Logic Programming

A Category - Theoretic Framework

James Lipton (Wesleyan)

Categories and Logic

Overview...from Tarski to Lawvere

Tarski Models for many-sorted logic

Let $\mathfrak{M}$ be a model of first-order many-sorted logic. i.e. over a language with basic *sorts* $S = \{s_1, \ldots, s_n, \ldots\}$, compound sorts or *types* which are sequences of basic sorts including the (empty) sequence $\mathbf{1}$, and arrow expressions $\sigma_1 \longrightarrow \sigma_2$ where σ_1 is a sequence of basic sorts, and σ_2 is a basic sort, and *typed*

- constant symbols $c : \alpha$
- function symbols $f : \alpha \to \beta$
- relation (predicate) symbols $p : \alpha$ (e.g. $\textit{prime} : \text{int}$)

Tarski Models for many-sorted logic

Interpretation

$\mathfrak{M}$ is equipped with an interpretation function $[\![\_]\!]$, which maps sorts σ to sets (or domains) $\mathfrak{M}_\sigma$, and extends to compound sorts via $[\![s_1 \cdots\ s_n]\!] = [\![s_1]\!] \times \cdots \times [\![s_n]\!]$
Furthermore in $\mathfrak{M}$ we interpret

- constants c of type α as members of $[\![\alpha]\!] \equiv \mathfrak{M}_\alpha$,
- function symbols $f : \alpha \to \beta$ as functions $[\![f]\!] : [\![\alpha]\!] \to [\![\beta]\!]$,
- relation symbols p of type σ as subsets $[\![p]\!] \subseteq [\![\sigma]\!]$.

Interpreting open terms and formulas in $\mathfrak{M}$

We start by fixing a sequence $\vec{x}$ of variables $x_1 : \alpha_1, \ldots, x_n : \alpha_n$ containing all free variables occurring in interpreted terms and formulas. Letting $\vec{\alpha} = \alpha_1 \cdots \alpha_n$, we can now assign an interpretation $[\![\_]\!] = [\![\_]\!]^{\mathfrak{M}}_{\vec{x}}$ *relative to* $\vec{x}$ of all open terms and formulas as follows.

- $[\![x_i]\!]$ is the ith projection function: $[\![\vec{\alpha}]\!] \to [\![\alpha_i]\!]$,
- Suppose f is a function symbol of type: $\sigma \to \rho$, where $\sigma = \sigma_1 \cdots \sigma_k$ and t_i are terms of sort σ_i. Then $[\![f(t_1, \ldots, t_k)]\!] = \langle [\![t_1]\!], \ldots, [\![t_n]\!] \rangle [\![f]\!]$.
  Notice that this is a function from the domain $[\![\vec{\alpha}]\!]$ to $[\![\rho]\!]$:

$$[\![\vec{\alpha}]\!] \xrightarrow{\langle [\![t_1]\!], \ldots, [\![t_n]\!] \rangle} [\![\sigma]\!] \xrightarrow{f} [\![\rho]\!]$$

- If p is a relation symbol of sort σ then $[\![p(t_1, \ldots, t_k)]\!] = [\![\langle t_1, \ldots, t_n \rangle]\!]^{-1}([\![p]\!])$. This formulation of the meaning of atomic formulas coincides with the usual definition in Tarski semantics. Taking a special case in one-sorted logic

$$[\![p(f(x))]\!] = \{a : \mathfrak{M} \models p(f(a))\} = \{a : [\![f]\!](a) \in [\![p]\!]\} = [\![f]\!]^{-1}([\![p]\!])$$

Connectives...

In classical logic we interpret Boolean connectives $\wedge, \vee$ using $\bigcap, \bigcup$, and negation with complementation.

