-=iso) $\dfrac{\Gamma, x{:}\tau_1 \vdash \Phi \Longrightarrow e{:}\tau_2 \text{ iso}}{\Gamma, c{:}T\tau_1 \vdash \Phi, [y{\Leftarrow}(\text{let } x{\Leftarrow}c \text{ in } [e])]\phi \Longrightarrow [x{\Leftarrow}c]([e/y]\phi)} \;\; x \notin \mathrm{FV}(\Phi, \phi)$

   where $\Gamma, x{:}\tau_1 \vdash \Phi \Longrightarrow e{:}\tau_2$ iso *means that for some* e' *the entailment judgements* $\Gamma, x{:}\tau_1 \vdash \Phi \Longrightarrow x =_{\tau_1} [e/y]e'$ *and* $\Gamma, y{:}\tau_2 \vdash \Phi \Longrightarrow y =_{\tau_2} [e'/x]e$ *are derivable*
2. ($\Box^+$-=) $\Gamma, c{:}T\tau_1 \vdash ([x{\Leftarrow}c]e_1 =_{T\tau_2} e_2) \Longrightarrow (\text{let } x{\Leftarrow}c \text{ in } e_1) =_{T\tau_2} (\text{let } x{\Leftarrow}c \text{ in } e_2)$
3. (=-let)

Proof see Appendix ∎

Lemma 2.10 *In* HML[Comp-P](Σ) *the following are derivable:*

1. (ext-P) $X, Y{:}P\tau \vdash (\forall x{:}\tau.x \in_\tau X \leftrightarrow x \in_\tau Y) \Longrightarrow X \simeq_{P\tau} Y$
2. (ext-Ω) $X{:}\Omega \vdash X \Longrightarrow X \simeq_\Omega \top$

Proof see Appendix ∎

Lemma 2.11 *In* $HML_\Rightarrow$[Comp-$\Rightarrow$](Σ) *the rule* (=-λ) *is derivable.*

Proof see Appendix ∎

Theorem 2.12 *In* HML_T[Comp-P, $\Box\text{-}\top^*, \Box\text{-}\Longrightarrow, \Box\text{-}T^*, \Box\text{-}=](\Sigma)$ *the following assertions are equivalent:*

- $\Gamma, c{:}T\tau \vdash [x{\Leftarrow}c]\phi$
- $\Gamma, c{:}T\tau \vdash (\text{let } x{\Leftarrow}c \text{ in } [\phi]) =_{T\Omega} (\text{let } x{\Leftarrow}c \text{ in } [\top])$

- $\Gamma, c{:}T\tau \vdash \Box(\mathrm{let}\ x{\Leftarrow}c\ \mathrm{in}\ [\phi])$, *where* $c{:}T\Omega \vdash \Box(c) \stackrel{\Delta}{\equiv} [X{\Leftarrow}c]X$.

Proof see Appendix ∎

Theorem 2.13 *In* $HML_T[\Box\text{-}\top^*,\Box\text{-}\Longrightarrow,\Box\text{-}T^*,\ \Box\text{-}\supset^*,\Box\text{-}\forall^*](\Sigma)$ *the following assertions are derivable:*

1. $c{:}T\tau \vdash [x{\Leftarrow}c](c{\Downarrow}x)$
2. $\Gamma, c{:}T\tau \vdash ([x{\Leftarrow}c]\phi) \Longleftrightarrow (\forall x{:}\tau.c{\Downarrow}x \supset \phi)$
3. $\Gamma, c{:}T\tau \vdash (\langle x{\Leftarrow}c\rangle\phi) \Longleftrightarrow (\exists x{:}\tau.c{\Downarrow}x \wedge \phi)$
4. $c{:}T\tau, v{:}\tau \vdash (c{\Downarrow}v) \Longleftrightarrow \langle x{\Leftarrow}c\rangle(x =_\tau v)$

Proof see Appendix ∎

3 Categorical semantics

Given a category $\mathcal{C}$ with finite products, the general pattern for interpreting a typed calculus according to Lawvere's functorial semantics goes as follows:

- a context $\Gamma \vdash$ and a type $\vdash \tau$ type are interpreted by objects of $\mathcal{C}$, by abuse of notation we will indicate these objects with Γ and τ respectively;

  in particular, the empty context $\emptyset \vdash$ is interpreted by the terminal object 1 and the context $\Gamma, x{:}\tau \vdash$ is interpreted by $\Gamma{\times}\tau$;

- a term $\Gamma \vdash e{:}\tau$ is interpreted by a morphism from Γ to τ in $\mathcal{C}$, indicated with e (again by abuse of notation);

- a (well-formed) equational judgement $\Gamma \vdash e_1 = e_2{:}\tau$ is true iff $e_1 = e_2$ as morphisms in $\mathcal{C}$.

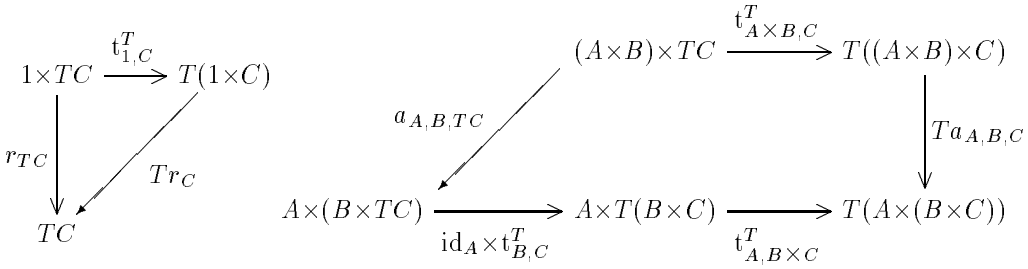
We refer to the literature for details on the categorical semantics of typed calculi (e.g. see [Pit88, Pit89, Mog91]).
In categorical logic there are two approaches which extend Lawvere's functorial semantics to typed predicate logics: the *topos-theoretic* approach interprets formulas as subobjects, while the *hyperdoctrine* approach interprets formulas in the fibers of a $\mathcal{C}$-indexed category. The obvious trade-off between these two approaches is that the first is closer to models, while the second is closer to theories. In this section we recall the categorical structures proposed in the literature for interpreting computational types (strong monads) and logical constants.

3.1 Strong monads

Strong monads are used for interpreting computational types. We refer to [Mog91] for details about the interpretation and the proof of soundness and completeness. Monads over a category $\mathcal{C}$ can be viewed as monoids in the strict monoidal category $End(\mathcal{C})$ of endofunctors and natural transformations, where the tensor $S \otimes T$ is composition $S;T$. Strong monads over a category $\mathcal{C}$ with finite products (more generally a symmetric monoidal category) enjoy a similar characterisation as monoids in the strict monoidal category $SEndo(\mathcal{C})$ of strong endofunctors and strong transformations.

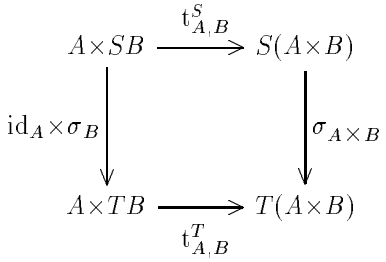
Definition 3.1 *Given a category with finite products, we define the strict monoidal category* $SEndo(\mathcal{C})$ *as follows:*

- *an object is a* **strong endofunctor** (T, t^T)*, i.e.* $T{:}\mathcal{C} \to \mathcal{C}$ *is a functor and* $\mathrm{t}^T_{A,B}{:}A{\times}TB \to T(A{\times}B)$ *is a natural transformation s.t.*



  where $r_A{:}(1{\times}A) \to A$ *and* $a_{A,B,C}{:}(A{\times}B){\times}C \to A{\times}(B{\times}C)$ *are the obvious natural isomorphisms (in the sequel they are left implicit);*

- *a morphism from* (S, t^S) *to* (T, t^T) *is a* **strong transformation** $\sigma{:}(S, \mathrm{t}^S) \to (T, \mathrm{t}^T)$*, i.e.* $\sigma{:}S \to T$ *is a natural transformation s.t.*



  composition is given by vertical composition of natural transformation;

- *the tensor* $(S, \mathrm{t}^S) \otimes (T, \mathrm{t}^T)$ *is* $(S; T, \mathrm{t})$*, where* $\mathrm{t}_{A,B} = \mathrm{t}^T_{A,SB}; T(\mathrm{t}^S_{A,B})$*;*
  $\sigma \otimes \tau$ *is given by horizontal composition of natural transformation;*
  the unit I *for* $\otimes$ *is* $(\mathrm{id}_{\mathcal{C}}, \mathrm{t})$*, where* $\mathrm{t}_{A,B} = \mathrm{id}_{A\times B}$*.*

In the sequel we will consider additional properties/structure for strong monads, which will allow to extend the interpretation beyond $ML_T(\Sigma)$.

3.2 Hyperdoctrine semantics

The hyperdoctrine approach based on indexed categories can represent also proof-theoretic aspects of a logic, but we are interested only in provability (i.e. we need only indexed preorders). To simplify things even further we will also identify formulas that are provably equivalent (i.e. we need only indexed posets).

Definition 3.2 *If* $\mathcal{C}$ *is a (small) category, then a* **$\mathcal{C}$-indexed category** *is a functor* $\mathcal{P}{:}\mathcal{C}^{op} \to$ **Cat**, *where* **Cat** *is the category of small categories and functors. We call* $\mathcal{P}[A]$ *the* **fiber** *over* $A \in \mathcal{C}$ *and the functor* $\mathcal{P}[f]{:}\mathcal{P}[A] \to \mathcal{P}[B]$ **substitution** *along* $f{:}B \to A$ *(substitution along* f *is indicated also with* f^**, when* $\mathcal{P}$ *is clear from the context). By analogy, when* **Cat** *is replaced with some other category* $\mathcal{W}$*, say the category of* widgets*, we will call a functor* $\mathcal{P}{:}\mathcal{C}^{op} \to \mathcal{W}$ *a* **$\mathcal{C}$-indexed widget**.

In the sequel we will consider only $\mathcal{P}{:}\mathcal{C}^{op} \to \mathcal{W}$ with $\mathcal{W}$ subcategory of **PoSet**. The *hyperdoctrine* semantics of a typed predicate logic in a $\mathcal{C}$-indexed meet semi-lattice $\mathcal{P}$ extends Lawvere's functorial semantics as follows:

- The interpretation of a formula $\Gamma \vdash \phi$ prop is an element of $\mathcal{P}[\Gamma]$, indicated with ϕ (again by abuse of notation).

- The sequent $\Gamma \vdash \phi_1, \ldots, \phi_n \Longrightarrow \phi$ is true iff $(\wedge_i \phi_i) \leq \phi$ in $\mathcal{P}[\Gamma]$.

For interpreting logical constants (in general) one must require additional properties on indexed meet semi-lattices.

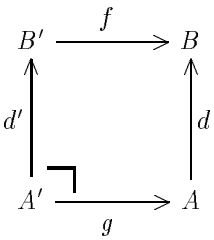
Definition 3.3 *Given a $\mathcal{C}$-indexed meet semi-lattice $\mathcal{P}$ (i.e. each fiber $\mathcal{P}[A]$ has a top element $\top$ and binary meets $a \wedge b$, which are preserved by substitution functors) we say that $\mathcal{P}$ is closed w.r.t.* **implication** *iff each fiber has pseudo-complements $a \supset b$ (i.e. $c \leq a \supset b$ iff $c \wedge a \leq b$), which are preserved by substitution functors.*

Indexed lattices with implication correspond to indexed Heyting algebras.

Lemma 3.4 *A poset with finite meets has automatically pullbacks, and if it has also pseudo-complements then it is a locally cartesian closed category.*

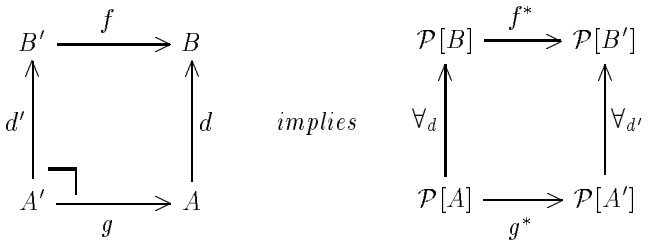
Proof The pullback of $b \to a$ and $c \to a$ is given by $b \wedge c$. The exponential of $c \to a$ to $b \to a$ in the slice category over a is given by $(b \supset c) \wedge a$. ∎

Definition 3.5 ([Tay87]) *A* **class of display maps** *$\mathcal{D}$ over $\mathcal{C}$ is a class of morphisms in $\mathcal{C}$ closed w.r.t. pullbacks, i.e. for every $d: A \to B$ in $\mathcal{D}$ and $f: B' \to B$ exist $d': A' \to B'$ in $\mathcal{D}$ and $g: A' \to A$ s.t.*



Definition 3.6 *Given a $\mathcal{C}$-indexed poset $\mathcal{P}$ and a class of display maps $\mathcal{D}$ over $\mathcal{C}$ we say that:*

- *$\mathcal{P}$ is closed w.r.t.* **universal quantification** *along maps in $\mathcal{D}$ iff for all $d: A \to B$ in $\mathcal{D}$ exists the right adjoint $\forall_d$ to $d^*: \mathcal{P}[B] \to \mathcal{P}[A]$ satisfying the Beck-Chevalley condition*



(since $\mathcal{P}$ is an indexed poset, the right adjoint $\forall_d$ is unique, and the usual Beck-Chevalley condition is equivalent to the one given above)

Definition 3.7 *Given a category $\mathcal{C}$ with finite products and a $\mathcal{C}$-indexed meet semi-lattice $\mathcal{P}$ we say that $\mathcal{P}$ has* **equality** *over A iff exists $=_A \in \mathcal{P}[A \times A]$ s.t. for all $X \in \mathcal{C}$ the monotonic function $(\mathrm{id}_X \times \Delta_A)^*: \mathcal{P}[X \times A \times A] \to \mathcal{P}[X \times A]$ has a left adjoint $\exists: \mathcal{P}[X \times A] \to \mathcal{P}[X \times A \times A]$ given by $\exists(a) = \pi_1^*(a) \wedge \pi_2^*(=_A)$, where $a \in \mathcal{P}[X \times A]$, $\pi_1: X \times A \times A \to X \times A$ and $\pi_2: X \times A \times A \to A \times A$.*

The predicate $=_A$ is necessarily unique, and $\exists$ defined above satisfies the Beck-Chevalley and Frobenious Reciprocity conditions, because substitution preserves finite meets.

Definition 3.8 ([Pit81]) *Given a $\mathcal{C}$-indexed meet semi-lattice $\mathcal{P}$ we say that:*

- *$t \in \mathcal{P}[\Omega]$ is a* **(skeletal) generic predicate** *iff for all $A \in \mathcal{C}$ and $a \in \mathcal{P}[A]$ there exists (unique) $f: A \to \Omega$ s.t. $a = \mathcal{P}[f]t$, where Ω is some given object of $\mathcal{C}$.*
- *$\mathcal{P}$ is a* **tripos** *iff*

– *$\mathcal{C}$ has finite products, and for some distinguished object Ω it has exponentials Ω^A for every $A \in \mathcal{C}$;*

– *$\mathcal{P}$ is closed w.r.t. implication and universal quantification along first projections*

– *there is some distinguished generic predicate $t \in \mathcal{P}[\Omega]$.*

Remark 3.9 There are some minor differences with the definition of tripos given in [Pit81]:

- we do not require that $\mathcal{P}$ is closed w.r.t. universal quantification along arbitrary maps, but this seems irrelevant for constructing a topos out of a tripos;
- we do not explicitly require disjunction and existential quantification, since in HOL they are definable by implication and universal quantification;

Therefore, a tripos according to our definition is a tripos in Pitts' sense (except for quantification along arbitrary maps).

Lemma 3.10 *If $t \in \mathcal{P}[\Omega]$ and $t' \in \mathcal{P}[\Omega']$ are skeletal generic predicates, then the unique $f{:}\Omega \to \Omega'$ s.t. $t = \mathcal{P}[f]t'$ is an isomorphism and $t' = \mathcal{P}[f^{-1}]t$.*

3.2.1 Hyperdoctrine interpretation

Given a category $\mathcal{C}$ with finite products and a $\mathcal{C}$-indexed meet semi-lattice $\mathcal{P}$, the interpretation of formulas in $\mathcal{P}$ is defined by induction on the derivation of $\Gamma \vdash \phi$ prop (see [Pit89]). This is done by assigning to each predicate symbol $p \in \Sigma^t_\tau$ an interpretation $p \in \mathcal{P}[\tau]$, and by defining for each formation rule (for formulas) the interpretation of its conclusion in terms of the interpretation of its premisses (and additional structure/properties for $\mathcal{P}$):

$\Gamma \vdash e{:}\tau$	$=$	$e{:}\Gamma \to \tau$
$\Gamma \vdash p(e)$ prop	$=$	$e^*p \in \mathcal{P}[\Gamma]$
$\Gamma \vdash$	$=$	$\Gamma \in \mathcal{C}$
$\Gamma \vdash \top$ prop	$=$	$\top \in \mathcal{P}[\Gamma]$
$\Gamma \vdash \phi_1$ prop	$=$	$\phi_1 \in \mathcal{P}[\Gamma]$
$\Gamma \vdash \phi_2$ prop	$=$	$\phi_2 \in \mathcal{P}[\Gamma]$
$\Gamma \vdash \phi_1 \wedge \phi_2$ prop	$=$	$\phi_1 \wedge \phi_2 \in \mathcal{P}[\Gamma]$

- if $\mathcal{P}$ is closed w.r.t. implication, then

$\Gamma \vdash \phi_1$ prop	$=$	$\phi_1 \in \mathcal{P}[\Gamma]$
$\Gamma \vdash \phi_2$ prop	$=$	$\phi_2 \in \mathcal{P}[\Gamma]$
$\Gamma \vdash \phi_1 \supset \phi_2$ prop	$=$	$\phi_1 \supset \phi_2 \in \mathcal{P}[\Gamma]$

- if $\mathcal{P}$ is closed w.r.t. universal quantification along $\pi_1{:}A{\times}\tau \to A$ (for $A \in \mathcal{C}$ and τ interpretation of a well-formed type), then

$\Gamma, x{:}\tau \vdash \phi$ prop	$=$	$\phi \in \mathcal{P}[\Gamma{\times}\tau]$
$\Gamma \vdash \forall x{:}\tau.\phi$ prop	$=$	$\forall_d(\phi) \in \mathcal{P}[\Gamma]$ where $d = \pi_1{:}\Gamma{\times}\tau \to \Gamma$

- if $\mathcal{P}$ has equality over τ (for τ interpretation of a well-formed type), then

$\Gamma \vdash e_1{:}\tau$	$=$	$e_1{:}\Gamma \to \tau$
$\Gamma \vdash e_2{:}\tau$	$=$	$e_2{:}\Gamma \to \tau$
$\Gamma \vdash e_1 =_\tau e_2$ prop	$=$	$\langle e_1, e_2\rangle^*(=_\tau) \in \mathcal{P}[\Gamma]$

- if $t \in \mathcal{P}[\Omega]$ is a generic predicate and $\mathcal{C}$ has exponentials of the form Ω^τ (for τ interpretation of a well-formed type), then

$\vdash \Omega$ type	$=$	$\Omega \in \mathcal{C}$
$\vdash \tau$ type	$=$	$\tau \in \mathcal{C}$
$\vdash P\tau$ type	$=$	$\Omega^\tau \in \mathcal{C}$
$\Gamma, x{:}\tau \vdash \phi{:}\Omega$	$=$	$\phi{:}\Gamma\times\tau \to \Omega$
$\Gamma \vdash \{x{:}\tau \mid \phi\}{:}P\tau$	$=$	$\Lambda(\phi){:}\Gamma \to \Omega^\tau$
$\Gamma \vdash e{:}\tau$	$=$	$E{:}\Gamma \to \tau$
$\Gamma \vdash E{:}P\tau$	$=$	$e{:}\Gamma \to \Omega^\tau$
$\Gamma \vdash e \in_\tau E{:}\Omega$	$=$	$\langle E, e\rangle; eval{:}\Gamma \to \Omega$
$\Gamma \vdash \phi{:}\Omega$	$=$	$\phi{:}\Gamma \to \Omega$
$\Gamma \vdash \phi$ prop	$=$	$\phi^* t \in \mathcal{P}[\Gamma]$

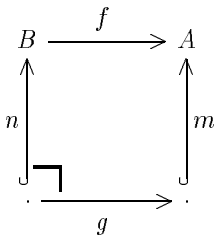
  moreover, the interpretation of function symbols representing logical constants and predicate symbols must be *consistent* (see [See87]), e.g. the interpretation of $\wedge \in \Sigma^f_{\Omega\times\Omega,\Omega}$ must satisfy $(\langle\phi_1,\phi_2\rangle;\wedge)^* t = \phi_1^* t \wedge \phi_2^* t$ for every $\phi_1, \phi_2 : A \to \Omega$ (where $\wedge$ in the rhs is meet in $\mathcal{P}[A]$).

3.3 Topos-theoretic semantics

The *topos-theoretic* semantics of formulas mimics the set-theoretic one by using the categorical analogue of subsets (i.e. *subobjects*), and can be viewed as a special case of hyperdoctrine semantics, where formulas are interpreted in indexed posets made of subobjects in the base.

Definition 3.11 *The* **inclusion preorder** $\leq$ *on monos is defined by* $m_1 \leq m_2$ *iff exists* $m{:}d_1 \to d_2$ *(necessarily unique and mono) s.t.* $m_1 = m; m_2$*. A* **subobject** *(of* A*) is the equivalence class* $[m]$ *of a mono* $m{:}d \hookrightarrow A$ *w.r.t. the equivalence relation on monos induced by the preorder* $\leq$*. The* **inclusion order** $\leq$ *on subobjects (induced by the preorder* $\leq$ *on monos) is defined by* $[m_1] \leq [m_2]$ *iff* $m_1 \leq m_2$ *(and does not depend on the choice of witness). On subobjects we define the following operations:*

- **composition** $_; n$ *with a mono* $n{:}A \hookrightarrow B$*, i.e.* $[m]; n = [m; n]$ *(provided* m *has codomain* A*);*
- **inverse image** *along a morphism* $f{:}B \to A$*, i.e.* $f^*[m] = [n]$*, where*



  (provided m *has codomain* A *and a pullback exists).*

These two operations are well-defined, i.e. they do not depend on the choice of witness, and preserve the inclusion order, i.e. $[m_1] \leq [m_2]$ *implies* $[m_1]; n \leq [m_2]; n$ *and* $f^*[m_1] \leq f^*[m_2]$*.*

In general, the interpretation of a formula must be a *well-behaved* subobjects. This is achieved by restricting to subobjects induced by a *dominion*.