. . . and quantifiers

Let σ_1, σ_2 be sorts and π the projection of $[\![\sigma_1 \times \sigma_2]\!]$ to $[\![\sigma_2]\!]$. In the setting just given, quantification $\exists x, \forall x$ of a formula φ with two free variables x and y of sorts σ_1 and σ_2 can be captured using the operations $\exists_\pi, \forall_\pi : \mathit{Sub}([\![\sigma_1 \times \sigma_2]\!]) \to \mathit{Sub}([\![\sigma_2]\!])$ given by

quantifiers (cont.)

$$\begin{aligned}\exists_\pi(S) &= \{b \in [\![\sigma_2]\!] \mid \exists a \in [\![\sigma_1]\!]\,(a,b) \in S\} \\ &= Im(\pi) \\ \forall_\pi(S) &= \{b \in [\![\sigma_2]\!] \mid \forall a \in [\![\sigma_1]\!](a,b) \in S\}.\end{aligned}$$

Thus, $[\![\exists x.\varphi]\!] = \exists_\pi[\![\varphi]\!]$ and $[\![\forall x.\varphi]\!] = \forall_\pi[\![\varphi]\!]$ yield precisely the interpretation of quantifiers given by Tarski semantics.

Lawvere observed in the 1960's that these operations are precisely the left and right adjoints of the inverse image

$$\pi^{-1} : Sub(\llbracket \sigma_2 \rrbracket) \to Sub(\llbracket \sigma_1 \times \sigma_2 \rrbracket)$$

or, in lattice theoretic terms, that the pairs $\langle \exists_\pi, \pi^{-1} \rangle$ and $\langle \pi^{-1}, \forall_\pi \rangle$ each form a Galois correspondence. We illustrate this in the diagram below, where π is generalized to an arrow $\llbracket t \rrbracket : \llbracket \sigma \rrbracket \to \llbracket \rho \rrbracket$, the interpretation in Set of the term t.

The definitions for such a general $f = [\![t]\!]$, with $f : [\![\sigma]\!] \to [\![\rho]\!]$ and $S \subseteq [\![\sigma]\!]$ are

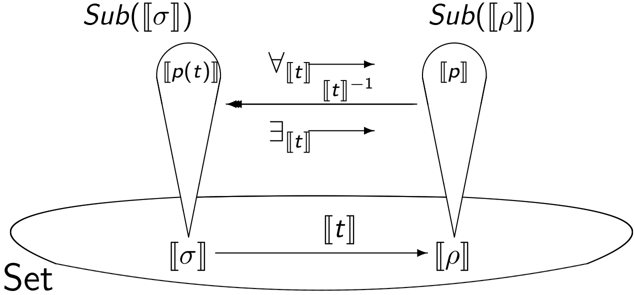
$$\begin{aligned}
\exists_f(S) &= \{b \in [\![\rho]\!] \mid \exists a \in [\![\sigma]\!]\,(f(a) = b \wedge a \in S)\} \\
&= Im(f) \\
\forall_f(S) &= \{b \in [\![\rho]\!] \mid \forall a \in [\![\sigma]\!](f(a) = b \Rightarrow a \in S)\}. \\
&= \{b \in [\![\rho]\!] : f^{-1}(b) \subseteq S\}
\end{aligned}$$

They really are adjoints: $\exists_f \dashv f^{-1} \dashv \forall_f$

$$\frac{S \subseteq f^{-1}(T)}{\exists_f(S) \subseteq T}$$

and

$$\frac{S \subseteq \forall_f(T)}{f^{-1}(S) \subseteq T}$$



Sub(⟦σ⟧)
Sub(⟦ρ⟧)
⟦p(t)⟧
⟦p⟧
∀⟦t⟧
⟦t⟧⁻¹
∃⟦t⟧
⟦t⟧
⟦σ⟧
⟦ρ⟧
Set

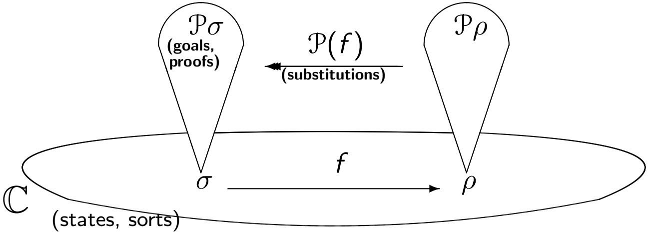
Logic in LP Doctrines

The same definitions just given for interpreting terms, formulas, types in a model, make sense if we take an indexed category $\mathcal{P} : \mathbb{C} \to \mathsf{Cat}$ where

- $\mathbb{C}$ has products and a terminal object, and plays the role of Set in interpreting types and terms.
- For each object $A = [\![\sigma]\!]$ of $\mathbb{C}$ the fiber $\mathcal{P}(A)$ interprets predicates p of type σ as objects. For each arrow (term) $A \xrightarrow{t} B$ in the base category, the functor $\mathcal{P}_t : \mathcal{P}(B) \to \mathcal{P}(A)$ plays the role of t^{-1} in the Set based model.