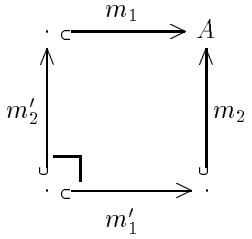
Definition 3.12 ([RR88]) *A* **dominion** $\mathcal{M}$ *over* $\mathcal{C}$ *is a class of display maps where all displays are monos and closed w.r.t. identities and composition, i.e.*

- $\mathrm{id}_C : C \to C$ *is in* $\mathcal{D}$ *for all* $C \in \mathcal{C}$
- *if* $d_1 : A \to B$ *and* $d_2 : B \to C$ *are in* $\mathcal{D}$*, then* $d_1; d_2$ *is in* $\mathcal{D}$

The *topos-theoretic* semantics of a typed predicate logic in a dominion $\mathcal{M}$ over a category $\mathcal{C}$ with finite products is defined in terms of the *hyperdoctrine* semantics by giving a $\mathcal{C}$-indexed meet semi-lattice. In fact, the indexed meet semi-lattices induced by dominions are particularly well-behaved (e.g. with them one can deal more easily with equality).

Definition 3.13 *A dominion* $\mathcal{M}$ *over* $\mathcal{C}$ *induces a* $\mathcal{C}$*-indexed meet semi-lattice* $\mathcal{M}$ *(of* $\mathcal{M}$*-subobjects) defined as follows:*

- *the fiber* $\mathcal{M}[A]$ *over* A *is the poset of subobjects* $[m]$ *of* A *witnessed by monos in* $\mathcal{M}$ *with the inclusion order as partial order, with meet-semilattice operations given by* $\top = [\mathrm{id}_A]$ *and* $[m_1] \wedge [m_2] = [m_2'; m_1]$*, where*

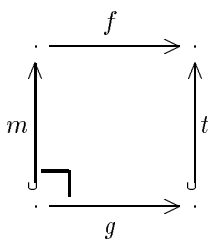


- *substitution* $\mathcal{M}[f]$ *along* $f : B \to A$ *is the inverse image operation* $f^*_$ *restricted to* $\mathcal{M}[A]$ *(*f^* *preserves* $\top$ *and* $\wedge$ *by general properties of pullbacks).*

In general, for interpreting logical constants one must require additional properties on dominions. Often one can express these additional properties in terms of the induced indexed meet semi-lattices, but sometimes it is more convenient to express them directly in terms of dominions.

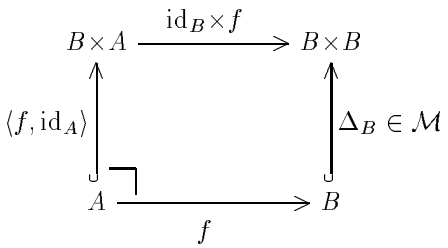
Definition 3.14 *Given a dominion* $\mathcal{M}$ *over a category* $\mathcal{C}$*, we say that:*

- $\mathcal{M}$ **has equalities** *iff (*$\mathcal{C}$ *has finite products and)* $\Delta_A : A \hookrightarrow A \times A \in \mathcal{M}$ *for every* $A \in \mathcal{C}$
- $\mathcal{M}$ *is closed w.r.t.* **universal quantification** *along maps in* $\mathcal{D}$ *(where* $\mathcal{D}$ *is a class of display maps over* $\mathcal{C}$*) iff the induced* $\mathcal{C}$*-indexed meet semi-lattice is closed w.r.t. universal quantification along maps in* $\mathcal{D}$*.*
- $t \in \mathcal{M}$ *is a* **dominance** *iff for all* $m \in \mathcal{M}$ *exist unique* f *s.t. for some* g



Lemma 3.15 *If* $\mathcal{M}$ *has equalities and is closed w.r.t. implication and universal quantification along projections, then it is closed w.r.t. universal quantification along any morphism in* $\mathcal{C}$*.*

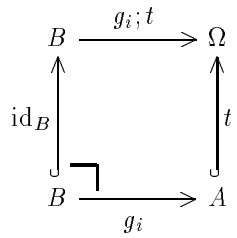
Proof It is enough to show that every $f : A \to B$ in $\mathcal{C}$ is decomposable in a mono m in $\mathcal{M}$ followed a first projection. Take $m = \langle f, \mathrm{id}_A \rangle : A \hookrightarrow B \times A$ and $pi_1 : B \times A \to B$. Then $f = m; \pi_1$, and $m \in \mathcal{M}$, because



∎

Lemma 3.16 *If $t: A \hookrightarrow \Omega$ is a dominance, then A is a terminal object of $\mathcal{C}$.*

Proof Given $B \in \mathcal{C}$, exists $g: B \to A$, because $\mathrm{id}_B \in \mathcal{M}$. On the other hand, if $g_1, g_2: B \to A$, then



because t is mono. Therefore, $g_1;t = g_2;t$ because of the unicity property for dominances, and $g_1 = g_2$ because t is mono. ∎

Proposition 3.17 *Given a dominion $\mathcal{M}$ over $\mathcal{C}$, let $\mathcal{P}$ be the induced $\mathcal{C}$-indexed meet semi-lattice, then*

- *$\mathcal{P}$ has equalities, when $\mathcal{M}$ has equalities;*
- *$\mathcal{P}$ is closed w.r.t. implication iff it is closed w.r.t. universal quantification along maps in $\mathcal{M}$;*
- *$[t] \in \mathcal{P}[\Sigma]$ is a skeletal generic predicate iff $t \in \mathcal{M}$ is a dominance.*

Proof If $\mathcal{M}$ has equalities, then $=_A$ is given by $[\Delta_A] \in \mathcal{P}[A \times A]$.
We show only the correspondence between pseudo-complement and universal quantification. If $[m], [m'] \in \mathcal{P}[A]$, then $[m] \supset [m'] = \forall_m(m^*[m'])$. If $m: A \hookrightarrow B \in \mathcal{M}$ and $[m'] \in \mathcal{P}[A]$, then $\forall_m([m']) = [m] \supset [m'; m]$. ∎

3.3.1 Topos-theoretic interpretation

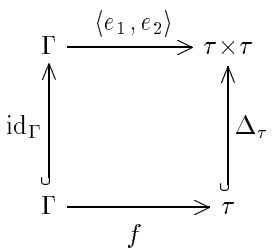
Given a category $\mathcal{C}$ with finite products and a dominion $\mathcal{M}$ over $\mathcal{C}$, the interpretation of formulas in the induced $\mathcal{C}$-indexed meet semi-lattice $\mathcal{M}$ is defined as in Section 3.2.1, provided the dominion $\mathcal{M}$ satisfies the additional properties corresponding to those required for $\mathcal{C}$-indexed meet semi-lattices. if $\mathcal{M}$ has equalities, then the interpretation of equality predicates is defined by

$\Gamma \vdash e_1: \tau$	$=$	$e_1: \Gamma \to \tau$
$\Gamma \vdash e_2: \tau$	$=$	$e_2: \Gamma \to \tau$
$\Gamma \vdash e_1 =_\tau e_2$ prop	$=$	$\langle e_1, e_2 \rangle^*[\Delta_\tau] \in \mathcal{M}[\Gamma]$

and there is a close correspondence between equational judgements and equality predicates.

Proposition 3.18 *If $\Gamma \vdash e_i: \tau$ $(i = 1, 2)$ are well-formed terms and $\mathcal{M}$ has equalities, then $\Gamma \vdash e_1 = e_2: \tau$ is true in $\mathcal{C}$ iff $\Gamma \vdash \emptyset \Longrightarrow e_1 =_\tau e_2$ is true in $\mathcal{M}$.*

Proof Let $e_i: \Gamma \to \tau$ be the interpretation of $\Gamma \vdash e_i: \tau$, then we have to prove that $e_1 = e_2$ iff $\langle e_1, e_2 \rangle^*[\Delta_\tau] = [\mathrm{id}_\Gamma]$, or equivalently



is a pullback for some f (necessarily unique). Therefore, the claim follows from basic properties of pullbacks. ∎

4 Semantic for necessity

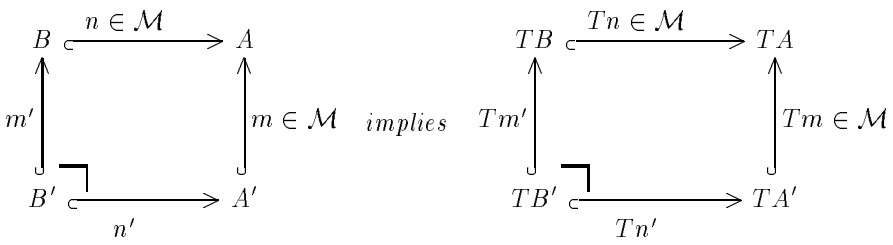
We define a categorical semantics of necessity, which provides criteria for judging logical rules for necessity (by establishing their soundness and possibly completeness). We do this in the general setting of a category $\mathcal{C}$ with finite products equipped with a dominion $\mathcal{M}$ (to interpret formulas according to the topos-theoretic approach) and a strong monad $(T, \mathrm{t}, \eta, \mu)$ (to interpret computational types according to [Mog91]), by analogy with the set-theoretic semantics of section 1.2. we introduce various properties of T in relation to $\mathcal{M}$, and prove soundness of the rules for necessity (see sections 2.3.7 and 2.5.3) in strong monads satisfying some of these additional properties.

4.1 Properties of strong monads w.r.t. a dominion

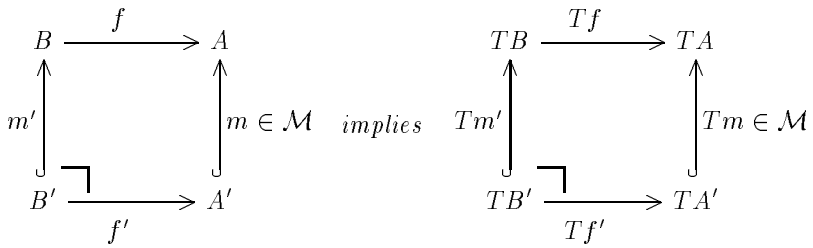
In this section we define additional properties for (strong) monads over a category $\mathcal{C}$ with (finite products and) a dominion $\mathcal{M}$.

Definition 4.1 *Given a category $\mathcal{C}$ and a dominion $\mathcal{M}$ over it, we say that an endofunctor $T\colon \mathcal{C} \to \mathcal{C}$ is:*

- **mono preserving** *(w.r.t. $\mathcal{M}$) iff $m \in \mathcal{M}$ implies $Tm \in \mathcal{M}$ (up to equivalence),*
- **meet preserving** *(w.r.t. $\mathcal{M}$) iff*

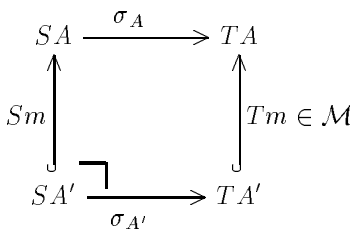


- **inverse image preserving** *(w.r.t. $\mathcal{M}$), $\mathcal{M}$-functor for short, iff*

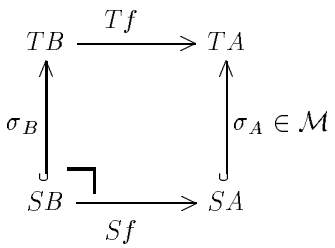


We say that a natural transformation $\sigma\colon S \to T$ between (mono preserving) endofunctors is

- **$\mathcal{M}$-cartesian** *iff for every $m\colon A' \to A$ in $\mathcal{M}$*



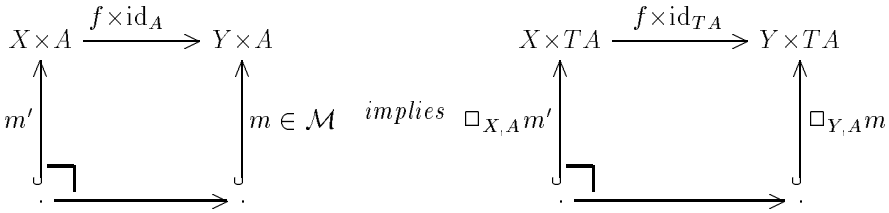
- $\mathcal{M}$-**natural** *iff for every* $f\colon B \to A$ *in* $\mathcal{C}$



Lemma 4.2 *$\mathcal{M}$-functors are closed w.r.t. composition, and the same is true for mono/meet preserving endofunctors. $\mathcal{M}$-cartesian ($\mathcal{M}$-natural) transformations are closed w.r.t. vertical composition. $\mathcal{M}$-cartesian ($\mathcal{M}$-natural) transformations between $\mathcal{M}$-functors are closed w.r.t. horizontal composition.*

The properties defined above for endofunctors and natural transformations specialise (in the obvious way) to the case of strong endofunctors and strong transformations. However, in this setting we can consider a further property:

Definition 4.3 *Given a category $\mathcal{C}$ with finite products and a dominion $\mathcal{M}$ over it, we say that a strong endofunctor (T, t) is* **strongly mono preserving** *(w.r.t. $\mathcal{M}$) iff (it is mono preserving and)*



where $\Box_{Y,A}m$ is a mono of $\mathcal{M}$ (unique up to equivalence) s.t. $[\Box_{Y,A}m] = \mathrm{t}^*_{Y,A}[Tm]$.

In general strongly mono preserving endofunctors are not closed w.r.t. composition.

Definition 4.4 *Given a category $\mathcal{C}$ with finite products, a dominion $\mathcal{M}$ and a strong endofunctor (T, t) which is mono preserving (w.r.t. $\mathcal{M}$). we can interpret necessity as follows:*

$\Gamma, x\colon A \vdash \phi$ prop	$=$	$[m] \in \mathcal{M}[\Gamma \times A]$
$\Gamma \vdash e\colon TA$	$=$	$f\colon \Gamma \to TA$
$\Gamma \vdash [x \Leftarrow e]\phi$ prop	$=$	$\langle \mathrm{id}_\Gamma, f\rangle^*[\Box_{\Gamma,A}] \in \mathcal{M}[\Gamma]$

Unfortunately, we do not have an equally general semantics for the evaluation predicate and possibility, it is only by going to full HOL that we are able to define their semantics.
Although the definition of necessity requires only that T is mono preserving, it is only when (T, t) is strongly mono preserving that necessity is well-behaved.

Proposition 4.5 *If (T, t) is strongly mono preserving (w.r.t. $\mathcal{M}$) and substitution holds for $\Gamma, x\colon A \vdash \phi$ prop and $\Gamma \vdash e\colon TA$, then it holds for $\Gamma \vdash [x \Leftarrow e]\phi$ prop. Where* **substitution holds** *for $x_1\colon \tau_1, \ldots, x_n\colon \tau_n \vdash e\colon \tau/\phi$ prop means that*

$x_1\colon \tau_1, \ldots, x_n\colon \tau_n \vdash e\colon \tau$	$=$	$f\colon \tau_1 \times \ldots \times \tau_n \to \tau$
$x_1\colon \tau_1, \ldots, x_n\colon \tau_n \vdash \phi$ prop	$=$	$[m] \in \mathcal{M}[\tau_1 \times \ldots \times \tau_n]$
$\Gamma \vdash e_i\colon \tau_i$	$=$	$f_i\colon \Gamma \to \tau_i \quad (i = 1, \ldots, n)$
$\Gamma \vdash [\overline{e}/\overline{x}]e\colon \tau$	$=$	$\langle f_1, \ldots, f_n\rangle; f\colon \Gamma \to \tau$
$\Gamma \vdash [\overline{e}/\overline{x}]\phi$ prop	$=$	$\langle f_1, \ldots, f_n\rangle^*[m] \in \mathcal{M}[\Gamma]$

for every $\Gamma \vdash e_i\colon \tau_i$ $(i = 1, \ldots, n)$.

Proposition 4.6 *The following implications (among properties of strong endofunctors) hold:*

- *$\mathcal{M}$-functor $\supset$ meet preserving/strongly mono preserving,*
- *meet preserving/strongly mono preserving $\supset$ mono preserving.*

Proof We prove only that if (T,t) is a strong endofunctor and T is an $\mathcal{M}$-functor, then (T,t) is strongly mono preserving. The other implications are immediate. Given $f\colon X \to Y$, $[m] \in \mathcal{M}[Y\times A]$ and $[m'] \in \mathcal{M}[X\times A]$ s.t. $[m'] = (f\times\mathrm{id}_A)^*[m]$, we have to show that $[\Box_{X,A}m'] = (f\times\mathrm{id}_{TA})^*(\Box_{Y,A}m)$. Since $[\Box_{X,A}m'] = \mathrm{t}^*_{X,A}[Tm']$ (by definition of $\Box$) and $[Tm'] = T(f\times\mathrm{id}_A)^*[Tm]$ (because T is an $\mathcal{M}$-functor), the equation above rewrites to $(\mathrm{t}_{X,A};T(f\times\mathrm{id}_A))^*[Tm] = ((f\times\mathrm{id}_{TA});\mathrm{t}_{Y,A})^*[Tm]$, which is true by naturality of t. ∎

Definition 4.7 *Given $\mathcal{C}$, $\mathcal{M}$, (T,t) as in Definition 4.3 and a class of display maps $\mathcal{D}$ over $\mathcal{C}$, if T is mono preserving and $\mathcal{M}$ is closed w.r.t. universal quantification along maps in $\mathcal{D}$, then we say that necessity (for T)* **commutes with** *universal quantification (along $\mathcal{D}$) iff*

$$\begin{array}{ccc} \mathcal{M}[X\times A] & \xrightarrow{\forall_{d\times\mathrm{id}_A}} & \mathcal{M}[Y\times A] \\ \Big\downarrow{\scriptstyle \Box_{X,A}} & & \Big\downarrow{\scriptstyle \Box_{Y,A}} \\ \mathcal{M}[X\times TA] & \xrightarrow[\forall_{d\times\mathrm{id}_{TA}}]{} & \mathcal{M}[Y\times TA] \end{array}$$

for every $X,Y,A \in \mathcal{C}$ and $d\colon X \to Y \in \mathcal{D}$

4.2 Soundness of rules for necessity

In this section we prove soundness of the rules given in Sections 2.3.7 and 2.5.3.

Theorem 4.8 *Given a category $\mathcal{C}$ with finite products, a dominion $\mathcal{M}$ over $\mathcal{C}$ and a strong monad (T,t,η,μ) over $\mathcal{C}$ s.t. (T,t) is strongly mono preserving (w.r.t. $\mathcal{M}$), we have the following soundness results:*

- *the rules ($\Box$-$\top$*,$\Box$-$\Longrightarrow$,$\Box$-T,$\Box$-t*, $\Box$-η,$\Box$-μ) are sound;*
- *($\Box$-$\wedge$*) is sound, provided T is meet preserving;*
- *the rule ($\Box$-T*) is sound, provided T is an $\mathcal{M}$-functor;*
- *the rule ($\Box$-η*) is sound, provided η is $\mathcal{M}$-cartesian;*
- *the rule ($\Box$-μ*) is sound, provided μ is $\mathcal{M}$-cartesian and T is an $\mathcal{M}$-functor;*
- *the rule ($\Box$-$\supset$*) is sound, provided $\mathcal{M}$ is closed w.r.t. universal quantification along $m \in \mathcal{M}$ and necessity commutes with it;*
- *the rule ($\Box$-$\forall$*) is sound, provided $\mathcal{M}$ is closed w.r.t. universal quantification along projections and necessity commutes with it;*
- *the rule ($\Box$-=) is sound, provided $\mathcal{M}$ has equalities;*
- *the rule* (Comp-T) *is sound, provided $\mathcal{C}$ is a topos and $\mathcal{M}$ is the dominion of all monos.*

Proof see Appendix ∎

We consider further examples of computational monads, and check whether they satisfy those additional properties which ensure soundness of the various inference rules for necessity.

Example 4.9 Let $\mathcal{C}$ be a topos and $\mathcal{M}$ the dominion of all monos, we consider some strong monads $(T, \mathrm{t}, \eta, \mu)$ over $\mathcal{C}$. For each of them we give the strongest properties (among those defined above) satisfied by (T, t), η and μ, and the interpretation of necessity (see also Example 1.1).

- $T(A) = A + E$ **exceptions**. T is an $\mathcal{M}$-functor, η is $\mathcal{M}$-natural, μ is $\mathcal{M}$-cartesian
  $[x \Leftarrow c]\phi$ iff $\forall x{:}A.[x] = c \supset \phi)$.
- $T(A) = (A \times S)^S$ **side-effects** (S non trivial). T is an $\mathcal{M}$-functor, η is $\mathcal{M}$-natural, (μ is not $\mathcal{M}$-cartesian)
  $[x \Leftarrow c]\phi$ iff $\forall s, s'{:}S, x{:}A.\langle x, s'\rangle = c(s) \supset \phi)$.
- $T(A) = \mathcal{P}_{fin}(A)$ **non-determinism**. T is an $\mathcal{M}$-functor, η is $\mathcal{M}$-cartesian, μ is $\mathcal{M}$-cartesian
  $[x \Leftarrow c]\phi$ iff $\forall x{:}A.x \in_A c \supset \phi)$.
- $T(A) = \Omega^{(\Omega^A)}$ **continuations**. T is meet and strongly mono preserving, η is $\mathcal{M}$-cartesian, (μ is not $\mathcal{M}$-cartesian)
  $[x \Leftarrow c]\phi$ iff $\forall k, k'{:}\Omega^A.(\forall x{:}A.\phi \supset kx =_\Omega k'x) \supset ck =_\Omega ck'$.
  These properties of T continue to hold, when Ω is replaced by an R s.t. $\Omega \lhd R$.
- $T(A) = A \times N$ **complexity** (where N is a monoid). T is an $\mathcal{M}$-functor, η is not $\mathcal{M}$-natural, μ is $\mathcal{M}$-cartesian
  $[x \Leftarrow c]\phi$ iff $\forall n{:}N, x{:}A.\langle x, n\rangle = c \supset \phi)$.
- $T(A) = 1$ **trivial**. T is an $\mathcal{M}$-functor, η is not $\mathcal{M}$-cartesian, μ is $\mathcal{M}$-natural
  $[x \Leftarrow c]\phi$ iff $\top$.

In all examples above, except for continuations, necessity commutes with implication and universal quantification, i.e. ($\Box$-$\supset$*) and ($\Box$-$\forall$*) are valid. In the case of continuations, one can find counter-examples to both axioms.

Example 4.10 Monads for continuations $T(A) = R^{(R^A)}$ provide quite a lot of counter-examples to preservation of monos.

- In the Effective Topos (see [Hyl91]) there are several $\Sigma \subset \Omega$ s.t. $2 \subseteq \Sigma \cong \Sigma^\Omega$. If R is any of such Σ, then $T(2 \hookrightarrow \Omega)$ cannot be monic.
- In the category of posets and monotonic functions there are three possible dominances: $1 \in 2$ for decidable subobjects, $\top \in \Sigma$ for open subobjects and $\bot \in \Sigma$ for closed subobjects (where 2 is the flat poset with two elements 0 and 1 and Σ is the poset with two elements linearly ordered $\bot < \top$).
  - If R is 2, then $2 \hookrightarrow 2_\bot$ is an open subobject, but $T2 \not\hookrightarrow T(2_\bot)$, since $|T2| = 16$ and $|T(2_\bot)| = 4$. However, T preserves decidable subobjects.
  - If R is Σ, then $2 \hookrightarrow \Sigma$ is a subobject, but $T2 \not\hookrightarrow T\Sigma$, since $|T2| = 6$ and $|T\Sigma| = 4$. However, T preserves regular subobjects.

  In the category of cpos and continuous functions there are similar results for $2^{(2^X)}$, but not for $TX = \Sigma^{(\Sigma^X)}$. In fact, T does not preserves regular subobjects (i.e. inclusive subsets). One can show that T preserves subspaces (i.e. $X \hookrightarrow Y$ s.t. the Scott topology on X is the restriction of that on Y). However, in the category of cpos (unlike that of topological spaces) subspaces are properly contained in regular subobjects, and do not form a dominion.

Example 4.11 Let $\mathcal{W}$ be the category of **state shapes** (see [Ole85]), i.e.

- an object is a **non-empty** set W,
  if $\emptyset$ were in $\mathcal{W}$, then ($\Box$-$\supset$*) and ($\Box$-$\forall$*) would not be valid

- a morphism from W to X is a pair (l,u), where $l\colon X \to W$ (l for lookup) and $u\colon W\times X \to X$ (u for update) satisfy the equations
  $u(l(x),x) = x$, $l(u(w,x)) = w$ and $u(w,u(w',x)) = u(w,x)$
  in particular, $X \cong W\times V$ (for some V) and l is the first projection
- composition of $(l_1,u_1)\colon W \to X$ followed by $(l_2,u_2)\colon X \to Y$ is
  the pair $(l,u)\colon W \to Y$ s.t. $l(y) = l_1(l_2(y))$ and $u(w,y) = u_2(u_1(w,l_2(y)),y)$.

Let (T,t) be the strong functor (which can be easily extended to a strong monad) over the topos $\mathbf{Set}^{\mathcal{W}}$ s.t.

- for every $A \in \mathbf{Set}^{\mathcal{W}}$ and $(l,u)\colon W \to X$ in $\mathcal{W}$
  - $TAW = (A(W)\times W)^W$,
  - $TA(l,u)cx = \langle A(l,u)a, u(w,x)\rangle$, where $c \in TAW$, $x \in X$ and $\langle a,w\rangle = c(l(x))$
- for every $\sigma\colon A \dot{\to} B$ in $\mathbf{Set}^{\mathcal{W}}$ and $W \in \mathcal{W}$
  $(T\sigma)_W\, cw = \langle \sigma_W a, w'\rangle$, where $c \in TAW$, $w \in W$ and $\langle a,w'\rangle = c(w)$
- for every $A,B \in \mathbf{Set}^{\mathcal{W}}$ and $W \in \mathcal{W}$
  $\mathrm{t}_{A,B,W}(a,c)w = \langle\langle a,b\rangle, w'\rangle$, where $a \in AW$, $c \in TBW$, $w \in W$ and $\langle b,w'\rangle = cw$

Since T preserves pullbacks, necessity can be interpreted according to the topos-theoretic semantics

$W\|\!\vdash [a \Leftarrow c]p(x,a)$ iff $\forall w,w'\colon W, a\colon AW.\langle a,w'\rangle = c(w) \supset W\|\!\vdash p(x,a)$

for every $X,A \in \mathbf{Set}^{\mathcal{W}}$, p predicate over $B\times A$, $W \in \mathcal{W}$, $c \in TAW$ and $x \in XW$.
Necessity commutes with implication and universal quantification, and the interpretation of the evaluation predicate and possibility is

- $W\|\!\vdash \langle a \Leftarrow c\rangle p(x,a)$ iff $\exists w,w'\colon W, a\colon AW.\langle a,w'\rangle = c(w) \wedge W\|\!\vdash p(x,a)$
- $W\|\!\vdash c\Downarrow a$ iff $\exists w,w'\colon W.\langle a,w'\rangle = c(w)$.

One can consider a monad for local variables better than T, by working with *parametric* functors and *parametric* natural transformations (see [OT93]). This monad seems to enjoy properties similar to those of T, but it is not known whether the category of parametric functors is a topos.

Example 4.12 Let $\mathcal{I}$ be the category of finite cardinals and injective maps, and let (T,t) be the strong functor over the topos $\mathbf{Set}^{\mathcal{I}}$ (see [Mog89]) s.t.

- for every $A \in \mathbf{Set}^{\mathcal{I}}$ and $f\colon m \to n$ in $\mathcal{I}$
  - $TAm = \Sigma k\colon N.A(m+k)$
  - $TAf\langle k,a\rangle = \langle k, A(f+k)a\rangle$ where $\langle k,a\rangle \in TAm$
- for every $\sigma\colon A \dot{\to} B$ in $\mathbf{Set}^{\mathcal{I}}$ and $m \in \mathcal{I}$
  $(T\sigma)_m\langle k,a\rangle = \langle k, \sigma_{m+k}a\rangle$, where $\langle k,a\rangle \in TAm$
- for every $A,B \in \mathbf{Set}^{\mathcal{I}}$ and $m \in \mathcal{I}$
  $\mathrm{t}_{A,B,m}(a,\langle k,b\rangle) = \langle k, \langle A(m \hookrightarrow m+k)a, b\rangle\rangle$, where $a \in Am$ and $\langle k,b\rangle \in TBm$

Since T preserves pullbacks, necessity can be interpreted according to the topos-theoretic semantics

$m\|\!\vdash [a \Leftarrow c]p(x,a)$ iff
$\forall k\colon N, a\colon A(m+k).\langle k,a\rangle = c \supset m+k\|\!\vdash p(X(m \hookrightarrow m+k)x, a)$

for every $X,A \in \mathbf{Set}^{\mathcal{I}}$, p predicate over $B\times A$, $m \in \mathcal{I}$, $c \in TAm$ and $x \in Xm$.
Necessity commutes with neither implication nor universal quantification, and the interpretation of the evaluation predicate and possibility is

- $m\|\!\!-\langle a \Leftarrow c\rangle\, p(x,a)$ iff $\exists a\colon A(m).\langle 0,a\rangle = c \wedge m\|\!\!-p(x,a)$
- $m\|\!\!-c\Downarrow a$ iff $\exists a\colon A(m).\langle 0,a\rangle = c$.

Unfortunately, the interpretation of possibility cannot express observational judgements such as termination, i.e. $m\|\!\!-c\Downarrow$ iff $\exists k\colon N, a\colon A(m+k).\langle k,a\rangle = c$.
One way to avoid this problem is to consider the sub-topos $(\mathbf{Set}^{\mathcal{I}})_{\neg\neg}$ of sheaves for the double-negation topology, which coincide with pullback preserving functors from $\mathcal{I}$ to $\mathbf{Set}$. In fact, (T,t) cuts down to a strong functor over $(\mathbf{Set}^{\mathcal{I}})_{\neg\neg}$, necessity commutes with implication (but not with universal quantification), and the interpretation of the evaluation predicate and possibility becomes

- $m\|\!\!-\langle a \Leftarrow c\rangle\, p(x,a)$ iff
  $\exists k\colon N, a\colon A(m+k).\langle k,a\rangle = c \wedge m+k\|\!\!-p(X(m \hookrightarrow m+k)x, a)$
- $m\|\!\!-c\Downarrow a$ iff $\exists k\colon N, a\colon A(m+k).\langle k, A(m \hookrightarrow m+k)a\rangle = c$.

One can consider monads for dynamic allocation better than T, by replacing coproducts with a different form of colimit (see [Mog89, Pit92a]). These monads seem to enjoy properties similar to those of T, when restricted to $(\mathbf{Set}^{\mathcal{I}})_{\neg\neg}$. There are also attempts to define monads for dynamic allocation over the category of parametric functors, but it is not clear whether they cut down to something like $(\mathbf{Set}^{\mathcal{I}})_{\neg\neg}$.

5 Completeness results

In this section we prove completeness results for some typed predicate logics with necessity (introduced in Section 2) w.r.t. the topos-theoretic semantics (defined in Section 3.3). First we establish completeness w.r.t. the hyperdoctrine semantics. Then we define suitable constructions for transforming hyperdoctrine models into topos-theoretic models with the same theory, so that one can turn completeness results w.r.t. the hyperdoctrine semantics into completeness results w.r.t. the topos-theoretic semantics. In summary, we have the following completeness results:

Theorem 5.1

1. *$ML_T[\top,\wedge,\Box]$ is complete w.r.t. topos-theoretic interpretations in categories $\mathcal{C}$ with finite products equipped with a dominion $\mathcal{M}$ and a strong monad (T,t,η,μ) s.t. T is a $\mathcal{M}$-functor;*
2. *$ML_T[\top,\wedge,=,\Box,\Box\text{-}=]$ is complete w.r.t. topos-theoretic interpretations in categories $\mathcal{C}$ with finite products equipped with a dominion $\mathcal{M}$ and a strong monad (T,t,η,μ) s.t. $\mathcal{M}$ has equalities and T is a $\mathcal{M}$-functor;*
3. *$HML_T[\Box,\Box\text{-}=,\text{Comp-}P,\text{Comp-}T]$ is complete w.r.t. topos-theoretic interpretations in toposes $\mathcal{C}$ equipped with a strong monad (T,t,η,μ) s.t. T is a $\mathcal{M}$-functor, where $\mathcal{M}$ is the dominion of all monos.*

By restricting the topos-theoretic interpretations to those $\mathcal{C}$ with (enough) exponentials and/or those $\mathcal{M}$ with (enough) implications and universal quantifications along first projections, then one has completeness also for:
$ML_{T\Rightarrow}[\top,\wedge,\supset,\forall,\Box]$, $ML_{T\Rightarrow}[\top,\wedge,=,\supset,\forall,\Box,\text{-}\lambda,\Box\text{-}=]$ *and*
$HML_{T\Rightarrow}[\Box,\Box\text{-}=,\text{Comp-}P,\text{Comp-}\Rightarrow,\text{Comp-}T]$.

Proof By completeness w.r.t. hyperdoctrine interpretations, and then by applying the properties of the constructions described in Section 5.2.1, 5.2.2 and 5.2.3. ∎

5.1 Completeness w.r.t. the hyperdoctrine semantics

The simplest way of proving completeness of a logic w.r.t. a class of models, is to show that for every theory Th there is a *generic model* $M(Th)$, whose theory is exactly Th. In this section we adapt the construction of a **classifying hyperdoctrine** $\mathcal{P}$ and a **generic model** $M(Th)$ for a first order theory Th (see [Pit89]) to theories in one of the typed predicate logics considered in Section 2. The key property of $M(Th)$ is that a well-formed equation or sequent is in Th iff it is true in $M(Th)$, while the classifying hyperdoctrine $\mathcal{P}$ is only instrumental to the definition of $M(Th)$.

5.1.1 Generic models

Given a theory Th for $ML[\top,\wedge](\Sigma)$, one can define an indexed meet semi-lattice $\mathcal{P}\colon\mathcal{C}^{op} \to \mathbf{PoSet}$ from the syntax and prove that it satisfies the necessary properties, by appealing to closure of Th w.r.t. the inference rules of $ML[\top,\wedge](\Sigma)$, namely:

- $\mathcal{C}$ is the category with products induced by the equations of Th, i.e. objects are well-formed types, morphisms from τ_1 to τ_2 are equivalence classes of well-formed terms $x\colon\tau_1 \vdash e\colon\tau_2$ modulo provable equality, i.e. $x\colon\tau_1 \vdash e_1 = e_2\colon\tau_2$ is in Th, and composition is given by syntactic substitution.

  The terminal object of $\mathcal{C}$ is the unit type 1, and the binary product $\tau_1\times\tau_2$ is the product type $\tau_1\times\tau_2$ with projections given by $[x\colon\tau_1\times\tau_2 \vdash \pi_i(x)\colon\tau_i]$.

- $\mathcal{P}[\tau]$ is the meet semi-lattice of formulas over τ, i.e. elements are equivalence classes of well-formed formulas $x\colon\tau \vdash \phi$ prop modulo provable equivalence, i.e. $x\colon\tau \vdash \phi_1 \iff \phi_2$ is in Th, and the partial order is given by provable entailment, i.e. $x\colon\tau \vdash \phi_1 \implies \phi_2$ is in Th.

  The top element of $\mathcal{P}\tau$ is $[x\colon\tau \vdash \top \text{ prop}]$ and the binary meet $[x\colon\tau \vdash \phi_1 \text{ prop}] \wedge [x\colon\tau \vdash \phi_2 \text{ prop}]$ is $[x\colon\tau \vdash \phi_1 \wedge \phi_2 \text{ prop}]$.

- $f^*\colon\mathcal{P}[\tau_2] \to \mathcal{P}[\tau_1]$, where $f = x\colon\tau_1 \vdash e\colon\tau_2$, is given by syntactic substitution, i.e. $f^*[x\colon\tau_2 \vdash \phi \text{ prop}] = [x\colon\tau_1 \vdash [e/x]\phi \text{ prop}]$

When Th is a theory for an extension of $ML[\top,\wedge](\Sigma)$ (obtained by adding functional types, implication, universal quantification and/or equality predicates) or for $HML(\Sigma)$ one can proceeds similarly, by showing that the indexed meet semi-lattice $\mathcal{P}$ (defined above) is equipped with additional structure, suitable for interpreting the typed predicate logic under consideration, namely:

- if Th has functional types, then the exponential ${\tau_2}^{\tau_1}$ in $\mathcal{C}$ is the functional type $\tau_1 \Rightarrow \tau_2$ with evaluation given by $[x\colon(\tau_1\Rightarrow\tau_2)\times\tau_1 \vdash \pi_1(x)(\pi_2 x)\colon\tau_2]$.

- if Th has implication, then the pseudo-complement $[x\colon\tau \vdash \phi_1 \text{ prop}] \supset [x\colon\tau \vdash \phi_2 \text{ prop}]$ in $\mathcal{P}[\tau]$ is given by $[x\colon\tau \vdash \phi_1 \supset \phi_2 \text{ prop}]$

- if Th has universal quantification, then the universal quantification along $\pi_1\colon\tau_1\times\tau \to \tau_1$ of $[x\colon\tau_1\times\tau \vdash \phi \text{ prop}]$ is given by $[x\colon\tau_1 \vdash \forall y\colon\tau.[\langle x,y\rangle/x]\phi \text{ prop}]$

- if Th has equality over τ, then $\mathcal{P}$ has equality over τ given by $[x\colon\tau\times\tau \vdash \pi_1(x) =_\tau \pi_2(x) \text{ prop}]$

- if Th is a theory for $HML(\Sigma)$, then $[X\colon\Omega \vdash X \text{ prop}]$ is a generic predicate, and the exponential Ω^τ in $\mathcal{C}$ is the powerset type $P\tau$ with evaluation given by $[x\colon(P\tau)\times\tau \vdash (\pi_2 x) \in_\tau (\pi_1 x)\colon\Omega]$.

The generic model $M(Th)$ of Th is obtained by choosing a suitable interpretation in $\mathcal{P}$ of the symbols in Σ, namely: $A \in \Sigma^t$ is interpreted by $A \in \mathcal{C}$, $f \in \Sigma^f_{\tau_1,\tau_2}$ is interpreted by the morphism $[x\colon\tau_1 \vdash f(x)\colon\tau_2]\colon\tau_1 \to \tau_2$, and $p \in \Sigma_\tau$ is interpreted by $[x\colon\tau \vdash p(x) \text{ prop}] \in \mathcal{P}[\tau]$. In this way we can ensure the following properties:

- the interpretation of a type τ is τ,

- the interpretation of a context $\Gamma \equiv x_1\colon\tau_1,\ldots,x_n\colon\tau_n$ is the type $\tau_\Gamma = 1\times\tau_1\times\ldots\times\tau_n$ (where association is on the left),

- the interpretation of a term $\Gamma \vdash e{:}\tau$ is $[x{:}\tau_\Gamma \vdash [\overline{e}/\overline{x}]e{:}\tau]{:}\tau_\Gamma \to \tau$ (intuitively e_i selects the i-component of x),
- the interpretation of a formula $\Gamma \vdash \phi$ prop is $[x{:}\tau_\Gamma \vdash [\overline{e}/\overline{x}]\phi \text{ prop}] \in \mathcal{P}[\tau_\Gamma]$.

From these properties one can prove that a well-formed equational or entailment judgement is in Th iff it is true in $M(Th)$.

5.1.2 Generic models for necessity

In this section we extend the construction of generic models to theories involving necessity. However, we must first specify the additional structure on indexed meet-semilattices for interpreting typed predicate logics with necessity.

Definition 5.2 *Given a category $\mathcal{C}$ with finite products, a $\mathcal{C}$-indexed meet semi-lattice $\mathcal{P}$, and a strong monad $(T, \mathrm{t}, \eta, \mu)$ over $\mathcal{C}$, then a* **box-modality** $\Box$ *(for T over $\mathcal{P}$) is a family of monotonic functions $\langle \Box_A{:}\mathcal{P}[A] \to \mathcal{P}[TA] | A \in \mathcal{C}\rangle$ s.t.*

$\Box\top^*$ $\Box_A \top = \top$

$\Box\eta$ $a \le \eta^*(\Box_A a)$

$\Box\mu$ $\Box_{TA}(\Box_A a) \le \mu^*(\Box_A a)$

$\Box\mathrm{t}$ $a \times (\Box_B b) \le \mathrm{t}^*_{A,B}(\Box_{A\times B}(a \times b))$, *where* $a \times b$ *is* $(\pi_1^* a) \wedge (\pi_2^* b) \in \mathcal{P}[A{\times}B]$ *when* $a \in \mathcal{P}[A]$ *and* $b \in \mathcal{P}[B]$.

$\Box\wedge^*$ $\Box_A(a \wedge b) = (\Box_A a) \wedge (\Box_A b)$

$\Box T^*$ $\Box_B(f^* a) = (Tf)^*(\Box_A a)$, *where* $f{:}B \to A$

If $\Box$ is a box-modality for T over $\mathcal{P}$, then the hyperdoctrine interpretation of Section 3.2.1 can be extended to formulas with necessity by

$\Gamma, x{:}A \vdash \phi$ prop	$=$	$\phi \in \mathcal{P}[\Gamma{\times}A]$
$\Gamma, c{:}TA \vdash [x{\Leftarrow}c]\phi$ prop	$=$	$\mathrm{t}^*_{\Gamma,A}(\Box_{\Gamma\times A}\phi) \in \mathcal{P}[\Gamma{\times}TA]$

When $\mathcal{P}$ is the indexed meet-semilattice induced by a dominion $\mathcal{M}$ over $\mathcal{C}$ and the strong monad $(T, \mathrm{t}, \eta, \mu)$ is s.t. T is a $\mathcal{M}$-functor, then it is easy to show that $\Box_A[m] = [Tm]$ is a box-modality for T over $\mathcal{P}$.

Remark 5.3 Box-modalities are related to T-modalities of [Pit91], which are families $\Box_{A,B}{:}\mathcal{P}[A{\times}B] \to \mathcal{P}[A{\times}TB]$ of monotonic functions satisfying certain equations. First of all (under mild assumptions) there is a 1-1 correspondence between families of the form $\Box_A$ and those of the form $\Box_{A,B}$, namely: $\Box_{A,B} a = \mathrm{t}^*_{A,B}(\Box_{A\times B} a)$ and $\Box_A a = \langle !_A, \mathrm{id}_A\rangle^*(\Box_{1,A}(\pi_2^* a))$. Modulo this correspondence, Pitts' T-modalities are those families $\Box_A$ of monotonic functions satisfying: $(\Box\eta)$, $(\Box\mu)$ and $(\Box T^*)$, but with $\le$ replaced by $=$. The properties $(\Box\top^*)$ and $(\Box\wedge^*)$ are specific to necessity, while property $(\Box\mathrm{t})$ is not required by Pitts, even for necessity, since it fails in the non-standard model of EL_T for side-effects (see Example 1.3).