Indexed Categories

The following diagram illustrates the fundamental components of an indexed category:



For future reference, we have also indicated the logic programming concepts (**goals & proofs, substitutions, states & sorts**) formalized by these components.

Taking this one step further we can view an indexed category as a logic taking

- objects/arrows in the base as types and terms
- objects in the fibers as predicates,
- operations and structure in the fibers as connectives.
- arrows between predicates as proofs
- functors between fibers as generalized substitutions
- adjoints to these functors as generalized quantifiers

...although the quantifiers will not necessarily be well behaved unless they satisfy certain (Beck and Frobenius) conditions.

Step 1: Sorts and Terms in an FP (finite product + terminator) category

Start with an FP category $\mathbb{C}$, a many-sorted first order signature (S, Σ, Π) where

- S is a set of *primitive sorts*
- Σ is a set of function symbols accompanied by their sorts (f, σ) with constants represented as pairs $(c, (\cdot) \longrightarrow \sigma)$
- Π is a set of predicate symbols accompanied by their sorts (p, σ)

and finally a set of sorted variables V.

Definition

A $\mathbb{C}$-*structure* on (S, Σ, Π) is a function M that maps

- each primitive sort σ to an object $M(\sigma)$ of $|\mathbb{C}|$
- each compound sort $\sigma_1 \cdots \sigma_n$ to $M(\sigma_1) \times \cdots \times M(\sigma_n)$
- each function symbol of sort $\sigma_1 \cdots \sigma_n \longrightarrow \rho$ to an arrow $M(f) : M(\sigma_1) \times \cdots \times M(\sigma_n) \longrightarrow M(\rho)$. Constant symbols are mapped to arrows: $\mathbf{1} \xrightarrow{M(c)} M(\sigma)$

M maps predicate symbols (p, σ) to *monic* arrows $\stackrel{p}{\longmapsto} M(\sigma)$ We will often abuse language and write $M(p) \longmapsto M(\sigma)$.

Remark: A more general framework is obtained by mapping predicates to any class of arrows that are stable under pullbacks.

Terms

A $\mathbb{C}$-structure M induces an interpretation for all open terms over V. Given a sequence $\vec{x} = \langle x_1, \ldots, x_n \rangle$ of variables, with x_i of sort σ_i, we define $M(\vec{x}) = M(\vec{\sigma}) = M(\sigma_1) \times \cdots \times M(\sigma_n)$. Given a term t of sort ρ all of whose variables are among $\vec{x}$, we define the arrow $M_{\vec{x}(t)} : M(\vec{x}) \longrightarrow M(\rho)$ as follows:

- $\mathbf{t} = \mathbf{x_i}$: $M_{\vec{x}}(x_i)$ is the projection $\pi_i : M(\mathbf{x}) \longrightarrow M(\sigma_i)$. In this case ρ is σ_i.
- $\mathbf{t} = \mathbf{c}$: For a constant c of sort ρ, $M_{\vec{x}}(c)$ is defined as the following composition:

$$M(\vec{x}) \xrightarrow{!_{M(\vec{x})}} \mathbf{1} \xrightarrow{M(c)} M(\rho)$$

- $\mathbf{t} = \mathbf{f}(\mathbf{t_1}, \ldots, \mathbf{t_n})$: If each t_i is of sort α_i, the $M_{\vec{x}}$ is the following composition:

$$M(\vec{x}) \xrightarrow{\langle M_{\vec{x}}(t_1),\ldots,M_{\mathbf{x}}(t_n)\rangle} M(\vec{\alpha}) \xrightarrow{M(f)} M(\rho)$$