Theorem 5.4 *If Th is a theory for $ML_T[\top, \wedge, \Box](\Sigma)$, then the indexed meet semi-lattice $\mathcal{P}{:}\mathcal{C}^{op} \to$* **PoSet**, *defined in Section 5.1.1, is equipped with a strong monad $(T, \mathrm{t}, \eta, \mu)$ over $\mathcal{C}$ (see [Mog91]) and a box-modality $\Box$ for T over $\mathcal{P}$ given by $\Box_\tau([x{:}\tau \vdash \phi \text{ prop}]) = [c{:}T\tau \vdash [x{\Leftarrow}c]\phi \text{ prop}]$.*

Proof $\Box_\tau$ is well-defined and monotonic because of (entail). We have to prove (by derivation) that $\Box$ satisfies the properties in Definition 5.2:

$\Box\top$ it amounts to ($\Box$-$\top$);

$\Box\wedge$ it amounts to ($\Box$-$\wedge$);

$\Box\eta$ it amounts to ($\Box$-lift), since $\eta_\tau = [x{:}\tau \vdash [x]{:}T\tau]$;

$\Box\mu$ $\Box_{TA}(\Box_A a) \leq \mu^*(\Box_A a)$ is an instance of ($\Box$-let), since
$\mu_\tau = [c\colon T^2\tau \vdash \text{let } x \Leftarrow c \text{ in } x\colon T\tau]$;

$\Box T$ it amounts to ($\Box$-T), since
$T([x\colon \tau_1 \vdash e\colon \tau_2]) = [c\colon T\tau_1 \vdash \text{let } x \Leftarrow c \text{ in } [e]\colon T\tau_2]$

$\Box$t it amounts to prove the assertion

$$x\colon A, c\colon TB \vdash \phi \wedge [y \Leftarrow c]\psi \Longrightarrow [\langle x, y\rangle \Leftarrow (\text{let } y \Leftarrow c \text{ in } [\langle x, y\rangle])](\phi \wedge \psi)$$

for every $x\colon A \vdash \phi$ prop and $y\colon B \vdash \psi$ prop.

$x\colon A, c\colon TB \vdash [\langle x, y\rangle \Leftarrow (\text{let } y \Leftarrow c \text{ in } [\langle x, y\rangle])](\phi \wedge \psi) \Longleftrightarrow$ by axiom ($\Box$-t)
$x\colon A, c\colon TB \vdash [y \Leftarrow c](\phi \wedge \psi) \Longleftrightarrow$ by axiom ($\Box$-$\wedge$)
$x\colon A, c\colon TB \vdash ([y \Leftarrow c]\phi) \wedge ([y \Leftarrow c]\psi)$ prop.

The assertion follows immediately from $x\colon A, c\colon TB \vdash \phi$ prop$[y \Leftarrow c]\phi$, which is an instance of ($\Box$-D1) derived in Lemma 2.7.

∎

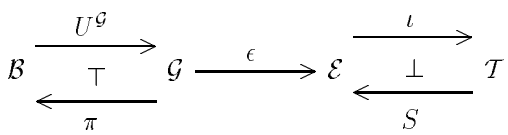
Using the additional structure defined in the previous theorem, it is easy to construct generic models for theories involving also necessity.

5.2 From hyperdoctrine to topos-theoretic models

We introduce three constructions for transforming hyperdoctrine models into topos-theoretic models with the same theory (over some suitable language). The first and second are based on the Grothendieck construction, while the third is studied in [Pit81]. In general, these constructions take a category $\mathcal{B}$ and a $\mathcal{B}$-indexed meet semi-lattice $\mathcal{P}$ (but further properties may be needed), and produce

- a category $\mathcal{C}$ and a *structure preserving* functor $U^{\mathcal{C}}\colon \mathcal{B} \to \mathcal{C}$,
- a dominion $\mathcal{M}^{\mathcal{C}}$ over $\mathcal{C}$ and a $\mathcal{B}$-indexed isomorphism $I^{\mathcal{C}}\colon \mathcal{P} \to U^{\mathcal{C}}; \mathcal{M}^{\mathcal{C}}$.

They differ mainly in the requirements on $\mathcal{B}$ and $\mathcal{P}$ needed for performing the construction, and how additional properties/structure on $\mathcal{B}$ and $\mathcal{P}$ are reflected on $\mathcal{C}$, $\mathcal{M}^{\mathcal{C}}$, $U^{\mathcal{C}}$ and $I^{\mathcal{C}}$.
We investigate which additional properties/structures on $\mathcal{B}$ and $\mathcal{P}$ induce similar properties/structures on $\mathcal{C}$ and $\mathcal{M}^{\mathcal{C}}$ and ensure good preservation properties of $U^{\mathcal{C}}$ and $I^{\mathcal{C}}$. In particular, we consider those properties and additional structures for interpreting the typed predicate logics introduced in Section 2. More precisely, let $(\mathcal{G}, \mathcal{M}^{\mathcal{G}})$, $(\mathcal{E}, \mathcal{M}^{\mathcal{E}})$ and $(\mathcal{T}, \mathcal{M}^{\mathcal{T}})$ be the categories with dominions produced by the three constructions (when defined), then they are related by various functors and enjoy the following properties



- $\mathcal{G}$ has finite products, $U^{\mathcal{G}}$ is full and faithful, π is faithful and left exact (i.e. it preserves finite products), $\mathcal{P}$ is isomorphic to $U^{\mathcal{G}}; \mathcal{M}^{\mathcal{G}}$;
- if $\mathcal{P}$ has equality, then $\mathcal{E}$ has finite limits, ϵ is full, bijective on objects, and preserves products, $\mathcal{M}^{\mathcal{E}}$ has equalities, $\mathcal{M}^{\mathcal{G}}$ is isomorphic to $\epsilon; \mathcal{M}^{\mathcal{E}}$;
- if $\mathcal{P}$ is a tripos satisfying (Comp-P), then $\mathcal{T}$ is a topos, S is full and faithful, ι is faithful and left exact, $\mathcal{M}^{\mathcal{T}}$ is equivalent to the dominion of all monos in $\mathcal{T}$, $\mathcal{M}^{\mathcal{E}}$ is isomorphic to $\iota; \mathcal{M}^{\mathcal{T}}$, $\mathcal{M}^{\mathcal{T}}$ is isomorphic to $S; \mathcal{M}^{\mathcal{E}}$.

Remark 5.5 The key properties of indexed isomorphisms is that they commute with the interpretation of logical constants defined in terms of universal properties. In general, this is does not imply that they commute with the interpretation of formulas of a given predicate logic, unless the functors between base categories preserve the relevant structure for that predicate logic.

5.2.1 The Grothendieck construction

In general the Grothendieck construction transforms indexed categories into categories, but we will apply it only to indexed meet semi-lattices. Throughout this section we fix a category $\mathcal{B}$ with finite products, a $\mathcal{B}$-indexed meet semi-lattice $\mathcal{P}$, a strong monad T over $\mathcal{B}$ and a box-modality $\Box$ for T over $\mathcal{P}$.

Definition 5.6 *Given the assumptions above we can define:*

- *a category $\mathcal{G}$ with finite products s.t.*
  - *an object is a pair $\langle A, a\rangle$ s.t. $A \in \mathcal{B}$ and $a \in \mathcal{P}[A]$,*
  - *a morphism from $\langle A, a\rangle$ to $\langle B, b\rangle$ is an $f : A \to B$ in $\mathcal{B}$ s.t. $a \leq f^* b$ in $\mathcal{P}[A]$,*
  - *composition is induced by composition in $\mathcal{B}$;*
- *a functor $\pi : \mathcal{G} \to \mathcal{B}$ s.t. $\pi(\langle A, a\rangle) = A$ and $\pi(f : \langle A, a\rangle \to \langle B, b\rangle) = f : A \to B$;*
- *a functor $U^{\mathcal{G}} : \mathcal{B} \to \mathcal{G}$ s.t. $U^{\mathcal{G}} A = \langle A, \top\rangle$ and $U^{\mathcal{G}}(f : A \to B) = f : \langle A, \top\rangle \to \langle B, \top\rangle$;*
- *a dominion $\mathcal{M}^{\mathcal{G}}$ over $\mathcal{G}$ whose elements are $\mathrm{id}_A : \langle A, a'\rangle \to \langle A, a\rangle$, where $A \in \mathcal{B}$ and $a' \leq a$ in $\mathcal{P}[A]$;*
- *a $\mathcal{B}$-indexed isomorphism $I^{\mathcal{G}} : \mathcal{P} \to U^{\mathcal{G}}; \mathcal{M}^{\mathcal{G}}$ s.t. $I^{\mathcal{G}}[A]a = [\mathrm{id}_A : \langle A, a\rangle \to \langle A, \top\rangle] \in \mathcal{M}^{\mathcal{G}}[U^{\mathcal{G}} A]$;*
- *a strong monad $T^{\mathcal{G}}$ over $\mathcal{G}$ s.t.*
  - *$T^{\mathcal{G}}(\langle A, a\rangle) = \langle TA, \Box_A a\rangle$ and $T^{\mathcal{G}}(f : \langle A, a\rangle \to \langle B, b\rangle) = Tf$*
  - *$\mathrm{t}^{T^{\mathcal{G}}}_{\langle A,a\rangle,\langle B,b\rangle} = \mathrm{t}^T_{A,B}$*
  - *$\eta^{T^{\mathcal{G}}}_{\langle A,a\rangle} = \eta^T_A$*
  - *$\mu^{T^{\mathcal{G}}}_{\langle A,a\rangle} = \mu^T_A$*

The following theorem establishes key properties of the previous construction.

Theorem 5.7 *The construction in 5.6 satisfies the following properties:*

1. *$\pi \dashv U^{\mathcal{G}}$;*
2. *π is faithful, and it preserves and weakly creates finite limits;*
3. *$\mathcal{G}$ has finite products;*
4. *$U^{\mathcal{G}}$ is full and faithful, and it preserves exponentials;*
5. *$\mathcal{M}^{\mathcal{G}}$ is a dominion over $\mathcal{G}$;*
6. *$I^{\mathcal{G}}$ is a $\mathcal{B}$-indexed isomorphism;*
7. *$T^{\mathcal{G}}$ is a strong monad over $\mathcal{G}$;*
8. *$U^{\mathcal{G}}$ commutes with computational types, i.e. $(U^{\mathcal{G}}, \mathrm{id})$ is a strong monad morphism from T to $T^{\mathcal{G}}$;*
9. *$T^{\mathcal{G}}$ is an $\mathcal{M}^{\mathcal{G}}$-functor;*
10. *$I^{\mathcal{G}}$ commutes with necessity, i.e. $I^{\mathcal{G}}[TA](\Box_A a) = [T^{\mathcal{G}} m]$ when $[m] = I^{\mathcal{G}}[A]a$.*

Proof see Appendix ∎

The main consequence of Theorem 5.7 (and completeness w.r.t. the hyperdoctrine semantics) is completeness of $ML_T[\top,\wedge,\Box](\Sigma)$ w.r.t. the topos-theoretic semantics. Using Theorem 5.9 (see below) one can easily extend this completeness result also to $ML_T[\top,\wedge,\supset,\Box](\Sigma)$, $ML_T[\top,\wedge,\supset,\forall,\Box](\Sigma)$ and $ML_{T\Rightarrow}[\top,\wedge,\supset,\forall,\Box](\Sigma)$.

Lemma 5.8 *If $\mathcal{D}$ is a class of display maps over $\mathcal{B}$ and $\mathcal{P}$ is closed w.r.t. universal quantification along $d \in \mathcal{D}$, then we can define a class of display maps $\mathcal{E} = \{d{:}\langle A', d^*a\rangle \to \langle A, a\rangle | d{:}A' \to A \in \mathcal{D}, a \in \mathcal{P}[A]\}$ over $\mathcal{G}$ s.t. the $\mathcal{G}$-indexed meet semi-lattice $\mathcal{M}^{\mathcal{G}}$ is closed w.r.t. universal quantification along $\mathcal{E}$.*

Proof see Appendix ∎

Theorem 5.9

1. *If $\mathcal{P}$ is closed w.r.t. implication, then the indexed meet semi-lattice $\mathcal{M}^{\mathcal{G}}$ is closed w.r.t. universal quantification along $\mathcal{M}^{\mathcal{G}}$.*
2. *If $\mathcal{P}$ is closed w.r.t. universal quantification over $A \in \mathcal{B}$, then $\mathcal{M}^{\mathcal{G}}$ is closed w.r.t. universal quantification over $U^{\mathcal{G}}A \in \mathcal{G}$.*
3. *If $\mathcal{P}$ is closed w.r.t. implication and universal quantification along projections, then $\mathcal{M}^{\mathcal{G}}$ is closed w.r.t. universal quantification along projections.*
4. *If $\mathcal{B}$ has exponentials and $\mathcal{P}$ is closed w.r.t. implication and universal quantification along projections, then $\mathcal{G}$ has exponentials.*

Proof see Appendix ∎

5.2.2 The modified Grothendieck construction

The construction defined in Section 5.2.1 does not produce dominions with equalities, when applied to hyperdoctrines with equalities. However, this problem can be overcome by quotienting w.r.t. a suitable congruence. Throughout this section we fix a category $\mathcal{B}$ with finite products, a $\mathcal{B}$-indexed meet semi-lattice $\mathcal{P}$ with equalities, a strong monad T over $\mathcal{B}$ and a box-modality $\Box$ for T over $\mathcal{P}$.

Definition 5.10 *Given the assumptions above and $\Box$ satisfying ($\Box$-=) we can define a congruence $\equiv$ on the morphisms of $\mathcal{G}$ given by $f \equiv g$ iff $x{:}A \vdash a(x) \Longrightarrow fx =_B gx$, for every $f, g{:}\langle A, a\rangle \to \langle B, b\rangle$. Moreover, we can define:*

- *a category $\mathcal{E}$ given by the quotient $\mathcal{G}/\equiv$;*
- *a functor $\epsilon{:}\mathcal{G} \to \mathcal{E}$ given by the projection mapping $f{:}\langle A, a\rangle \to \langle B, b\rangle$ to its equivalence class modulo $\equiv$;*
- *a dominion $\mathcal{M}^{\mathcal{E}}$ over $\mathcal{E}$ given by the image of $\mathcal{M}^{\mathcal{G}}$ along ϵ;*
- *a $\mathcal{G}$-indexed isomorphism $J{:}\mathcal{M}^{\mathcal{G}} \to \epsilon;\mathcal{M}^{\mathcal{E}}$ given by $J[\langle A, a\rangle]([m]) = [\epsilon m]$;*
- *a strong monad $T^{\mathcal{E}}$ over $\mathcal{E}$ given by the quotient $T^{\mathcal{G}}/\equiv$, i.e.*
  - $T^{\mathcal{E}}(\langle A, a\rangle) = T^{\mathcal{G}}(\langle A, a\rangle)$ *and* $T^{\mathcal{E}}(\epsilon f) = \epsilon(Tf)$
  - $\mathrm{t}^{T^{\mathcal{E}}}_{\langle A,a\rangle,\langle B,b\rangle} = \epsilon(\mathrm{t}^{T^{\mathcal{G}}}_{\langle A,a\rangle,\langle B,b\rangle})$
  - $\eta^{T^{\mathcal{E}}}_{\langle A,a\rangle} = \epsilon(\eta^{T^{\mathcal{G}}}_{\langle A,a\rangle})$
  - $\mu^{T^{\mathcal{E}}}_{\langle A,a\rangle} = \epsilon(\mu^{T^{\mathcal{G}}}_{\langle A,a\rangle})$

Proof We prove that $\equiv$ is a congruence (i.e. an equivalence which is respected by composition) using the internal logic of $\mathcal{P}$.
The relation $\equiv$ is an equivalence, because $=_B$ satisfies the axioms for an equivalence. To prove that composition respects $\equiv$, it is enough to derive $x{:}A \vdash a(x) \Longrightarrow f'(fx) =_C g'(gx)$ from (1) $x{:}A \vdash a(x) \Longrightarrow fx =_B gx$, (2) $x{:}A \vdash a(x) \Longrightarrow b(fx)$ and (3) $y{:}B \vdash b(y) \Longrightarrow f'y =_C g'y$.

- $y, y' : B \vdash b(y), y =_B y' \Longrightarrow f'y =_C g'y'$ by (3) and (=)
- $x : A \vdash b(fx), fx =_B gx \Longrightarrow f'(fx) =_C g'(gx)$ by (subst)
- $x : A \vdash a(x) \Longrightarrow f'(fx) =_C g'(gx)$ by (1), (2) and (cut).

∎

The following theorem establishes key properties of the previous construction.

Theorem 5.11 *The construction in 5.10 satisfies the following properties:*

1. *ϵ is full and bijective on objects, and it preserves finite products and pullbacks of monos in $\mathcal{M}^{\mathcal{G}}$;*
2. *$\mathcal{E}$ has finite limits;*
3. *$\mathcal{M}^{\mathcal{E}}$ is a dominion over $\mathcal{E}$, and it has equalities;*
4. *J is a $\mathcal{G}$-indexed isomorphism;*
5. *$T^{\mathcal{E}}$ is a strong monad over $\mathcal{E}$;*
6. *ϵ commutes with computational types, i.e. (ϵ, id) is a strong monad morphism from $T^{\mathcal{G}}$ to $T^{\mathcal{E}}$;*
7. *$T^{\mathcal{E}}$ is an $\mathcal{M}^{\mathcal{E}}$-functor;*
8. *J commutes with necessity, i.e. $[\epsilon(T^{\mathcal{G}} m)]) = [T^{\mathcal{E}}(\epsilon m)]$ when $m \in \mathcal{M}^{\mathcal{G}}$.*

Proof see Appendix ∎

From the previous theorem it is easy to derive the key properties of the construction relating it directly to $\mathcal{B}$, $\mathcal{P}$, T and $\Box$.

Corollary 5.12 *If $U^{\mathcal{E}} = U^{\mathcal{G}}; \epsilon : \mathcal{B} \to \mathcal{E}$ and $I^{\mathcal{E}} : \mathcal{P} \to U^{\mathcal{E}}; \mathcal{M}^{\mathcal{E}}$ is the $\mathcal{B}$-indexed isomorphism s.t. $I^{\mathcal{E}}[A]a = J[U^{\mathcal{G}} A](I^{\mathcal{G}}[A]a)$, then*

1. *$U^{\mathcal{E}}$ is full and preserves finite products;*
2. *$I^{\mathcal{E}}$ is a $\mathcal{B}$-indexed isomorphism;*
3. *$U^{\mathcal{E}}$ commutes with computational types, i.e. $(U^{\mathcal{E}}, \mathrm{id})$ is a strong monad morphism from T to $T^{\mathcal{E}}$;*
4. *$I^{\mathcal{E}}$ commutes with necessity, i.e. $I^{\mathcal{E}}[TA](\Box_A a) = [T^{\mathcal{E}} m]$ when $[m] = I^{\mathcal{E}}[A]a$.*

Proof

1. $U^{\mathcal{E}}$ is full and preserve finite products, since $U^{\mathcal{G}}$ and ϵ do.
2. $I^{\mathcal{E}}$ is an indexed isomorphism, since $I^{\mathcal{G}}$ and J are.
3. $(U^{\mathcal{E}}, \mathrm{id})$ is a strong monad morphism, because $(U^{\mathcal{G}}, \mathrm{id})$ and (ϵ, id) are.
4. $I^{\mathcal{E}}$ commutes with necessity, since $I^{\mathcal{G}}$ and J do.

∎

Theorem 5.13

1. *If $\mathcal{P}$ is closed w.r.t. implication, then the $\mathcal{E}$-indexed meet semi-lattice $\mathcal{M}^{\mathcal{E}}$ is closed w.r.t. universal quantification along $\mathcal{M}^{\mathcal{E}}$ (or equivalently implication).*
2. *If $\mathcal{M}^{\mathcal{G}}$ is closed w.r.t. universal quantification over $\langle A, a \rangle$, then so is the $\mathcal{E}$-indexed meet semi-lattice $\mathcal{M}^{\mathcal{E}}$.*
3. *If $\mathcal{P}$ is closed w.r.t. implication and universal quantification along projections, then the $\mathcal{E}$-indexed meet semi-lattice $\mathcal{M}^{\mathcal{E}}$ is closed w.r.t. universal quantification along projections.*
4. *If $\mathcal{B}$ has exponentials and $\mathcal{P}$ satisfies (=-λ), then ϵ preserves exponentials of the form $\langle B, b \rangle^{(U^{\mathcal{G}} A)}$.*

Proof see Appendix ∎

5.2.3 The Tripos construction

Throughout this section we fix a $\mathcal{B}$-tripos $\mathcal{P}$, a strong monad T over $\mathcal{P}$ and a box-modality $\Box$ for T over $\mathcal{P}$. We will freely use the *internal logic* of a tripos to describe constructions and prove their properties.

Definition 5.14 ([Pit81]) *Given a $\mathcal{B}$-tripos $\mathcal{P}$ we can define:*

- *a category $\mathcal{T}$ s.t.*
  - *an object is a pair $\langle A, =_a \rangle$ s.t. $A \in \mathcal{B}$ and $=_a \in \mathcal{P}[A \times A]$ is a* **partial equivalence relation** *(per) on A, i.e.*
    () $x, y : A \vdash x =_a y \Longrightarrow y =_a x$
    () $x, y, z : A \vdash x =_a y, y =_a z \Longrightarrow x =_a z$
  - *a morphism from $\langle A, =_a \rangle$ to $\langle B, =_b \rangle$ is a* **functional relation** *$F \in \mathcal{P}[A \times B]$, i.e.*
    (ext) $x_1, x_2 : A, y_1, y_2 : B \vdash F(x_1, y_1), x_1 =_a x_2, y_1 =_b y_2 \Longrightarrow F(x_2, y_2)$
    (strict) $x : A, y : B \vdash F(x, y) \Longrightarrow E_a(x) \wedge E_b(y)$
    (SV) $x : A, y_1, y_2 : B \vdash F(x, y_1), F(x, y_2) \Longrightarrow y_1 =_b y_2$
    (total) $x : A \vdash E_a(x) \Longrightarrow \exists y : B . F(x, y)$
    where $E_a(x) \in \mathcal{P}[A]$ stands for $x =_a x$
  - *composition of $F : \langle A, =_a \rangle \to \langle B, =_b \rangle$ with $G : \langle B, =_b \rangle \to \langle C, =_c \rangle$ is given by* **relational composition**, *i.e. $x : A, z : C \vdash (F ; G)(x, z) \stackrel{\Delta}{\equiv} \exists y : B . F(x, y) \wedge G(y, z)$;*
- *a functor $\iota : \mathcal{E} \to \mathcal{T}$ s.t. $\iota(\langle A, a \rangle) = \langle A, \simeq_a \rangle$ and $\iota(\epsilon(f) : \langle A, a \rangle \to \langle B, b \rangle) = F : \langle A, \simeq_a \rangle \to \langle B, \simeq_b \rangle$*
  where $\simeq_a \in \mathcal{P}[A \times A]$ is Leibniz' equality over $A \in \mathcal{B}$ restricted to $a \in \mathcal{P}[A]$, and $x : A, y : B \vdash F(x, y) \stackrel{\Delta}{\equiv} a(x) \wedge y \simeq_b fx$;
- *a dominion $\mathcal{M}^{\mathcal{T}}$ whose elements are $R : \langle A, R \rangle \to \langle A, =_a \rangle$, where $R \in \mathcal{P}[A \times A]$ is a per s.t. $=_a ; R ; =_a \leq R \leq =_a$;*
- *a $\mathcal{E}$-indexed isomorphism $K : \mathcal{M}^{\mathcal{E}} \to \iota ; \mathcal{M}^{\mathcal{T}}$ given by $K[\langle A, a \rangle]([m]) = [\iota m]$.*

Theorem 5.15 ([Pit81]) *The construction in 5.14 satisfies the following properties:*

1. *$\mathcal{T}$ is a topos;*
2. *ι is faithful and preserves finite limits;*
3. *$\mathcal{M}^{\mathcal{T}}$ is a dominion over $\mathcal{T}$ s.t. every subobject in $\mathcal{T}$ is represented by a unique mono in $\mathcal{M}^{\mathcal{T}}$;*
4. *K is a $\mathcal{E}$-indexed isomorphism.*

Proof The proof that $\mathcal{T}$ is a topos and that every subobject is represented by a unique mono in $\mathcal{M}$ can be found in [Pit81]. It is immediate to verify that ι is faithful. While to prove that ι preserves finite limits and K is an indexed isomorphism, one has to mimic the proof (given in [Pit81]) that $\Delta_{\mathcal{P}} : \mathcal{B} \to \mathcal{T}$ preserves finite limits and $\mathcal{P}[A]$ and $\mathcal{T}[\Delta_{\mathcal{P}} A]$ are naturally isomorphic. ∎

Corollary 5.16 ([Pit81]) *Under the assumptions of Definition 5.14, if $U^{\mathcal{T}} = U^{\mathcal{E}} ; \iota : \mathcal{B} \to \mathcal{T}$ and $I^{\mathcal{T}} : \mathcal{P} \to U^{\mathcal{T}} ; \mathcal{M}^{\mathcal{T}}$ is the $\mathcal{B}$-indexed isomorphism s.t. $I^{\mathcal{T}}[A]a = K[U^{\mathcal{E}} A](I^{\mathcal{E}}[A]a)$, then*

1. *$U^{\mathcal{T}}$ preserves finite limits;*
2. *$I^{\mathcal{T}}$ is a $\mathcal{B}$-indexed isomorphism.*

Theorem 5.17 *If $\mathcal{B}$ has exponentials and $\mathcal{P}$ is a $\mathcal{B}$-tripos satisfying (Comp-$\Rightarrow$), then ι preserves exponentials of the form $\langle B, b \rangle^{(U^{\mathcal{E}} A)}$.*

Proof For simplicity we assume that $b = \top_B$, and prove bijectivity of the canonical morphism $f{:}B^A, R{:}P(A{\times}B) \vdash F(f,R) \stackrel{\Delta}{\equiv} \forall x{:}A, y{:}B.R(x,y) \leftrightarrow y \simeq_B fx$ from $\langle B^A, \simeq_{A \Rightarrow B}\rangle$ (the image of the exponential in $\mathcal{E}$) to the exponential $\langle P(A{\times}B), \simeq_{FR}\rangle$ in $\mathcal{T}$ (of the images), where $R{:}P(A{\times}B) \vdash FR(R) \stackrel{\Delta}{\equiv} \forall x{:}A.\exists! y{:}B.R(x,y)$.

- $FR(R) \Longrightarrow \exists! f{:}B^A.\forall x{:}A.R(x,fx)$ by (Comp-$\Rightarrow$)
- $FR(R), \forall x{:}A.R(x,fx) \Longrightarrow R(x,y) \leftrightarrow y \simeq_B fx$ by derivation in HML
- therefore $R{:}P(A{\times}B) \vdash FR(R) \Longrightarrow \exists! f{:}B^A.F(f,R)$, which is bijectivity of F.

∎

Definition 5.18 *Given a $\mathcal{B}$-tripos $\mathcal{P}$ satisfying (Comp-P), we can define:*

- *a functor $S{:}\mathcal{T} \to \mathcal{E}$ s.t. $S(\langle A, =_a\rangle) = \langle \Omega^A, S_a\rangle$ and $S(F{:}\langle A,=_a\rangle \to \langle B,=_b\rangle) = (\epsilon(f){:}\langle \Omega^A, S_a\rangle \to \langle \Omega^B, S_b\rangle)$*

  where $X{:}PA \vdash S_a(X) \stackrel{\Delta}{\equiv} \exists x{:}A.E_a(x) \wedge (\forall x'{:}A.x' \in_A X \leftrightarrow x' =_a x)$, and $f{:}PA \to PB$ s.t. $X{:}PA, y{:}B \vdash S_a(X) \Longrightarrow y \in_B fX \leftrightarrow (\exists x{:}A.x \in_A X \wedge F(x,y))$

- *a $\mathcal{T}$-indexed isomorphism $L{:}\mathcal{M}^{\mathcal{T}} \to S;\mathcal{M}^{\mathcal{E}}$ given by $L[\langle A,=_a\rangle]([m]) = [Sm]$;*

Proof To prove that S is well-defined we will use the following consequences of (Comp-P)

(ext-P) $X, Y{:}P\tau \vdash (X =_e Y) \Longleftrightarrow X \simeq_{P\tau} Y$, where $X =_e Y \stackrel{\Delta}{\equiv} (\forall x{:}\tau.x \in_\tau X \leftrightarrow x \in_\tau Y)$

S is well-defined, because $X{:}PA \vdash f(X) \stackrel{\Delta}{\equiv} \{y{:}B | S_a(X) \wedge (\exists x{:}A.x \in_A X \wedge F(x,y))\}{:}PB$ is a witness for $S(F)$, i.e. $X{:}PA \vdash S_a(X) \Longrightarrow S_b(fX)$ is derivable (without using (Comp-P)), and if f' is another witness for $S(F)$, then $X{:}PA \vdash S_a(X) \Longrightarrow fX \simeq_{PB} f'X$ is derivable:

- $X{:}PA, y{:}B \vdash S_a(X) \Longrightarrow y \in_B fX \leftrightarrow y \in_B f'X$ by assumption on f and f',
- $X{:}PA, y{:}B \vdash S_a(X) \Longrightarrow fX =_e f'X$ by formal derivation,
- $X{:}PA, y{:}B \vdash S_a(X) \Longrightarrow fX \simeq_{PB} f'X$ by (ext-P).

∎

Theorem 5.19 *The construction in 5.18 satisfies the following properties:*

1. *$\iota \dashv S$;*
2. *S is full and faithful;*
3. *L is a $\mathcal{T}$-indexed isomorphism;*
4. *$S(z^{\iota(y)})$ is an exponential of $S(z)$ to y in $\mathcal{E}$ (for every $y \in \mathcal{E}$ and $z \in \mathcal{T}$), and ι preserves such exponentials;*
5. *the functor S preserves exponentials;*
6. *$\iota(\langle \Omega, \top\rangle)$ is a subobject classifier of $\mathcal{T}$, and $\eta_{\langle \Omega,\top\rangle}{:}\langle \Omega,\top\rangle \to S(\iota(\langle \Omega,\top\rangle))$ is an isomorphism.*

Proof see Appendix ∎

Remark 5.20 When (Comp-P) holds, we don' t need to use pers to make a topos. In fact, $\mathcal{T}$ becomes equivalent to the topos constructed in [LS86] to prove completeness of Type Theory.

Corollary 5.21 *Under the assumptions of Definition 5.18 $U^{\mathcal{T}}$ preserves powerobjects, i.e. $U^{\mathcal{T}}(\Omega^A)$ is isomorphic to $P(U^{\mathcal{T}}(A))$.*

Proof First one shows that $U^{\mathcal{E}}$ of Theorem 5.13 preserves exponentials of the form Ω^A, in fact (=-λ) for $A \Rightarrow \Omega$ is just a reformulation of (ext-P). Then, one has to apply Lemma 5.19. ∎

Definition 5.22 *Given a $\mathcal{B}$-tripos $\mathcal{P}$ satisfying (Comp-P), a strong monad T over $\mathcal{P}$ and a box-modality $\Box$ for T over $\mathcal{P}$ satisfying ($\Box$-=) and (Comp-T), we can define:*

- *an inverse $\sigma : \iota ; S ; T^{\mathcal{E}} ; \iota \to T^{\mathcal{E}} ; \iota$ to the natural transformation $(T^{\mathcal{E}} ; \iota)(\eta)$ s.t.*

  $C : T(PA), c : TA \vdash \sigma_{\langle A,a \rangle}(C, c) \stackrel{\Delta}{\equiv} [x \Leftarrow c]a(x) \wedge C \simeq_{T(PA)} (\mathrm{let}\ x \Leftarrow c\ \mathrm{in}\ [\{x\}])$

- *a strong monad $T^{\mathcal{T}}$ over $\mathcal{T}$ s.t.*

  – $T^{\mathcal{T}} = S ; T^{\mathcal{E}} ; \iota$

  – $\mathrm{t}^{T^{\mathcal{T}}}_{x,y} = x \times (S ; T^{\mathcal{E}} ; \iota) y \xrightarrow[\sim]{\epsilon_x^{-1} \times \mathrm{id}} \iota(Sx) \times \iota((S ; T^{\mathcal{E}}) y)$

  $\cong \iota((Sx) \times T^{\mathcal{E}}(Sy)) \xrightarrow{\iota(\mathrm{t}^{T^{\mathcal{E}}}_{Sx,Sy})} (T^{\mathcal{E}} ; \iota)(S(x) \times S(y))$

  $\cong (S ; T^{\mathcal{E}} ; \iota)(x \times y)$

  – $\eta^{T^{\mathcal{T}}}_x = x \xrightarrow[\sim]{\epsilon_x^{-1}} \iota(Sx) \xrightarrow{\iota(\eta^{T^{\mathcal{E}}}_{Sx})} (S ; T^{\mathcal{E}} ; \iota)(x)$

  – $\mu^{T^{\mathcal{T}}} = (S ; T^{\mathcal{E}} ; \iota ; S ; T^{\mathcal{E}} ; \iota)(x) \xrightarrow[\sim]{\sigma_{T^{\mathcal{E}}(Sx)}} (S ; T^{\mathcal{E}} ; T^{\mathcal{E}} ; \iota)(x) \xrightarrow{\iota(\mu^{T^{\mathcal{E}}}_{Sx})} (S ; T^{\mathcal{E}} ; \iota)(x)$

Proof To prove that σ is well-defined and is a natural isomorphism, we use (Comp-T) for showing that $(T^{\mathcal{E}} ; \iota)(\eta_{\langle A,a \rangle})$ is invertible.

- $\eta_{\langle A,a \rangle} : \langle A, a \rangle \to \langle PA, S_a \rangle$, where $X ; PA \vdash S_a(X) \stackrel{\Delta}{\equiv} \exists! x : A.a(x) \wedge x \in_A X$, is given by the equivalence class of (the interpretation of) $x : A \vdash \{x\} : PA$, hence
- $(T^{\mathcal{E}} ; \iota)(\eta_{\langle A,a \rangle})$ is given by (the interpretation of) the predicate
  $c : TA, C : T(PA) \vdash F(c, C) \stackrel{\Delta}{\equiv} ([x \Leftarrow c]a(x)) \wedge C \simeq_{T(PA)} (\mathrm{let}\ x \Leftarrow c\ \mathrm{in}\ [\{x\}])$, and
- bijectivity of $(T^{\mathcal{E}} ; \iota)(\eta_{\langle A,a \rangle})$ in $\mathcal{T}$ amounts to $C : T(PA) \vdash [X \Leftarrow C] S_a(X) \Longrightarrow \exists! c : TA.F(c, C)$, which follows from

  – $[X \Leftarrow C] S_a(X) \Longrightarrow \exists! c : TA.C \simeq_{T(PA)} (\mathrm{let}\ x \Leftarrow c\ \mathrm{in}\ [\{x\}])$
  $[X \Leftarrow C] S_a(X) \Longrightarrow$ by ($\Box$-$\Longrightarrow$)
  $[X \Leftarrow C](\exists! x : A.x \in_A X) \Longrightarrow$ by (Comp-T)
  $\exists! c : TA.C \simeq_{T(PA)} (\mathrm{let}\ x \Leftarrow c\ \mathrm{in}\ [\{x\}])$

  – and $[X \Leftarrow C] S_a(X), C \simeq_{T(PA)} (\mathrm{let}\ x \Leftarrow c\ \mathrm{in}\ [\{x\}]) \Longrightarrow [x \Leftarrow c]a(x)$
  $[X \Leftarrow C] S_a(X), C \simeq_{T(PA)} (\mathrm{let}\ x \Leftarrow c\ \mathrm{in}\ [\{x\}]) \Longrightarrow$ by (=) and (subst)
  $[X \Leftarrow (\mathrm{let}\ x \Leftarrow c\ \mathrm{in}\ [\{x\}])] S_a(X) \Longrightarrow$ by ($\Box$-T*)
  $[y \Leftarrow c] S_a(\{y\}) \Longrightarrow$ by ($\Box$-$\Longrightarrow$) and $a(x), x \in_A \{y\} \Longrightarrow a(y)$
  $[x \Leftarrow c]a(x)$.

The definition of $T^{\mathcal{T}}$ relies only on ι and S preserving finite limits (see Theorem 5.15 and 5.19), and σ and ϵ being natural isomorphisms. ∎

Theorem 5.23 *The construction in 5.22 satisfies the following properties:*

1. *$T^{\mathcal{T}}$ is a strong monad over $\mathcal{T}$;*
2. *ι commutes with computational types up to iso, i.e.*
   (ι, σ) is a strong monad morphism from $T^{\mathcal{E}}$ to $T^{\mathcal{T}}$;

3. $T^{\mathcal{T}}$ is an $\mathcal{M}^{\mathcal{T}}$-functor;

4. K commutes with necessity up to iso, i.e.
$[\iota(T^{\mathcal{E}}m)]) = \sigma^*_{\langle A,a\rangle}[T^{\mathcal{T}}(\iota m)]$ when $[m] \in \mathcal{M}^{\mathcal{E}}[\langle A, a\rangle]$.

Proof see Appendix ∎

Corollary 5.24 *Under the assumptions of Definition 5.22*

1. $U^{\mathcal{T}}$ commutes with computational types up to iso, i.e.
$(U^{\mathcal{T}}, U^{\mathcal{E}}; \sigma)$ is a strong monad morphism from T to $T^{\mathcal{T}}$;

2. $I^{\mathcal{T}}$ commutes with necessity up to iso, i.e.
$I^{\mathcal{T}}[TA](\Box_A a) = (U^{\mathcal{E}}; \sigma)^*_A[T^{\mathcal{T}}(\iota m)]$ when $[m] = I^{\mathcal{T}}[A]a$.

Proof Immediate from Theorem 5.11 and 5.23. ∎

Conclusions

The main contribution of this paper are the formal systems introduced in Section 2, and their soundness and completeness w.r.t. natural classes of models. We have tried to justify the generality and flexibility of EL_T by several examples. However, its usability can only be established by applying it to large scale examples. More specific issues which need to be addressed are:

- to compare the expressiveness of EL_T with that of other program logics. For instance, it is not clear whether one can extend the translation of Dynamic Logic into EL_T (given in Example 1.3) to more complex program logics such as Reynolds' Specification Logic or the one introduced in [HMST92]
- to achieve a better integration of EL_T with SDT (see Remark 2.1). In this respect, non-determinism seems particularly problematic. In fact, the full sub-category of domains is not closed w.r.t. the universal construction of free commutative idempotent monoids (corresponding to Plotkin's powerdomain) relative to a topos for SDT
- to find additional properties of T to ensure the existence of powerful inductive/co-inductive principles (see [Pit92b]).

Acknowledgements

I would like to thanks Andy Pitts, Ian Stark, Paul Taylor for technical discussions, and Pino Rosolini for comments on a previous draft.

References

[CP92] R.L. Crole and A.M. Pitts. New foundations for fixpoint computations: Fix hyperdoctrines and the fix logic. *Information and Computation*, 98, 1992.

[GMW79] M.J.C. Gordon, R. Milner, and C.P. Wadsworth. *Edinburgh LCF: A Mechanized Logic of Computation*, volume 78 of *Lecture Notes in Computer Science*. Springer Verlag, 1979.

[HMST92] F. Honsell, I.A. Mason, S.F. Smith, and C. Talcott. A variable typed logic of effects. In *Proceedings 1992 Annual Conference of the European Association for Computer Science Logic CSL92, San Miniato*, volume to appear of *Lecture Notes in Computer Science*. Springer-Verlag, 1992.

[Hyl91] J.M.E. Hyland. First steps in synthetic domain theory. In A. Carboni, editor, *Conference on Category Theory '90*, volume 1488 of *Lecture Notes in Mathematics*. Springer Verlag, 1991.

[LS86] J. Lambek and P.J. Scott. *Introduction to Higher-Order Categorical Logic*, volume 7 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, 1986.

[Mog89] E. Moggi. An abstract view of programming languages. Technical Report ECS-LFCS-90-113, Edinburgh Univ., Dept. of Comp. Sci., 1989. Lecture Notes for course CS 359, Stanford Univ.

[Mog91] E. Moggi. Notions of computation and monads. *Information and Computation*, 93(1), 1991.

[Ole85] F.J. Oles. Type algebras, functor categories and block structure. In M. Nivat and J.C. Reynolds, editors, *Algebraic Methods in Semantics*, 1985.

[OT93] P.W. O'Hearn and R.D. Tennent. Relational parametricity and local variables. In *20th POPL*. ACM, 1993.

[Pit81] A.M. Pitts. *The Theory of Triposes*. PhD thesis, University of Cambridge, 1981.

[Pit88] A.M. Pitts. Categories and types. Lecture Notes for a course at the SERC Logic for IT Summer School on Constructive Logics and Category Theory, August 1988.

[Pit89] A.M. Pitts. Notes on categorical logic. Lecture Notes for a graduate course in the University of Cambridge Computer Laboratory, Lent Term, 1989.

[Pit91] A.M. Pitts. Evaluation logic. In G. Birtwistle, editor, *IVth Higher Order Workshop, Banff 1990*, volume 283 of *Workshops in Computing*. Springer Verlag, 1991.

[Pit92a] A.M. Pitts. A model of the ν-calculus. manuscript, January 1992.

[Pit92b] A.M. Pitts. Relational properties of recursively defined datatypes. preprint, November 1992.

[RR88] E. Robinson and G. Rosolini. Categories of partial maps. *Information and Computation*, 79(2), 1988.

[See87] R.A.G. Seely. Categorical semantics for higher order polymorphic lambda calculus. *Journal of Symbolic Logic*, 52(2), 1987.

[Tay87] P. Taylor. *Recursive Domains, Indexed Category Theory and Polymorphism*. PhD thesis, University of Cambridge, 1987.

A Appendix

Proof of 2.7.

1. $\Gamma, x{:}\tau \vdash \phi, \top \Longrightarrow \phi$ is true
   $\Gamma, c{:}T\tau \vdash \phi, [x{\Leftarrow}c]\top \Longrightarrow [x{\Leftarrow}c]\phi$ by ($\Box$-$\Longrightarrow$) and $x \notin \mathrm{FV}(\phi)$
   $\Gamma, c{:}T\tau \vdash \phi \Longrightarrow [x{\Leftarrow}c]\phi$ by ($\Box$-$\top$*)

2. $\Gamma, x{:}\tau \vdash \phi_1 \wedge \phi_2 \Longrightarrow \phi_i$ is true
   $\Gamma, c{:}T\tau \vdash [x{\Leftarrow}c](\phi_1 \wedge \phi_2) \Longrightarrow [x{\Leftarrow}c]\phi_i$ by ($\Box$-$\Longrightarrow$)
   from this one can easily derive the axiom

3. $\Gamma, x{:}\tau \vdash \phi_1, (\phi_1 \supset \phi_2) \Longrightarrow \phi_2$ is true
   $\Gamma, c{:}T\tau \vdash \phi_1, [x{\Leftarrow}c](\phi_1 \supset \phi_2) \Longrightarrow [x{\Leftarrow}c]\phi_2$ by ($\Box$-$\Longrightarrow$) and $x \notin \mathrm{FV}(\phi_1)$
   from this one can easily derive the axiom

4. $\Gamma, x{:}\tau_1, y{:}\tau_2 \vdash \forall y{:}\tau_2.\phi \Longrightarrow \phi$ is true
   $\Gamma, c{:}T\tau_1, y{:}\tau_2 \vdash [x{\Leftarrow}c]\forall y{:}\tau_2.\phi \Longrightarrow [x{\Leftarrow}c]\phi$ by ($\Box$-$\Longrightarrow$)
   from this and $y \notin \mathrm{FV}([x{\Leftarrow}c]\forall y{:}\tau_2.\phi)$ one can easily derive the axiom

5. the direction $\Longrightarrow$ is an instance of ($\Box$-T), so we derive the other entailment:
   $[y{\Leftarrow}(\text{let } x{\Leftarrow}c \text{ in } [e])]\phi \Longrightarrow$ by (congr) using $y = [e'/x]e$ and $x \notin \mathrm{FV}(\phi)$
   $[y{\Leftarrow}(\text{let } x{\Leftarrow}c \text{ in } [e])][e'/x]([e/y]\phi) \Longrightarrow$ by ($\Box$-T)
   $[x{\Leftarrow}\text{let } y{\Leftarrow}(\text{let } x{\Leftarrow}c \text{ in } [e]) \text{ in } [e']]([e/y]\phi) \Longrightarrow$ by (congr) using equational rules
   $[x{\Leftarrow}(\text{let } x{\Leftarrow}c \text{ in } [[e/y]e'])]([e/y]\phi) \Longrightarrow$ by (congr) using $x = [e/y]e'$
   $[x{\Leftarrow}(\text{let } x{\Leftarrow}c \text{ in } [x])]([e/y]\phi) \Longrightarrow$ by (congr) using equational rules
   $[x{\Leftarrow}c]([e/y]\phi)$

6. $[x{\Leftarrow}c]([y{\Leftarrow}e]\phi) \Longrightarrow$ by ($\Box$-T)
   $[z{\Leftarrow}(\text{let } x{\Leftarrow}c \text{ in } [e])]([y{\Leftarrow}z]\phi) \Longrightarrow$ by ($\Box$-μ)
   $[y{\Leftarrow}\text{let } z{\Leftarrow}(\text{let } x{\Leftarrow}c \text{ in } [e]) \text{ in } z]\phi \Longrightarrow$ by (congr) using equational rules
   $[y{\Leftarrow}(\text{let } x{\Leftarrow}c \text{ in } e)]\phi$

∎

Proof of 2.8

1. ($\Box$-t*) is ($\Box$-T*) applied to the term $\Gamma, x{:}\tau_1, y{:}\tau_2 \vdash \langle x, y\rangle : \tau_1 \times \tau_2$ and the formula $\Gamma, x{:}\tau_1, z{:}\tau_1 \times \tau_2 \vdash \phi$ prop

2. The derivation of ($\Box$-let*) is obtained from that of ($\Box$-let), given in Lemma 2.7, by replacing $\Longrightarrow$ with $\Longleftrightarrow$ and using the rules ($\Box$-T*) and ($\Box$-μ*).