Given enough pullbacks, it is possible to interpret in $\mathbb{C}$ atomic formulas of first-order logic. Recall that for every predicate symbol p of Π of sort $\sigma_1 \cdots \sigma_n$ we have a monic $M(p) \rightarrowtail M(\vec{\sigma})$. For an atomic formula $\phi = p(t_1, \ldots, t_n)$ with all variables among $\vec{x}$, we interpret $M_{\vec{x}}(\phi)$ as the pullback of the monic $M(p) \rightarrowtail M(\vec{\sigma})$ along the arrow $\langle M_{\vec{x}}(t_1), \ldots, M_{\vec{x}}(t_n)\rangle$:

$$
\begin{array}{ccc}
M_{\vec{x}}(\phi) & \longrightarrow & M(p) \\
\downarrow & \lrcorner & \downarrow \\
M(\vec{x}) & \xrightarrow{\langle M_{\vec{x}}(t_1), \ldots, M_{\mathbf{x}}(t_n)\rangle} & M(\vec{\sigma})
\end{array}
$$

We will say that the formula ϕ is *true* in the interpretation when $M_{\mathbf{x}}(\phi) \longmapsto M(\mathbf{x})$ is an isomorphism. In the category Set this coincides with the usual definition of truth in Tarski semantics, i.e. every member of the interpretation of the sort of ϕ is in $M_{\vec{x}}(\phi)$.

Substitutions

Let $\theta = \{x_1/t_1, \ldots x_n/t_n\}$ be an idempotent substitution[1]. Assume that all the variables in $t_1 \ldots t_n$ are in the sequence $\vec{y}$. Then one can define a corresponding categorical substitution $\Theta_{\vec{y}}$ as the arrow:

$$M(\vec{y}) \xrightarrow{\langle M_{\vec{y}}(t_1),\ldots,M_{\vec{y}}(t_n)\rangle} M(\vec{x})$$

[1] $\theta\theta = \theta \quad \mathit{iff} \quad \mathit{dom}\theta \cap FV(\mathit{range}\theta) = \emptyset$.

⋆Exercise

It is easy to prove by structural induction (on s) that given a term s all of whose variables are among $\vec{x}$

$$M_{\vec{y}}(s\theta) = \Theta_{\vec{y}} M_{\vec{x}}(s).$$

Note: make use of (and prove) the fact that $h\langle u, v\rangle = \langle hu, hv\rangle$.

Application of the substitution θ to an atomic *predicate* ϕ whose sort is $M(\vec{x})$ is accomplished by taking the pullback of the monic $M_{\vec{x}}(\phi)$ along the arrow $\Theta_{\vec{y}}$ just defined.

Unifiers

Given two terms s and t of the same sort ρ with all variables in $\vec{x}$, if θ is a unifier, then $\Theta_{\vec{y}}$ equalizes $M_{\vec{x}}(s)$ and $M_{\vec{x}}(t)$, i.e., makes the following diagram commute

$$M(\vec{y}) \xrightarrow{\Theta_{\vec{y}}} M(\vec{x}) \underset{M_{\vec{x}}(t)}{\overset{M_{\vec{x}}(s)}{\rightrightarrows}} M(\rho)$$

In the appropriate category, if θ is a most general unifier, $\Theta_{\vec{y}}$ is an equalizer and conversely.

Let (S, Σ, Π) be a signature where S consists of a single sort ι representing the single type of terms in the Herbrand Universe. Let $LAT_\Sigma(\emptyset)$ be the category with

- objects: The natural numbers
- arrows: a distinguished arrow f from n to 1 for each function symbol of arity n in Σ. In particular an arrow c from 0 to 1 for each constant symbol c, together with all the projections and diagonal maps (and compositions thereof) required to make $LAT_\Sigma(\emptyset)$ into a finite product category, with the product of n and m given by $n + m$.

A more formal approach: make $LAT_\Sigma(\emptyset)$ the opposite category of the category Ord of finite ordinals with all set-theoretic maps between them. Thus we have, for example

$$1 \underset{\pi_1}{\overset{\pi_0}{\leftleftarrows}} \xrightarrow{\;\delta\;} 2$$

Now freely adjoin the function symbols to the underlying graph of this category and freely generate the finite product category with this graph. [See Lambek-Scott: Free cartesian closed categories generated by graphs and Polynomial categories]

Finally define M so that $M(c) = 0 \xrightarrow{c} 1$ and $M(f) = f$.