∎

Proof of 2.9.

1. The proof uses three sub-derivations:

   - $\Gamma, y{:}\tau_2 \vdash \Phi, \phi, y =_{\tau_2} [e'/x]e \Longrightarrow [e'/x]([e/y]\phi)$ by (=), (subst) and $x \notin \mathrm{FV}(\phi)$
     $\Gamma, y{:}\tau_2 \vdash \Phi, \phi \Longrightarrow [e'/x]([e/y]\phi)$
     by (cut) and the assumption $\Gamma, y{:}\tau_2 \vdash \Phi \Longrightarrow y =_{\tau_2} [e'/x]e$
     $\Gamma, c{:}T\tau_1 \vdash \Phi, [y{\Leftarrow}(\text{let } x{\Leftarrow}c \text{ in } [e])]\phi \Longrightarrow [y{\Leftarrow}(\text{let } x{\Leftarrow}c \text{ in } [e])][e'/x]([e/y]\phi)$
     by ($\Box$-$\Longrightarrow$) and (subst)
   - $\Gamma, c{:}T\tau_1 \vdash [y{\Leftarrow}(\text{let } x{\Leftarrow}c \text{ in } [e])][e'/x]([e/y]\phi) \Longrightarrow [x{\Leftarrow}(\text{let } x{\Leftarrow}c \text{ in } [[e/y]e'])]([e/y]\phi)$
     by ($\Box$-T) and (congr) using equational rules

- $\Gamma, x{:}\tau_1 \vdash \Phi \Longrightarrow x =_{\tau_1} [e/y]e'$ by assumption
  $\Gamma, c{:}T\tau_1 \vdash \Phi \Longrightarrow [x{\Leftarrow}c]x =_{\tau_1} [e/y]e'$ by ($\Box$-$\Longrightarrow$) and ($\Box$-$\top^*$)
  $\Gamma, c{:}T\tau_1 \vdash \Phi \Longrightarrow (\text{let } x{\Leftarrow}c \text{ in } [[e/y]e']) =_{T\tau_1} (\text{let } x{\Leftarrow}c \text{ in } [x])$ by ($\Box$-$=$) and (cut)
  $\Gamma, c{:}T\tau_1 \vdash \Phi \Longrightarrow (\text{let } x{\Leftarrow}c \text{ in } [[e/y]e']) =_{T\tau_1} c$ by (congr) and equational rules
  $\Gamma, c{:}T\tau_1 \vdash \Phi, [x{\Leftarrow}(\text{let } x{\Leftarrow}c \text{ in } [[e/y]e'])]([e/y]\phi) \Longrightarrow [x{\Leftarrow}c]([e/y]\phi)$
  by (=), (subst) and (cut)

2. $[x{\Leftarrow}c](e_1 =_{T\tau_2} e_2) \Longrightarrow$ by ($\Box$-$\Longrightarrow$) and $\Gamma, x{:}\tau_1 \vdash e_1 =_{T\tau_2} e_2 \Longrightarrow [e_1] =_{T^2\tau_2} [e_2]$
   $[x{\Leftarrow}c]([e_1] =_{T^2\tau_2} [e_2]) \Longrightarrow$ by ($\Box$-$=$)
   $(\text{let } x{\Leftarrow}c \text{ in } [e_1]) =_{T^2\tau_2} (\text{let } x{\Leftarrow}c \text{ in } [e_2]) \Longrightarrow$ by (=) and (subst)
   $\text{let } y{\Leftarrow}(\text{let } x{\Leftarrow}c \text{ in } [e_1]) \text{ in } y =_{T\tau_2} \text{let } y{\Leftarrow}(\text{let } x{\Leftarrow}c \text{ in } [e_2]) \text{ in } y \Longrightarrow$ by (congr) using equational rules
   $(\text{let } x{\Leftarrow}c \text{ in } e_1) =_{T\tau'} (\text{let } x{\Leftarrow}c \text{ in } e_2)$

3. $\Gamma, x{:}\tau \vdash \Phi, \top \Longrightarrow e_1' =_{T\tau'} e_2'$ is an assumption
   $\Gamma, c{:}T\tau \vdash \Phi \Longrightarrow [x{\Leftarrow}c](e_1' =_{T\tau'} e_2')$ by ($\Box$-$\Longrightarrow$) and ($\Box$-$\top^*$)
   $\Gamma, c{:}T\tau \vdash \Phi \Longrightarrow (\text{let } x{\Leftarrow}c \text{ in } e_1') =_{T\tau'} (\text{let } x{\Leftarrow}c \text{ in } e_2')$ by ($\Box^+$-$=$)
   $\Gamma \vdash \Phi, e_1 =_{T\tau} e_2 \Longrightarrow (\text{let } x{\Leftarrow}e_1 \text{ in } e_1') =_{T\tau'} (\text{let } x{\Leftarrow}e_2 \text{ in } e_2')$ by (=) and (subst)
   $\Gamma \vdash \Phi \Longrightarrow (\text{let } x{\Leftarrow}e_1 \text{ in } e_1') =_{T\tau'} (\text{let } x{\Leftarrow}e_2 \text{ in } e_2')$ by (cut) and
   the assumption $\Gamma \vdash \Phi \Longrightarrow e_1 =_{T\tau} e_2$

∎

Proof of 2.10.

1. Under the assumption $(\forall x{:}\tau.x \in_\tau X \leftrightarrow x \in_\tau Y)$ both X and Y satisfy
   $(\forall x{:}\tau.\phi \leftrightarrow x \in_\tau _)$, where $\phi \stackrel{\Delta}{\equiv} (x \in_\tau X)$.
   Therefore, they must be equal because of (Comp-P).

2. Under the assumption X both $\{x{:}1|X\}$ and $\{x{:}1|\top\}$ satisfy
   $(\forall x{:}\tau.X \leftrightarrow x \in_\tau _)$ because of (congr) and $(P.\beta)$.
   Therefore, they must be equal because of (Comp-P).

   Now consider the following derivation:
   $X{:}\Omega \vdash \{x{:}1|X\} \simeq_{P1} \{x{:}1|\top\} \Longrightarrow (* \in_1 \{x{:}1|X\}) \simeq_\Omega (* \in_1 \{x{:}1|\top\})$
   by (=) and (subst)
   $X{:}\Omega \vdash \{x{:}1|X\} \simeq_{P1} \{x{:}1|\top\} \Longrightarrow X \simeq_\Omega \top$ by (congr) and $(P.\beta)$.

∎

Proof of 2.11. If Φ and $(\forall x{:}\tau_1.e_1 =_{\tau_2} e_2)$, then both $(\lambda x{:}\tau_1.e_1)$ and $(\lambda x{:}\tau_1.e_2)$ satisfy
$(\forall x{:}\tau_1.\phi(x, _(x))$, where $\phi(x,y) \stackrel{\Delta}{\equiv} (y =_{\tau_2} e_1)$, because of $(\Rightarrow.\beta)$ and (congr).
Moreover, any $\phi(x,y) \equiv (y \simeq_{\tau_2} e)$ s.t. $y \notin \mathrm{FV}(e)$ satisfies $\forall x{:}\tau_1.\exists! y{:}\tau_2.\phi(x,y)$.
Therefore, they must be equal because of (Comp-$\Rightarrow$). ∎

Proof of 2.12. We prove a cycle of implications.

1. $[x{\Leftarrow}c]\phi \Longrightarrow$ by (ext-Ω) and ($\Box$-$\Longrightarrow$)
   $[x{\Leftarrow}c](\phi =_\Omega \top) \Longrightarrow$ by ($\Box$-$=$)
   $(\text{let } x{\Leftarrow}c \text{ in } [\phi]) =_{T\Omega} (\text{let } x{\Leftarrow}c \text{ in } [\top])$

2. $(\text{let } x{\Leftarrow}c \text{ in } [\phi]) =_{T\Omega} (\text{let } x{\Leftarrow}c \text{ in } [\top]) \Longrightarrow$
   by (cut), (=) and $\Box(\text{let } x{\Leftarrow}c \text{ in } [\top])$
   $\Box(\text{let } x{\Leftarrow}c \text{ in } [\phi])$

   so we have to derive $\Box(\text{let } x{\Leftarrow}c \text{ in } [\top])$:
   $[x{\Leftarrow}c]\top$ by ($\Box$-$\top^*$)
   $[X{\Leftarrow}(\text{let } x{\Leftarrow}c \text{ in } [\top])]X$ by ($\Box$-T)

3. $\Box(\text{let } x{\Leftarrow}c \text{ in } [\phi]) \Longrightarrow$ by definition of $\Box(c)$
   $[X{\Leftarrow}(\text{let } x{\Leftarrow}c \text{ in } [\phi])]X \Longrightarrow$ by $(\Box\text{-}T^*)$
   $[x{\Leftarrow}c]\phi$.

∎

Proof of 2.13.

1. $\forall X{:}P\tau.([x{\Leftarrow}c]x \in_\tau X) \supset ([v{\Leftarrow}c]v \in_\tau X)$ is true by α-conversion
   $\forall X{:}P\tau.[v{\Leftarrow}c](([x{\Leftarrow}c]x \in_\tau X) \supset v \in_\tau X)$ by $(\Box\text{-}\supset^*)$ and $v \notin \mathrm{FV}([x{\Leftarrow}c]x \in_\tau X)$
   $[v{\Leftarrow}c](\forall X{:}P\tau.([x{\Leftarrow}c]x \in_\tau X) \supset v \in_\tau X)$ by $(\Box\text{-}\forall^*)$

2. we derive the entailment judgements $\Gamma, c{:}T\tau, x{:}\tau \vdash ([x{\Leftarrow}c]\phi), c{\Downarrow}x \Longrightarrow \phi$ and
   $\Gamma, c{:}T\tau \vdash (\forall x{:}\tau.c{\Downarrow}x \supset \phi) \Longrightarrow ([x{\Leftarrow}c]\phi)$.

   $\Gamma, c{:}T\tau, x{:}\tau, X{:}P\tau \vdash ([x{\Leftarrow}c]\phi), c{\Downarrow}x, ([x{\Leftarrow}c]x \in_\tau X) \Longrightarrow x \in_\tau X$
   by (assume), $(\forall)$ and $(\supset)$
   $\Gamma, c{:}T\tau, x{:}\tau \vdash ([x{\Leftarrow}c]\phi), c{\Downarrow}x, ([x{\Leftarrow}c]x \in_\tau \{x{:}\tau|\phi\}) \Longrightarrow x \in_\tau \{x{:}\tau|\phi\}$ by (subst)
   $\Gamma, c{:}T\tau, x{:}\tau \vdash ([x{\Leftarrow}c]\phi), c{\Downarrow}x \Longrightarrow \phi$ by (congr) and $(P.\beta)$

   $\Gamma, c{:}T\tau, x{:}\tau \vdash (\forall x{:}\tau.c{\Downarrow}x \supset \phi), c{\Downarrow}x \Longrightarrow \phi$ by (assume), $(\forall)$ and $(\supset)$
   $\Gamma, c{:}T\tau \vdash (\forall x{:}\tau.c{\Downarrow}x \supset \phi), [x{\Leftarrow}c]c{\Downarrow}x \Longrightarrow ([x{\Leftarrow}c]\phi)$ by $(\Box\text{-}\Longrightarrow)$ and (subst)
   $\Gamma, c{:}T\tau \vdash (\forall x{:}\tau.c{\Downarrow}x \supset \phi) \Longrightarrow ([x{\Leftarrow}c]\phi)$ by (cut) and $[x{\Leftarrow}c]c{\Downarrow}x$

3. we establish a sequence of equivalences

   $(\langle x{\Leftarrow}c\rangle\phi) \Longleftrightarrow$ by definition
   $(\forall w{:}\Omega.([x{\Leftarrow}c](\phi \supset w)) \supset w) \Longleftrightarrow$ by $([x{\Leftarrow}c]\phi) \leftrightarrow (\forall x{:}\tau.c{\Downarrow}x \supset \phi)$
   $(\forall w{:}\Omega.(\forall x{:}\tau.c{\Downarrow}x \supset (\phi \supset w)) \supset w) \Longleftrightarrow$ by $\phi_1 \supset (\phi_2 \supset \phi_3) \leftrightarrow (\phi_1 \wedge \phi_2) \supset \phi_3$
   $(\forall w{:}\Omega.(\forall x{:}\tau.(c{\Downarrow}x \wedge \phi) \supset w) \supset w) \Longleftrightarrow$
   by $(\exists x{:}\tau.\phi) \leftrightarrow (\forall w{:}\Omega.(\forall x{:}\tau.(\phi \supset w)) \supset w)$
   $(\exists x{:}\tau.c{\Downarrow}x \wedge \phi)$

4. we establish a sequence of equivalences

   $(\langle x{\Leftarrow}c\rangle(x =_\tau v)) \Longleftrightarrow$ by $(\langle x{\Leftarrow}c\rangle\phi) \leftrightarrow (\exists x{:}\tau.c{\Downarrow}x \wedge \phi)$
   $(\exists x{:}\tau.c{\Downarrow}x \wedge x =_\tau v) \Longleftrightarrow$ by $(\exists x{:}\tau.\phi \wedge x =_\tau v) \leftrightarrow [v/x]\phi$
   $(c{\Downarrow}v)$

∎

Proof of 4.8. For convenience, we indicate in the same way a well-formed syntactic entity and its interpretation; we may write ϕ for a mono m, when the subobject $[m]$ is the interpretation of the formula ϕ; we may apply the operation $\Box_{X,A}[m] = \mathrm{t}^*_{X,A}[m]$ also on monos, with the requirement that $[\Box_{X,A}m] = \Box_{X,A}[m]$. When proving the soundness of a rule, we will mark in boldface the use of an additional assumption on strong monads.

$(\Box\text{-}\top^*)$ It is enough to show that $[\mathrm{id}_{\Gamma\times T\tau}] = \mathrm{t}^*_{\Gamma,\tau}[T\mathrm{id}_{\Gamma\times\tau}]$.
The assertion follows immediately from $T\mathrm{id}_X = \mathrm{id}_{TX}$ (functoriality of T) and $[\mathrm{id}_X] = f^*[\mathrm{id}_Y]$ (by general properties of pullbacks).

$(\Box\text{-}\Longrightarrow)$ We have to show that $[\Phi\times\mathrm{id}_\tau] \wedge \phi \leq \psi$ implies $[\Phi\times\mathrm{id}_{T\tau}] \wedge \Box_{\Gamma,\tau}\phi \leq \Box_{\Gamma,\tau}\psi$.
Given $m{:}X \to Y$ and $a, b \in \mathcal{M}[Y]$, the assertions $[m] \wedge a \leq b$ and $m^*a \leq m^*b$ are equivalent (by general properties of pullbacks). Therefore, the above implication rewrites to $(\Phi\times\mathrm{id}_\tau)^*\phi \leq (\Phi\times\mathrm{id}_\tau)^*\psi$ implies $(\Phi\times\mathrm{id}_{T\tau})^*(\Box_{\Gamma,\tau}\phi) \leq (\Phi\times\mathrm{id}_{T\tau})^*(\Box_{\Gamma,\tau}\psi)$.
Since T is **strongly mono preserving**, then the conclusion is equivalent to $\Box_{\Gamma,\tau}((\Phi\times\mathrm{id}_\tau)^*\phi) \leq \Box_{\Gamma,\tau}((\Phi\times\mathrm{id}_\tau)^*\psi)$.
Given $a, b \in \mathcal{M}[X,\tau]$, one has $a \leq b$ implies $\Box_{X,\tau}a \leq \Box_{X,\tau}b$ (by general properties of pullbacks). Therefore, by a suitable instantiation of a and b, we get the desired implication.

($\Box$-T) We have to show that $\Box_{\Gamma,\tau_1}[n] \leq [m]$, where $[n] = \langle \pi_1, e\rangle^*\phi$ and $[m] = \langle \pi_1, \mathrm{t}_{\Gamma,\tau_1}; Te\rangle^*(\Box_{\Gamma,\tau_2}\phi)$. Since $[m] = (\langle \pi_1, \mathrm{t}_{\Gamma,\tau_1}; Te\rangle; \mathrm{t}_{\Gamma,\tau_2})^*[T\phi]$, we have only to find some f (necessarily unique) s.t.

$$\begin{array}{ccc} \Gamma\times T\tau_1 & \xrightarrow{\langle \pi_1, \mathrm{t}_{\Gamma,\tau_1}; Te\rangle; \mathrm{t}_{\Gamma,\tau_2}} & T(\Gamma\times\tau_2) \\ \uparrow{\scriptstyle \Box_{\Gamma,\tau_1} n} & & \uparrow{\scriptstyle T\phi} \\ \cdot & \xrightarrow{\quad f \quad} & TX \end{array}$$

Let us consider the following sequence of commuting squares:

$$\begin{array}{ccccc} \Gamma\times T\tau_1 & \xrightarrow{\mathrm{t}_{\Gamma,\tau_1}} & T(\Gamma\times\tau_1) & \xrightarrow{T(\langle \pi_1, e\rangle)} & T(\Gamma\times\tau_2) \\ \uparrow{\scriptstyle \Box_{\Gamma,\tau_1} n} & (1) & \uparrow{\scriptstyle Tn} & (2) & \uparrow{\scriptstyle T\phi} \\ \cdot & \longrightarrow & \cdot & \longrightarrow & TX \end{array}$$

the justifications for commutativity are: (1) is the pullback defining $\Box_{\Gamma,\tau_1} n$, (2) is T applied to the pullback defining n.
Therefore, we need only to prove that $\langle \pi_1, \mathrm{t}_{\Gamma,\tau_1}; Te\rangle; \mathrm{t}_{\Gamma,\tau_2} = \mathrm{t}_{\Gamma,\tau_1}; T(\langle \pi_1, e\rangle)$, which amounts to the following equation in ML_T

$$\begin{array}{rl} u{:}\Gamma, c{:}T\tau_1 \vdash & \text{let } y \Leftarrow (\text{let } x \Leftarrow c \text{ in } [e]) \text{ in } [\langle u, y\rangle] = \\ & \text{let } x \Leftarrow c \text{ in } [\langle u, e\rangle]; T(\Gamma\times\tau_2) \end{array}$$

($\Box$-T*) We have to show that $\Box_{\Gamma,\tau_1}[n] = [m]$, where $[n] = \langle \pi_1, e\rangle^*\phi$ and $[m] = \langle \pi_1, \mathrm{t}_{\Gamma,\tau_1}; Te\rangle^*(\Box_{\Gamma,\tau_2}\phi)$. The assertion follows from the proof of ($\Box$-T) by observing that the commuting squares are pullbacks: (1) is the pullback defining $\Box_{\Gamma,T\tau} n$, (2) is T applied to the pullback defining n (and T is an **$\mathcal{M}$-functor**).

($\Box$-t*) We have to show that $(\mathrm{id}_\Gamma \times \mathrm{t}_{\tau_1,\tau_2})^*(\Box_{\Gamma,\tau_1\times\tau_2}\phi) = \Box_{\Gamma\times\tau_1,\tau_2}\phi$ (we leave implicit the isomorphisms corresponding to associativity of products), or equivalently $((\mathrm{id}_\Gamma \times \mathrm{t}_{\tau_1,\tau_2}); \mathrm{t}_{\Gamma,\tau_1\times\tau_2})^*[T\phi] = \mathrm{t}^*_{\Gamma\times\tau_1,\tau_2}[T\phi]$.
The assertion follows immediately from $(\mathrm{id}_\Gamma \times \mathrm{t}_{\tau_1,\tau_2}); \mathrm{t}_{\Gamma,\tau_1\times\tau_2} = \mathrm{t}_{\Gamma\times\tau_1,\tau_2}$ (see definition 3.1 of strong functor).

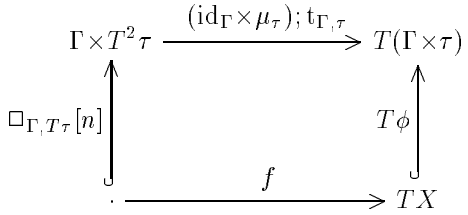
($\Box$-η) We have to show that $\phi \leq (\mathrm{id}_\Gamma \times \eta_\tau)^*(\Box_{\Gamma,\tau}\phi)$, or equivalently $\phi \leq ((\mathrm{id}_\Gamma \times \eta_\tau); \mathrm{t}_{\Gamma,\tau})^*[T\phi]$.
The assertion follows by general properties of pullbacks from $(\mathrm{id}_\Gamma \times \eta_\tau); \mathrm{t}_{\Gamma,\tau} = \eta_{\Gamma\times\tau}$ (η is strong)

and

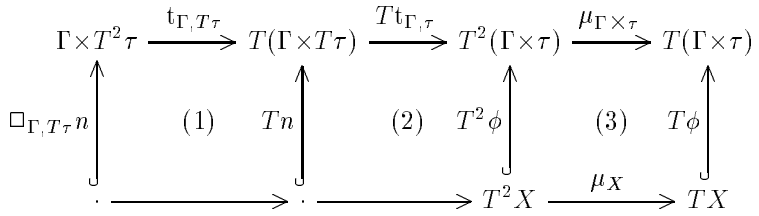
$$\begin{array}{ccc} \Gamma\times\tau & \xrightarrow{\eta_{\Gamma\times\tau}} & T(\Gamma\times\tau) \\ \uparrow{\scriptstyle \phi} & & \uparrow{\scriptstyle T\phi} \\ X & \xrightarrow[\eta_X]{} & TX \end{array}$$

($\Box$-η*) We have to show that $\phi = (\mathrm{id}_\Gamma \times \eta_\tau)^*(\Box_{\Gamma,\tau}\phi)$.
The assertion follows from the proof of ($\Box$-η) by observing that the commuting square is a pullback (η is **$\mathcal{M}$-cartesian**).

($\Box$-μ) We have to show that $\Box_{\Gamma,T\tau}[n] \leq (\mathrm{id}_\Gamma \times \mu_\tau)^*(\Box_{\Gamma,\tau}\phi) = [m]$, where $[n] = \Box_{\Gamma,\tau}\phi$.
Since $[m] = ((\mathrm{id}_\Gamma \times \mu_\tau); \mathrm{t}_{\Gamma,\tau})^*[T\phi]$, we have only to find some f (necessarily unique) s.t.



Let us consider the following sequence of commuting squares:



the justifications for commutativity are: (1) is the pullback defining $\Box_{\Gamma,T\tau}n$, (2) is T applied to the pullback defining n. (3) commutes by naturality of μ.
Therefore, we need only to prove that $\mathrm{t}_{\Gamma,T\tau}; (T\mathrm{t}_{\Gamma,\tau}); \mu_{\Gamma\times\tau} = (\mathrm{id}_\Gamma \times \mu_\tau); \mathrm{t}_{\Gamma,T\tau}$, which amounts to μ being strong.

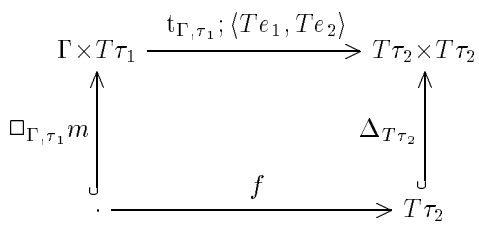
($\Box$-μ*) We have to show that $\Box_{\Gamma,T\tau}[n] = (\mathrm{id}_\Gamma \times \mu_\tau)^*(\Box_{\Gamma,\tau}\phi) = [m]$, where $[n] = \Box_{\Gamma,\tau}\phi$.
The assertion follows from the proof of ($\Box$-μ) by observing that the commuting squares are pullbacks: (1) is the pullback defining $\Box_{\Gamma,T\tau}n$, (2) is T applied to the pullback defining n (and T is an $\mathcal{M}$-**functor**), (3) is a pullback (μ is $\mathcal{M}$-**cartesian**).

($\Box$-$\wedge$*) Let $[m] = \phi_1 \wedge \phi_2$, We have to show that $\mathrm{t}^*_{\Gamma,\tau}[Tm] = (\mathrm{t}^*_{\Gamma,\tau}[T\phi_1]) \wedge (\mathrm{t}^*_{\Gamma,\tau}[T\phi_2])$.
The assertion follows immediately from $[Tm] = [T\phi_1] \wedge [T\phi_2]$ (T is **meet preserving**) and preservation of meets by substitution functors.

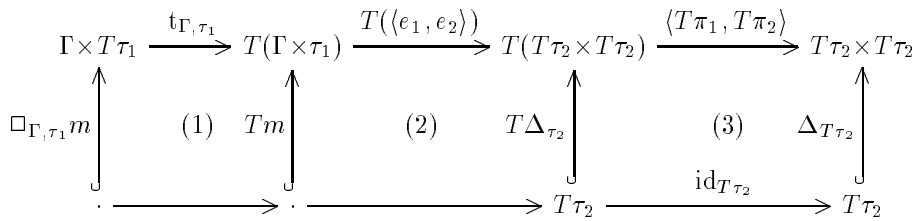
($\Box$-$\supset$*) Let $m\colon X \hookrightarrow \Gamma$ s.t. $[m] = \phi_1 \in \mathcal{M}[\Gamma]$, then we have to show that $\forall_{m\mathrm{id}_{T\tau}}((m\mathrm{id}_{T\tau})^*(\Box_{\Gamma,T\tau}\phi_2)) = \Box_{\Gamma,\tau}(\forall_{m\mathrm{id}_\tau}((m\mathrm{id}_\tau)^*\phi_2))$, because of the way implication is interpreted in terms of universal quantification along $m \in \mathcal{M}$.
The lhs rewrites to $\forall_{m\mathrm{id}_{T\tau}}(\Box_{X,\tau}((m\mathrm{id}_\tau)^*\phi_2))$, because T is **strongly mono preserving**.
Then, the equality follows immediately from **commutativity** of necessity with universal quantification along $m \in \mathcal{M}$.

($\Box$-$\forall$*) The proof is similar to that of ($\Box$-$\supset$*).

($\Box$-=) We have to show that $\Box_{\Gamma,\tau_1}[m] \leq \langle \mathrm{t}_{\Gamma,\tau_1}; Te_1, \mathrm{t}_{\Gamma,\tau_1}; Te_2\rangle^*[\Delta_{T\tau_2}]$, where $[m] = \langle e_1, e_2\rangle^*[\Delta_{\tau_2}]$.
We have only to find some f (necessarily unique) s.t.



Let us consider the following sequence of commuting squares:



the justifications for commutativity are: (1) is the pullback defining $\Box_{\Gamma,\tau_1}m$, (2) is T applied to the pullback defining m, (3) commutes because of $\Delta = \langle \mathrm{id}, \mathrm{id}\rangle$ (and a simple calculation).
Therefore, we need only to prove that $\mathrm{t}_{\Gamma,\tau_1}; \langle Te_1, Te_2\rangle = \mathrm{t}_{\Gamma,\tau_1}; T(\langle e_1, e_2\rangle); \langle T\pi_1, T\pi_2\rangle$, which follows immediately from $\langle Te_1, Te_2\rangle = T(\langle e_1, e_2\rangle); \langle T\pi_1, T\pi_2\rangle$.

(Comp-T) Because of the assumptions on $\mathcal{C}$ and $\mathcal{M}$ we have:

- Leibniz' equality $x \simeq_\tau y$ is interpreted by $[\Delta_\tau] \in \mathcal{M}[\tau\times\tau]$;
- $X\colon P\tau \vdash \exists!x\colon\tau.x \in_\tau X \iff \exists!x\colon\tau.\{x\} \simeq_\tau X$ holds;
- $x\colon\tau \vdash \{x\}\colon P\tau$ is interpreted by a mono $m\colon\tau \rightarrowtail P\tau$;
- $x\colon T\tau \vdash$ let $x \Leftarrow c$ in $[\{x\}]\colon T(P\tau)$ is interpreted by Tm, which is mono (since T is mono preserving);
- if $x\colon\tau_1 \vdash e\colon\tau_2$ is interpreted by a mono $m\colon\tau_1 \hookrightarrow \tau_2$, then $y\colon\tau_2 \vdash \exists!x\colon\tau_1.e \simeq_\tau y$ prop is interpreted by $[m] \in \mathcal{M}[\tau_2]$;

so both sides of the equivalence are interpreted by $[Tm] \in \mathcal{M}[T(P\tau)]$. ∎

Proof of 5.7.

1. The adjunction $\pi \dashv U^{\mathcal{G}}$ follows from $\mathcal{G}(\langle A, a\rangle, \langle B, \top\rangle) = \mathcal{B}(A, B)$.
2. π is clearly faithful. We prove only that π preserves and weakly creates binary products, since this proof can be easily extended to the general case:
   - **preservation** let $\pi_i\colon\langle B, b\rangle \to \langle A_i, a_i\rangle$ be a product diagram in $\mathcal{G}$, we have to show that $\pi_i\colon B \to A_i$ is a product diagram in $\mathcal{B}$. In fact, if $f_i\colon C \to A_i$, then $f_i\colon\langle C, f_1^*a_1 \wedge f_2^*a_2\rangle \to \langle A_i, a_i\rangle$, therefore exists (unique) $f\colon\langle C, f_1^*a_1 \wedge f_2^*a_2\rangle \to \langle B, b\rangle$ s.t. $f; \pi_i = f_i$. Moreover, if $f; \pi_i = g; \pi_i$ (for $i = 1, 2$) in $\mathcal{B}$, then $f = g$ because of the universal property for products in $\mathcal{G}$ and $f, g\colon\langle C, f^*b \wedge g^*b\rangle \to \langle B, b\rangle$
   - **weak creation** let $\pi_i\colon B \to A_i$ be a product diagram in $\mathcal{B}$, then $\pi_i\colon\langle B, \pi_1^*a_1 \wedge \pi_2^*a_2\rangle \to \langle A_i, a_i\rangle$ is a product diagram in $\mathcal{G}$, for every $a_1 \in \mathcal{P}[A_1]$ and $a_2 \in \mathcal{P}[A_2]$.
3. $\mathcal{G}$ has finite products, because $\mathcal{B}$ does and π weakly creates finite limits.
4. It is immediate from the definition that $U^{\mathcal{G}}$ is full and faithful. $U^{\mathcal{G}}$ preserves limits, because $\pi \dashv U^{\mathcal{G}}$. In order to prove that $U^{\mathcal{G}}$ preserves exponentials, suppose that $\mathcal{B}(_\times A, B) \cong \mathcal{B}(_, B^A)$ and consider the following natural isomorphisms:
   $\mathcal{G}(\langle X, x\rangle \times U^{\mathcal{G}}A, U^{\mathcal{G}}B) =$ by $\pi \dashv U^{\mathcal{G}}$ and π preserves finite limits
   $\mathcal{B}(X\times A, B) \cong$ by assumption
   $\mathcal{B}(X, B^A) =$ by $\pi \dashv U^{\mathcal{G}}$
   $\mathcal{G}(\langle X, x\rangle, U^{\mathcal{G}}(B^A))$.

5. It is immediate from the definition that $\mathcal{M}^{\mathcal{G}}$ is closed w.r.t. identities and composition. To prove that $\mathcal{M}^{\mathcal{G}}$ is closed w.r.t. pullbacks, observe that

$$\begin{array}{ccc} \langle B, b\rangle & \xrightarrow{f} & \langle A, a\rangle \\ {\scriptstyle \mathrm{id}_B}\uparrow & & \uparrow{\scriptstyle \mathrm{id}_A} \\ \langle B, b \wedge f^* a'\rangle & \xrightarrow[f]{} & \langle A, a'\rangle \end{array}$$

(because pullbacks of identities in $\mathcal{B}$ always exist, and because of the way π weakly creates finite limits).

6. $I^{\mathcal{G}}[A]$ is an isomorphism, because subobjects in $\mathcal{M}^{\mathcal{G}}[\langle A, a\rangle]$ are uniquely represented by monos in $\mathcal{M}^{\mathcal{G}}$ (and there is a clear 1-1 correspondence between $\mathcal{P}[A]$ and monos in $\mathcal{M}^{\mathcal{G}}$ with codomain $U^{\mathcal{G}}A$). Moreover, $I^{\mathcal{G}}$ commutes with substitution, i.e. $I^{\mathcal{G}}[B](f^*a) = [\mathrm{id}_B \colon \langle B, f^*b\rangle \rightarrowtail \langle B, \top\rangle]$ for every $f \colon B \to A$ and $a \in \mathcal{P}[A]$, because of the way pullbacks of monos in $\mathcal{M}^{\mathcal{G}}$ are computed.

7. The only thing to verify is that the relevant morphisms have the expected domain and codomain, since $T^{\mathcal{G}}$ satisfies the equational axioms for strong monads iff T does (because of $T^{\mathcal{G}}$'s definition):

   - $f \colon \langle A, a\rangle \to \langle B, b\rangle$ implies $Tf \colon \langle TA, \Box_A a\rangle \to \langle TB, \Box_B b\rangle$, or equivalently
     $a \leq f^*b$ implies $\Box_A a \leq (Tf)^*(\Box_B b)$, in fact
     $(Tf)^*(\Box_B b) =$ by $(\Box T^*)$ of Definition 5.2
     $\Box_A(f^*b) \geq$ by the assumption and monotonicity of $\Box_A$
     $\Box_A a$
   - $\mathrm{t}^T_{A,B} \colon \langle A, a\rangle \times \langle TB, \Box_B b\rangle \to \langle T(A \times B), \Box_{A\times B}(a \times b)\rangle$, which amounts to $(\Box \mathrm{t})$
   - $\eta^T_A \colon \langle A, a\rangle \to \langle TA, \Box_A a\rangle$, which amounts to $(\Box \eta)$
   - $\mu^T_A \colon \langle T^2A, \Box_{TA}(\Box_A a)\rangle \to \langle TA, \Box_A a\rangle$, which amounts to $(\Box \mu)$.

8. Since T and $T^{\mathcal{G}}$ *coincide* on morphisms, $(U^{\mathcal{G}}, \mathrm{id})$ is a strong monad morphism from T to $T^{\mathcal{G}}$, provided $T^{\mathcal{G}}(U^{\mathcal{G}}A) = \langle TA, \Box_A \top\rangle = \langle TA, \top\rangle = U^{\mathcal{G}}(TA)$, which follows immediately from property $(\Box \top^*)$ of Definition 5.2.

9. $T^{\mathcal{G}}$ is an $\mathcal{M}^{\mathcal{G}}$-functor, provided for every $f \colon B \to A$, $a, a' \in \mathcal{P}[A]$ and $b \in \mathcal{P}[B]$ s.t. $b \leq f^*a$ and $a' \leq a$

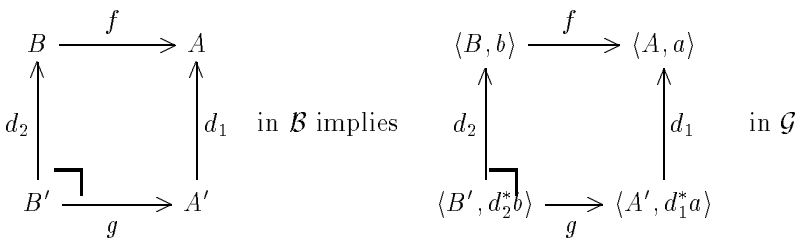
$$\begin{array}{ccc} \langle TB, \Box_B b\rangle & \xrightarrow{Tf} & \langle TA, \Box_A a\rangle \\ {\scriptstyle \mathrm{id}_B}\uparrow & & \uparrow{\scriptstyle \mathrm{id}_A} \\ \langle TB, \Box_B(b \wedge f^* a')\rangle & \xrightarrow[Tf]{} & \langle TA, \Box_A a'\rangle \end{array}$$

   i.e. $\Box_B(b \wedge f^*a') = (\Box_B b) \wedge (Tf)^*(\Box_A a')$. In fact
   $\Box_B(b \wedge f^*a') =$ by property $(\Box \wedge^*)$ of Definition 5.2
   $(\Box_B b) \wedge \Box_B(f^*a') =$ by property $(\Box T^*)$
   $(\Box_B b) \wedge (Tf)^*(\Box_A a')$.

10. Given $a \in \mathcal{P}[A]$ the only $m \in \mathcal{M}^{\mathcal{G}}$ s.t. $[m] = I^{\mathcal{G}}[A]a$ is $m = \mathrm{id}_A\langle A, a\rangle \hookrightarrow \langle A, \top\rangle$, therefore $I^{\mathcal{G}}[TA](\Box_A a) = [T^{\mathcal{G}}m]$ is equivalent to $[\mathrm{id}_{TA} \colon \langle TA, \Box_A a\rangle \hookrightarrow \langle TA, \top\rangle] = [\mathrm{id}_{TA} \colon \langle TA, \Box_A a\rangle \hookrightarrow \langle TA, \Box_A \top\rangle]$, or more simply $\top = \Box_A \top$, which follows from property $(\Box \top^*)$ of Definition 5.2.

∎

Proof of 5.8. First we prove that $\mathcal{E}$ is a class of display maps over $\mathcal{G}$, i.e. is closed w.r.t. pullbacks. In fact, for every $f\colon \langle B,b\rangle \to \langle A,a\rangle$



(since $d_2^*b = d_2^*b \wedge g^*(d_1^*a)$, because of $g^*(d_1^*a) = d_2^*(f^*a) \geq d_2^*b$).
Given $e = d\colon \langle A', d^*a\rangle \to \langle A,a\rangle \in \mathcal{E}$ and $b \in \mathcal{M}^{\mathcal{G}}[\langle A, d^*a\rangle]$, define $\forall_e(b) = (\forall_d b) \wedge a \in \mathcal{M}^{\mathcal{G}}[\langle A,a\rangle]$. We verify that $\forall_e$ is right adjoint to e^*, i.e. $d^*a' \wedge d^*a \leq b$ iff $a' \leq (\forall_d b) \wedge a$ for every $a' \in \mathcal{M}^{\mathcal{G}}[\langle A,a\rangle]$ and $b \in \mathcal{M}^{\mathcal{G}}[\langle A, d^*a\rangle]$.

- $d^*a' \wedge d^*a \leq b$ iff (since $a' \leq a$)
- $d^*a' \leq b$ iff (by definition of universal quantification)
- $a' \leq (\forall_d b)$ iff (since $a' \leq a$)
- $a' \leq (\forall_d b) \wedge a$.

The verification of the Beck-Chevalley condition is left to the reader. ∎

Proof of 5.9.

1. Since $\mathcal{M}^{\mathcal{G}}[\langle A,a\rangle]$ is isomorphic to $\mathcal{P}[A]$ restricted to the elements below a, we write $a' \in \mathcal{M}^{\mathcal{G}}[\langle A,a\rangle]$ for $[\mathrm{id}_A\colon \langle A,a'\rangle \to \langle A,a\rangle]$. Let $m = \mathrm{id}_A\colon \langle A,a_1\rangle \to \langle A,a\rangle \in \mathcal{M}^{\mathcal{G}}$, and $a_1' \in \mathcal{M}^{\mathcal{G}}[\langle A,a_1\rangle]$, then $\forall_m(a_1') = (a_1 \supset a_1') \wedge a \in \mathcal{M}^{\mathcal{G}}[\langle A,a\rangle]$. We verify that $\forall_m$ is right adjoint to m^*, i.e. $a' \wedge a_1 \leq a_1'$ iff $a' \leq (a_1 \supset a_1') \wedge a$ for every $a' \in \mathcal{M}^{\mathcal{G}}[\langle A,a\rangle]$ and $a_1' \in \mathcal{M}^{\mathcal{G}}[\langle A,a_1\rangle]$.

   - $a' \wedge a_1 \leq a_1'$ iff (by definition of pseudo-complement)
   - $a' \leq (a_1 \supset a_1')$ iff (since $a' \leq a$)
   - $a' \leq (a_1 \supset a_1') \wedge a$.

   The Beck-Chevalley condition amounts to prove that $((b \wedge f^*a_1) \supset (b \wedge f^*a_1')) \wedge b = f^*((a_1 \supset a_1') \wedge a) \wedge b$ in $\mathcal{P}[B]$ for every $\mathrm{id}_A\colon \langle A,a_1\rangle \to \langle A,a\rangle \in \mathcal{M}^{\mathcal{G}}$, $f\colon \langle B,b\rangle \to \langle A,a\rangle$ and $a_1' \in \mathcal{M}^{\mathcal{G}}[\langle A,a_1\rangle]$. Its verification is left to the reader.

2. Let $\mathcal{D}_A$ be the class of display maps $\mathcal{D}_A = \{\pi_1\colon B \times A \to B | B \in \mathcal{B}\}$ and similarly for $\mathcal{D}_{U^{\mathcal{G}}A}$, then we want to prove that $\mathcal{P}$ closed w.r.t. universal quantification along $\mathcal{D}_A$ implies $\mathcal{M}^{\mathcal{G}}$ closed w.r.t. universal quantification along $\mathcal{D}_{U^{\mathcal{G}}A}$. In fact, in Lemma 5.8 $\mathcal{E} = \mathcal{D}_{U^{\mathcal{G}}A}$, when $\mathcal{D} = \mathcal{D}_A$.

3. Let $\mathcal{D}$ be the class of first projections in $\mathcal{B}$, then $\pi_1\colon \langle X,x\rangle \times \langle A,a\rangle \to \langle X,x\rangle$ in $\mathcal{G}$ can be decomposed as a mono $m \in \mathcal{M}^{\mathcal{G}}$ followed by a map $e \in \mathcal{E}$ (as defined in Lemma 5.8), namely $m = id_{X\times A}\colon \langle X\times A, \pi_1^*x \wedge \pi_2^*a\rangle \to \langle X\times A, \pi_1^*x\rangle \in \mathcal{M}^{\mathcal{G}}$ and $e = \pi_1\colon \langle X\times A, \pi_1^*x\rangle \to \langle X,x\rangle \in \mathcal{E}$. Since $\forall_m$ and $\forall_e$ exist (by the first two claims of this theorem), then universal quantification along $\pi_1\colon \langle X,x\rangle \times \langle A,a\rangle \to \langle X,x\rangle$ is given by $\forall_m;\forall_e$.

4. The exponential $\langle B,b\rangle^{\langle A,a\rangle}$ can be defined as $\langle B^A, b^a\rangle$, where $b^a \in \mathcal{P}[B^A]$ is the interpretation of $f\colon A \Rightarrow B \vdash \forall x\colon A.a(x) \supset b(fx)$ prop.

∎

Proof of 5.11.

1. ϵ is bijective on objects and full, because the congruence $\equiv$ relates only morphisms with the same domain and codomain. In fact, we may take ϵ to be the identity on objects.