We should think of each object n as representing $\mathcal{H}^n$ where $\mathcal{H}$ is the Herbrand Universe for Σ. Then we have the following lemma, which we state without proof.

Lemma

In $LAT_{\Sigma}(\emptyset)$, given two arrows $M_{\vec{x}}(u)$ and $M_{\vec{x}}(t)$ with the same source n and target 1, the substitution θ is a most general unifier of u and t iff $\Theta_{\vec{y}}$ as defined above, is an equalizer of $M_{\vec{x}}(u)$ and $M_{\vec{x}}(t)$.

Consider the (idempotent) mgu $\theta = \{y/f(z,z), x/z\}$ of the two terms $g(f(x,x)), g(y)$. Giving them the common sort $M(x) \times M(y) = n_2$:

$$n_2 \xrightarrow{l} n_1 \xrightarrow{\langle id, id \rangle} n_2 \xrightarrow{f} n_1 \xrightarrow{g} n_1$$

$$n_2 \xrightarrow{r} n_1 \xrightarrow{g} n_1$$

The equalizer is Θ_z

$$M(z) = n_1 \xrightarrow{\langle id, M_z(f(z,z))\rangle} n_2 \qquad = \qquad n_1 \xrightarrow{\langle id, \langle id, id\rangle f\rangle} n_2$$

i.e.

$$n_1 \xrightarrow{\langle id, \langle id, id\rangle f\rangle} n_2 \underset{rg}{\overset{l\langle id, id\rangle fg}{\rightrightarrows}} n_1$$

Note that ($\star$exercise): given $A \xrightarrow{t} C$ and $B \xrightarrow{u} C$, the equalizer

$$E \xrightarrow{e} A \times B \underset{ru}{\overset{lt}{\rightrightarrows}} C$$

is the pullback of

$$\begin{array}{ccc} & & A \\ & & \downarrow{\scriptstyle t} \\ B & \xrightarrow[u]{} & C \end{array}$$

So we may use pullbacks instead of equalizers (provided we want to standardize apart the variables of t and u).

If we want t, u to share variables, we need a common domain to express this, e.g. $\{g(f(x, y)), g(y)\}$ as

$$n_2 \underset{\pi_2 g}{\overset{\langle \pi_1, \pi_2 \rangle fg}{\rightrightarrows}} n_1$$

or, equivalently

$$n_2 \underset{\pi_2 g}{\overset{fg}{\rightrightarrows}} n_1$$

Another example of a base category

Now we consider a different base category $\mathbb{C}$.

Building in Equational Theories

Let $\mathcal{T}_{\mathbb{N}}$ be the set of atomic equations true in the natural numbers $\mathbb{N}$. We now describe $LAT(\mathcal{T}_{\mathbb{N}})$ Form the LAT with objects n_k one for each of the natural numbers k and with proto-arrows

- $+ : n_2 \longrightarrow n_1$
- $\times : n_2 \longrightarrow n_1$
- $n_0 \xrightarrow{k} n_1$ for each natural number k
- All arrows defining a category with products $n_k \times n_j = n_{k+j}$ (e.g. all $\langle f, g \rangle$, all projections) and terminal object n_0

Let $\equiv$ be the congruence relation on arrows induced by all *FP* category equations e.g. $h\langle f, g\rangle = \langle hf, hg\rangle$, $\langle \pi_0 h, \pi_1 h\rangle = h$, $f! = !$, etc. *and* all equations in $\mathcal{T}$.
Now take arrows to be equivalence classes modulo $\equiv$. Thus, e.g. $[\langle 2, 2\rangle +] = [\langle\langle 2, 2\rangle \times\rangle]$

Attaching generic predicates

Definition (Generic Predicates)

Let X be a subobject of some object b in a finite product category $\mathbb{C}$, and let $\mathcal{D}$ be a family of arrows in $\mathbb{C}$.
We say X is a **generic subobject** of b with respect to the (display) maps $\mathcal{D}$ if

- For every arrow t in $\mathcal{D}$ targeted at b the pullback $t^{\#}(X)$ exists.
- No such pullback is an isomorphism.

Definition (The category $\mathbb{C}[X_1, \dots, X_n]$)

Let $\mathbb{C}$ be an FP category and $\vec{b} = b_1 \cdots b_n$ a sequence of objects of $\mathbb{C}$. Then $\mathbb{C}[\vec{b}]$ (or $\mathbb{C}[X_1, \dots, X_n]$), the *category obtained from* $\mathbb{C}$ *by freely adjoining indeterminate subobjects of* $\vec{b}$, is defined as follows:

objects: pairs $\langle A, S\rangle$ where $A \in |\mathbb{C}|$ and S is a sequence $S_1 \cdots S_n$ of finite sets $S_i \subset Hom_{\mathbb{C}}(A, b_i)$,

arrows: triples $\langle A, S\rangle \xrightarrow{f} \langle B, T\rangle$ where $A \xrightarrow{f} B$ is an arrow in $\mathbb{C}$ and $fT \subset S$, that is to say, for every i, $(1 \leq i \leq n)$ and every $t \in T_i$, $ft_i \in S_i$. The arrow f in $\mathbb{C}$ is called the **label** of $\langle A, S\rangle \xrightarrow{f} \langle B, T\rangle$. Composition of arrows is inherited from $\mathbb{C}$. Two arrows $\langle A, S\rangle \xrightarrow{f} \langle B, T\rangle$ and $\langle A', S'\rangle \xrightarrow{f'} \langle B', T'\rangle$ are **equal** if they have the same domain and range and if $f = f'$ in $\mathbb{C}$.

We also call $\mathbb{C}[X_1, \ldots, X_n]$ the category of generic predicates of **sort** $\vec{b}$.

Notice that an arrow in $\mathbb{C}[X_1, \ldots, X_n]$ may have an identity arrow in $\mathbb{C}$ as a label, and not even be an isomorphism in $\mathbb{C}[X_1, \ldots, X_n]$. We will be paying special attention to a certain class of such arrows.

Theorem

Let $\mathbb{C}$ be an FP category. The category $\mathbb{C}[X_1, \ldots, X_n]$ has

- *a terminal object $\langle \mathbf{1}, \vec{\emptyset} \rangle$, where $\vec{\emptyset}$ is the sequence $\emptyset, \ldots, \emptyset$ of length n,*
- *products: $\langle A, S \rangle \times \langle B, T \rangle = \langle A \times B, \pi_1 S \cup \pi_2 T \rangle$ where $A \xleftarrow{\pi_1} A \times B \xrightarrow{\pi_2} B$ is a product in $\mathbb{C}$.*

Furthermore, the functor $\mathbb{C} \xrightarrow{\iota} \mathbb{C}[X_1, \ldots, X_n]$ given by mapping objects A to $\langle A, \vec{\emptyset} \rangle$ and arrows $A \xrightarrow{f} B$ to $\langle A, \vec{\emptyset} \rangle \xrightarrow{f} \langle B, \vec{\emptyset} \rangle$, is a limit-preserving, full and faithful embedding.

Functoriality, faithfulness and fullness is obvious from the definition of morphism, composition and equality in $\mathbb{C}[X_1, \dots, X_n]$. Limit preservation follows from the fact that ι has a left adjoint, namely the forgetful functor U taking objects $\langle A, S \rangle$ to A and arrows to their labels.

Definition

In $\mathbb{C}[X_1, \ldots, X_n]$ define the *indeterminate subobjects* $X_1 \cdots X_n$ of sorts $b_1 \cdots b_n$ respectively, to be the subobjects $\langle b_i, J^i \rangle \xrightarrow{id_{b_i}} \langle b, \vec{\emptyset} \rangle$, where the J^i are the *basis vectors*

$$(J^i)_k = \begin{cases} \emptyset & \text{if} \quad i \neq k \\ \{id_{b_i}\} & \text{o.w.} \end{cases}$$

Theorem

The indeterminate subobjects X_i of b_i are generic with respect to the maps in the image of $Hom_{\mathbb{C}}(\_, b_i)$ *under* $\mathbb{C} \xrightarrow{\iota} \mathbb{C}[X_1, \ldots, X_n]$.