   ϵ preserves the terminal object and binary products, because ϵ is full, bijective on objects and the following sequents are derivable in the internal logic:

   - $x_1, y_1 : B_1, x_2, y_2 : B_2 \vdash x_1 =_{B_1} y_1, x_2 =_{B_2} y_2 \Longrightarrow \langle x_1, y_1\rangle =_{B_1 \times B_2} \langle x_2, y_2\rangle$
   - $x, y : B_1 \times B_2 \vdash x =_{B_1 \times B_2} y \Longrightarrow \pi_i(x) =_{B_i} \pi_i(y)$

   Finally, we prove that ϵ preserves pullbacks of monos in $\mathcal{M}^{\mathcal{G}}$, i.e. for every $f : \langle B, b\rangle \rightarrowtail \langle A, a\rangle$ and $a' \leq a$

   $$\begin{array}{ccc} \langle B, b\rangle & \xrightarrow{\epsilon(f)} & \langle A, a\rangle \\ {\scriptstyle \epsilon(\mathrm{id}_B)}\uparrow & & \uparrow{\scriptstyle \epsilon(\mathrm{id}_A)} \\ \langle B, b \wedge f^* a'\rangle & \xrightarrow[\epsilon(f)]{} & \langle A, a'\rangle \end{array}$$

   First of all $\epsilon(\mathrm{id}_B)$ is mono because of the way $\equiv$ is defined. Therefore, we need to check only the existence property. Suppose that

   $$\begin{array}{ccc} \langle B, b\rangle & \xrightarrow{\epsilon(f)} & \langle A, a\rangle \\ {\scriptstyle \epsilon(g)}\uparrow & & \uparrow{\scriptstyle \epsilon(\mathrm{id}_A)} \\ \langle C, c\rangle & \xrightarrow[\epsilon(h)]{} & \langle A, a'\rangle \end{array}$$

   then $g : \langle C, c\rangle \rightarrowtail \langle B, b \wedge f^* a'\rangle$, since $x : C \vdash c(x) \Longrightarrow a'(f(gx))$ is derivable from $x : C \vdash c(x) \Longrightarrow hx =_A f(gx)$ and $x : C \vdash c(x) \Longrightarrow a'(hx)$ in the internal logic. Therefore

   $$\begin{array}{ccc} \langle B, b\rangle & \xleftarrow{\epsilon(\mathrm{id}_B)} & \langle B, b \wedge f^* a'\rangle \\ {\scriptstyle \epsilon(g)}\uparrow & \nearrow_{\epsilon(g)} & \downarrow{\scriptstyle \epsilon(f)} \\ \langle C, c\rangle & \xrightarrow[\epsilon(h)]{} & \langle A, a'\rangle \end{array}$$

2. Since ϵ is bijective on objects and preserves finite products, then $\mathcal{E}$ has finite products, and we are left to show that it has also equalisers. In fact, the equaliser of $\epsilon(f), \epsilon(g) : \langle A, a\rangle \rightarrow \langle B, b\rangle$ is given by $\epsilon(\mathrm{id}_A) : \langle A, a \wedge eq(f, g)\rangle \rightarrowtail \langle A, a\rangle$, where $eq(f, g) \in \mathcal{P}[A]$ is the interpretation of $x : A \vdash fx =_A gx$ prop.

3. Since ϵ is bijective on objects, $\mathcal{M}^{\mathcal{E}}$ is closed w.r.t. identities and composition. Since ϵ is full and bijective on objects and preserves pullbacks of monos in $\mathcal{M}^{\mathcal{G}}$, then $\mathcal{M}^{\mathcal{E}}$ is closed w.r.t. pullbacks.

   To prove that $\mathcal{M}^{\mathcal{E}}$ has equalities, we have to show that $[\Delta_{\langle A, a\rangle}] \in \mathcal{M}^{\mathcal{E}}[\langle A, a\rangle]$. More precisely, we prove that $\epsilon(\langle \mathrm{id}_A, \mathrm{id}_A\rangle : \langle A, a\rangle \rightarrow \langle A \times A, a \times a\rangle)$ is equivalent to the mono $\epsilon(\mathrm{id}_{A \times A} : \langle A \times A, (a \times a) \wedge =_A \rangle \rightarrowtail \langle A \times A, a \times a\rangle)$ in $\mathcal{M}^{\mathcal{E}}$. This amounts to prove in the internal logic that the following sequents are true:

   - $x : A \vdash a(x) \Longrightarrow a(\pi_i(\langle x, x\rangle))$
   - $x : A \vdash a(x) \Longrightarrow x =_A x$,

- $y{:}A{\times}A \vdash a(\pi_1 y), a(\pi_2 y), \pi_1 y =_A \pi_2 y \Longrightarrow a(\pi_1 y)$
- $y{:}A{\times}A \vdash a(\pi_1 y), a(\pi_2 y), \pi_1 y =_A \pi_2 y \Longrightarrow y =_{A\times A} \langle \pi_1 y, \pi_1 y\rangle$

Only the proof of the last sequent uses rule (=).

4. We know (from Definition 5.6) that $\mathcal{M}^{\mathcal{G}}[\langle A, a\rangle]$ is isomorphic to the sub-poset $\mathcal{P}[A]$ consisting of the elements below a, and the isomorphism is given by $F{:}a' \mapsto [\mathrm{id}_A{:}\langle A, a'\rangle \to \langle A, a\rangle]$. Therefore, to prove that $J[\langle A, a\rangle]$ is an isomorphism it is enough to show that $G{:}a' \mapsto [\epsilon(\mathrm{id}_A{:}\langle A, a'\rangle \to \langle A, a\rangle)]$ is an isomorphism. Clearly, G is monotonic and surjective, so we need to prove only that $Ga_1 \leq Ga_2$ implies $a_1 \leq a_2$. In the internal logic $Ga_1 \leq Ga_2$ means that for some $f{:}A \to A$ the following sequents are true

- $x{:}A \vdash a_1(x) \Longrightarrow a_2(fx)$
- $x{:}A \vdash a_1(x) \Longrightarrow x =_A fx$

therefore $x{:}A \vdash a_1(x) \Longrightarrow a_2(x)$ is true, i.e. $a_1 \leq a_2$.

5. Since $T^{\mathcal{G}}$ is a strong monad over $\mathcal{G}$ and ϵ is full, bijective on objects and preserves finite products, we need only to check that $T^{\mathcal{G}}$ respects the congruence $\equiv$ over $\mathcal{G}$, i.e. $f \equiv g$ implies $T^{\mathcal{G}} f \equiv T^{\mathcal{G}} g$ for every $f, g{:}\langle A, a\rangle \to \langle B, b\rangle$. In the internal logic this amounts to prove that

$$() \ \frac{x{:}A \vdash a(x) \Longrightarrow b(fx)}{c{:}TA \vdash [x{\Leftarrow}c]a(x) \Longrightarrow [y{\Leftarrow}\mathrm{let}\ x{\Leftarrow}c\ \mathrm{in}\ [fx]]b(y)}$$

and

$$() \ \frac{x{:}A \vdash a(x) \Longrightarrow fx =_B gx}{c{:}TA \vdash [x{\Leftarrow}c]a(x) \Longrightarrow \mathrm{let}\ x{\Leftarrow}c\ \mathrm{in}\ [fx] =_{TB} \mathrm{let}\ x{\Leftarrow}c\ \mathrm{in}\ [gx]}$$

The first rule is derivable using ($\Box$-$\Longrightarrow$) and ($\Box$-T), while the second is derivable using ($\Box$-$\Longrightarrow$) and ($\Box$-=).

6. This follows immediately from the definition of $T^{\mathcal{E}}$ and the fact that ϵ preserves finite products.

7. $T^{\mathcal{E}}$ is an $\mathcal{M}^{\mathcal{E}}$-functor, because $T^{\mathcal{G}}$ is an $\mathcal{M}^{\mathcal{G}}$-functor, and ϵ commutes with computational types and maps pullbacks of monos in $\mathcal{M}^{\mathcal{G}}$ **onto** pullbacks of monos in $\mathcal{M}^{\mathcal{E}}$.

8. This follows immediately from the definition of J and $T^{\mathcal{E}}$.

∎

Proof of 5.13.

1. By Theorem 5.9, $\mathcal{M}^{\mathcal{G}}[\langle A, a\rangle]$ is cartesian closed. Since $J[\langle A, a\rangle]$ is an isomorphism, then also $\mathcal{M}^{\mathcal{E}}[\langle A, a\rangle]$ is cartesian closed. Moreover, implication in $\mathcal{M}^{\mathcal{E}}$ commutes with substitution, because implication in $\mathcal{M}^{\mathcal{G}}$ does and ϵ is full.

2. The proof is similar to that of Theorem 5.9. However, to prove that the Beck-Chevalley condition we need the following facts:

- ϵ preserves projections and pullbacks of projections, because it preserves finite products;
- the pullback square of a projection $\epsilon(\pi_1)$ along a morphism $\epsilon(f)$ in $\mathcal{E}$ can be chosen to be the image of a pullback square of π_1 along f in $\mathcal{G}$, because ϵ is bijective on objects and full.

3. The proof uses the first two claim of this theorem, and proceeds like that of Theorem 5.9.

4. Since ϵ is full and preserves products, the only thing to prove is uniqueness of $\Lambda(\epsilon(f))$, which amounts to the following instance of (=-λ)

$$() \ \frac{z{:}C, x{:}A \vdash c(z) \Longrightarrow f(z,x) =_B g(z,x)}{z{:}C \vdash c(z) \Longrightarrow \lambda x{:}A.f(z,x) =_{A\Rightarrow B} \lambda x{:}A.g(z,x)}$$

for every $f, g{:}C{\times}A \to B$ in $\mathcal{B}$ and $c \in \mathcal{P}[C]$.

Proof of 5.19.

1. The unit and counit of the adjunction are given by $x{:}\, A \vdash \eta_{\langle A,a\rangle}(x) \stackrel{\Delta}{\equiv} \{x\}{:}\, PA$ and $X{:}\, PA, x{:}\, A \vdash \epsilon_{\langle A,=_a\rangle}(X,x) \stackrel{\Delta}{\equiv} S_a(X) \wedge x \in_A X$. To prove that they are well-defined, i.e. $x, y{:}\, A \vdash a(x) \wedge x \simeq_A y \Longrightarrow \{x\} \simeq_{PA} \{y\}$ and $\epsilon_{\langle A,=_a\rangle}$ is strict (in his first argument), we need to use (ext-P).

   It is left to the reader to check that this forms an adjunction, in particular that the natural isomorphism between $\epsilon(f) \in \mathcal{E}(\langle A,a\rangle, S(\langle B,=_b\rangle))$ and $F \in \mathcal{T}(\iota(\langle A,a\rangle), \langle B,=_b\rangle)$ is given by $x{:}\, A, y{:}\, B \vdash F(x,y) \Longleftrightarrow a(x) \wedge y \in_B fx$.

2. "S is full and faithful" is equivalent to "the counit $\epsilon_{\langle A,=_a\rangle}$ is a natural isomorphism". The latter is an immediate consequence of $x{:}\, A \vdash E_a(x) \Longrightarrow S_a(\{x'|x' =_a x\}) \wedge x \in_A \{x'|x' =_a x\}$ and $x, y{:}\, A \vdash E_a(x), E_a(y), \{x'|x' =_a x\} \simeq_{PA} \{x'|x' =_a y\} \Longrightarrow x =_a y$.

3. We give a sequence of natural isomorphisms from $\mathcal{M}^{\mathcal{T}}[\langle A,=_a\rangle]$ to $\mathcal{M}^{\mathcal{E}}[S(\langle A,=_a\rangle)]$:

   $\mathcal{M}^{\mathcal{T}}[\langle A,=_a\rangle] \dot{\cong}$ by $\epsilon_{\langle A,=_a\rangle}$ iso
   $\mathcal{M}^{\mathcal{T}}[\iota(S(\langle A,=_a\rangle))] \dot{\cong}$ see Section 2.6 in [Pit81]
   $\{a' \in \mathcal{P}[PA] | a' \leq S_a\}$ by Theorem 5.11
   $\mathcal{M}^{\mathcal{E}}[\langle PA, S_a\rangle]$.

   We leave the check that $L[\langle A,=_a\rangle]$ is the composite of the natural isomorphisms given above.

4. We prove that $\mathcal{E}(x{\times}y, S(z)) \dot{\cong} \mathcal{E}(x, S(z^{\iota(y)}))$ for every $x, y \in \mathcal{E}$ and $z \in \mathcal{T}$

   $\mathcal{E}(x{\times}y, S(z)) \dot{\cong}$ because $\iota \dashv S$
   $\mathcal{T}(\iota(x{\times}y), z) \dot{\cong}$ because ι preserves finite limits
   $\mathcal{T}(\iota(x){\times}\iota(y), z) \dot{\cong}$ by definition of exponential
   $\mathcal{T}(\iota(x), z^{\iota(y)})$ because $\iota \dashv S$
   $\mathcal{E}(x, S(z^{\iota(y)}))$.

   Finally we prove that $\iota(S(z^{\iota(y)}))$ is an exponential of $\iota(S(z))$ to $\iota(y)$ in $\mathcal{T}$, i.e. it is isomorphic to $\iota(S(z))^{\iota(y)}$

   $\iota(S(z^{\iota(y)})) \cong$ because $\epsilon_x{:}\, \iota(S(x)) \to x$ is an iso
   $z^{\iota(y)}) \cong$ because $\epsilon_z{:}\, \iota(S(z)) \to z$ is an iso
   $\iota(S(z))^{\iota(y)}$.

5. S preserves exponentials, because $\epsilon_y{:}\, \iota(S(y)) \to y$ is an iso, and therefore $S(z^y)$ is isomorphic to the exponential $S(z^{\iota(S(y))})$ of $S(z)$ to $S(y)$ in $\mathcal{E}$.

6. The subobject classifier of $\mathcal{T}$ is $P(\langle 1, \top\rangle)$, i.e. $(P1, =_e)$ (see [Pit81]). So we have to prove that $\iota(\langle \Omega, \top\rangle) \cong \langle P1, =_e\rangle$

   $\iota(\langle \Omega, \top\rangle) \cong$ because $\Omega \cong P1$ in $\mathcal{B}$
   $\iota(\langle P1, \top\rangle) = \langle P1, \simeq_{P1}\rangle =$ by (ext-P)
   $\langle P1, =_e\rangle$.

   To prove that $\eta_{\langle \Omega,\top\rangle}$ is an isomorphism, show that $X{:}\, P\Omega \vdash f(X) \stackrel{\Delta}{\equiv} \top \in_\Omega X{:}\, \Omega$ is its inverse.

   In particular one has to derive $X{:}\, P\Omega \vdash S(X) \Longrightarrow X \simeq_{P\Omega} \{\top \in_\Omega X\}$, where $X{:}\, P\Omega \vdash S(X) \stackrel{\Delta}{\equiv} \exists x{:}\, \Omega. X =_e \{x\}$

   $X{:}\, P\Omega \vdash S(X) \Longrightarrow X \simeq \{\top \in X\}$ iff by (ext-P)
   $X{:}\, P\Omega \vdash S(X) \Longrightarrow X =_e \{\top \in X\}$ iff by definition of $=_e$
   $X{:}\, P\Omega, x{:}\, \Omega \vdash S(X) \Longrightarrow x \in X \leftrightarrow x \in \{\top \in X\}$ iff by definition of $x \in \{y\}$
   $X{:}\, P\Omega, x{:}\, \Omega \vdash S(X) \Longrightarrow x \in X \leftrightarrow x \simeq (\top \in X)$ iff by an analogue of (ext-P) for Ω
   $X{:}\, P\Omega, x{:}\, \Omega \vdash S(X) \Longrightarrow x \in X \leftrightarrow (x \leftrightarrow (\top \in X))$ iff by $x{:}\, \Omega \vdash x \Longleftrightarrow x \simeq \top$
   $X{:}\, P\Omega, x{:}\, \Omega \vdash S(X) \Longrightarrow x \in X \leftrightarrow ((x \simeq \top) \leftrightarrow (\top \in X))$ which follows easily from the definition of $S(X)$.

∎

Proof of 5.23.

1. $T^{\mathcal{T}}$ is a strong monad, i.e. it satisfies the necessary equations, because $T^{\mathcal{T}}$ is defined in terms of $T^{\mathcal{E}}$ using natural isomorphisms. We check the equations $T(\eta_x);\mu_x = \mathrm{id}_{Tx}$ and leave the others to the reader.
   $T^{\mathcal{T}}(\eta^{T^{\mathcal{T}}}_x);\mu^{T^{\mathcal{T}}}_x$ = by definition
   $(S;T^{\mathcal{E}};\iota)(\epsilon^{-1}_x;\iota(\eta^{T^{\mathcal{E}}}_{Sx}));\sigma_{T^{\mathcal{E}}(Sx)};\iota(\mu^{T^{\mathcal{E}}}_{Sx})$ = by functoriality of $S;T^{\mathcal{E}};\iota$
   $(S;T^{\mathcal{E}};\iota)(\epsilon^{-1}_x);(\iota;S;T^{\mathcal{E}};\iota)(\eta^{T^{\mathcal{E}}}_{Sx});\sigma_{T^{\mathcal{E}}(Sx)};\iota(\mu^{T^{\mathcal{E}}}_{Sx})$ = by naturality of σ
   $(S;T^{\mathcal{E}};\iota)(\epsilon^{-1}_x);\sigma_{Sx};(T^{\mathcal{E}};\iota)(\eta^{T^{\mathcal{E}}}_{Sx});\iota(\mu^{T^{\mathcal{E}}}_{Sx})$ = by functoriality of ι
   $(S;T^{\mathcal{E}};\iota)(\epsilon^{-1}_x);\sigma_{Sx};\iota(T^{\mathcal{E}}(\eta^{T^{\mathcal{E}}}_{Sx});\mu^{T^{\mathcal{E}}}_{Sx})$ = by $T(\eta_x);\mu_x = \mathrm{id}_{Tx}$ for $T^{\mathcal{E}}$
   $(S;T^{\mathcal{E}};\iota)(\epsilon^{-1}_x);\sigma_{Sx}$ = by functoriality of $S;T^{\mathcal{E}};\iota$ and by definition of σ
   $((S;T^{\mathcal{E}};\iota)\epsilon_x)^{-1};((T^{\mathcal{E}};\iota)\eta_{Sx})^{-1}$ = by functoriality of $T^{\mathcal{E}};\iota$
   $((T^{\mathcal{E}};\iota)(\eta_{Sx};S\epsilon_x))^{-1}$ = by one of the two equations for adjunctions
   $((T^{\mathcal{E}};\iota)\mathrm{id}_{Sx})^{-1} = \mathrm{id}_{T^{\mathcal{T}}x}$.

2. The verification that (ι,σ) satisfy the equations of a strong monad morphism is similar to the proof of $T(\eta_x);\mu_x = \mathrm{id}_{Tx}$ for $T^{\mathcal{T}}$ given above.

3. $T^{\mathcal{T}}$ is a $\mathcal{M}^{\mathcal{T}}$-functor, because S preserves finite limits and maps monos in $\mathcal{M}^{\mathcal{T}}$ to monos in $\mathcal{M}^{\mathcal{E}}$ (see Theorem 5.19), $T^{\mathcal{E}}$ is a $\mathcal{M}^{\mathcal{E}}$-functor (see Theorem 5.11), ι preserves finite limits and maps monos in $\mathcal{M}^{\mathcal{E}}$ to monos in $\mathcal{M}^{\mathcal{T}}$ (see Theorem 5.15).

4. K commutes with necessity up to iso, because the commuting square (corresponding to naturality of σ)

$$\begin{array}{ccc} T^{\mathcal{T}}(\iota x) & \xrightarrow[\sim]{\sigma_x} & \iota(T^{\mathcal{E}}x) \\ \uparrow{\scriptstyle T^{\mathcal{T}}(\iota m)} & & \uparrow{\scriptstyle \iota(T^{\mathcal{E}}m)} \\ T^{\mathcal{T}}(\iota x') & \xrightarrow[\sigma_{x'}]{\sim} & \iota(T^{\mathcal{E}}x') \end{array}$$

   is a pullback, since σ is a natural isomorphism.

∎

Contents

1 Informal semantics of EL_T **2**
- 1.1 Key features of ML_T and EL_T . . . 2
- 1.2 A set-theoretic semantics of EL_T . . . 2
- 1.3 Discussion on related semantics . . . 4

2 Typed predicate logics **5**
- 2.1 The typed predicate logic $ML(\Sigma)$ . . . 6
- 2.2 Additional types . . . 7
  - 2.2.1 Computational types: T . . . 7
  - 2.2.2 Functional types: $\Rightarrow$ . . . 7
- 2.3 Logical constants . . . 8
  - 2.3.1 True: $\top$ . . . 8
  - 2.3.2 Conjunction: $\wedge$ . . . 8
  - 2.3.3 Implication: $\supset$ . . . 8
  - 2.3.4 Universal quantification: $\forall$ . . . 8
  - 2.3.5 Equality: $=$ . . . 8
  - 2.3.6 Additional axioms for $=$ . . . 8
  - 2.3.7 Necessity: $\Box$ . . . 9
  - 2.3.8 Additional axioms for $\Box$ . . . 9
- 2.4 Higher Order Logic: HML . . . 9
- 2.5 Additional axioms for HML . . . 10
  - 2.5.1 Comprehension for powersets: Comp-P . . . 10
  - 2.5.2 Comprehension for functional types: Comp-$\Rightarrow$ . . . 10
  - 2.5.3 Comprehension for computational types: Comp-T . . . 10
- 2.6 Formal consequences . . . 10

3 Categorical semantics **12**
- 3.1 Strong monads . . . 12
- 3.2 Hyperdoctrine semantics . . . 13
  - 3.2.1 Hyperdoctrine interpretation . . . 15
- 3.3 Topos-theoretic semantics . . . 16
  - 3.3.1 Topos-theoretic interpretation . . . 18

4 Semantic for necessity **19**
- 4.1 Properties of strong monads w.r.t. a dominion . . . 19
- 4.2 Soundness of rules for necessity . . . 21

5 Completeness results **24**
- 5.1 Completeness w.r.t. the hyperdoctrine semantics . . . 25
  - 5.1.1 Generic models . . . 25
  - 5.1.2 Generic models for necessity . . . 26
- 5.2 From hyperdoctrine to topos-theoretic models . . . 27
  - 5.2.1 The Grothendieck construction . . . 28
  - 5.2.2 The modified Grothendieck construction . . . 29
  - 5.2.3 The Tripos construction . . . 31

A Appendix **36**