The following diagram is a pullback for any arrow $\langle A, \vec{\emptyset} \rangle \xrightarrow{t} \langle b_i, \vec{\emptyset} \rangle$:

$$\begin{array}{ccc} \langle A, tJ^i \rangle & \xrightarrow{t} & \langle b_i, \{J^i\} \rangle \\ {\scriptstyle id_A} \big\downarrow & \lrcorner & \big\downarrow {\scriptstyle id_{b_i}} \\ \langle A, \vec{\emptyset} \rangle & \xrightarrow[t]{} & \langle b_i, \vec{\emptyset} \rangle \end{array}$$

so $X(t) = \langle A, tJ^i \rangle \xrightarrow{id_A} \langle A, \vec{\emptyset} \rangle$ exists for all appropriate t. This arrow cannot be an isomorphism in $\mathbb{C}[X_1, \ldots, X_n]$: its inverse, which would have to be labelled with id_A, would have to satisfy $id_A t \in \emptyset$.

Definition

An object $\langle A, H \rangle$ is **atomic** if H is of the form tJ^i for a basis vector J^i and some arrow $A \xrightarrow[t]{} \sigma_i$. That is to say, H is the formula $X_i(t)$.

Definition

If A is an object of $\mathbb{C}$, we say that the monic $\langle B, S\rangle \stackrel{f}{\rightarrowtail} \langle A, \vec{\emptyset}\rangle$ is a canonical (representative of a) subobject of $\langle A, \vec{\emptyset}\rangle$ if B is A and the monic f is id_A.

The following theorems make precise the fact that $\mathbb{C}[X_1, \ldots, X_n]$ is called the category obtained by freely adjoining the indeterminate subobjects of the sorts $b_1 \cdots b_n$.

Lemma

Every object $\langle \sigma, S \rangle$ is representable as (i.e. equal on the nose to) the canonical intersection

$$\bigcap \{ t^{\#}(X_i) : t \in S_i, 1 \leq i \leq n \}$$

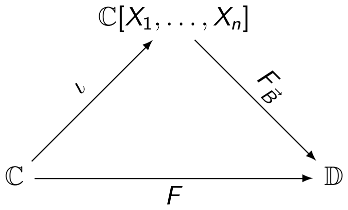
where the pullbacks are canonical:
$t^{\#}(X_i) = \langle \sigma, tJ^i \rangle = \langle \sigma, \emptyset \cdots \emptyset \underbrace{\{t\}}_{i} \emptyset \cdots \emptyset \rangle.$

Proof.

Immediate: Since $S_i = \bigcup \{ \{t\} : t \in S_i \}$, the indicated canonical intersection is precisely $\langle \sigma, S \rangle$. □

Theorem (Universal Mapping Property)

Suppose $F : \mathbb{C} \to \mathbb{D}$ *is a limit preserving functor from the finite-product category* $\mathbb{C}$ *to the finitely complete category* $\mathbb{D}$, *and that* $F(b_i) = d_i$ *for* $1 \leq i \leq n$. *Furthermore, let* $\vec{B} = B_1 \cdots B_n$ *be a sequence of subobjects of* $d_1 \cdots d_n$ *respectively, in* $\mathbb{D}$. *Then there is a limit-preserving functor* $F_{\vec{B}} : \mathbb{C}[X_1, \ldots, X_n] \to \mathbb{D}$, *unique up to isomorphism, such that the following diagram commutes and* $F_{\vec{B}}(X_i) = B_i$.



$F_{\vec{B}}$ *is called the* **evaluation** *functor induced by the* B_i.

Proof.

⋆ Define $F_{\vec{B}}$ on objects by

$$F_{\vec{B}}(\langle \sigma, S \rangle) = \varprojlim \{F(t)^{\#}(B_i) : t \in S_i, 1 \leq i \leq n\}$$

The universal mapping property of limits gives us the action on arrows: if $\langle \sigma, S \rangle \xrightarrow{f} \langle \sigma', S' \rangle$ is an arrow in $\mathbb{C}[X_1, \ldots, X_n]$ then $F_{\vec{B}}(\langle \sigma, S \rangle)$, the limit of the family of monics $\{F(t)^{\#}(B_i) : t \in S_i, 1 \leq i \leq n\}$ targeted at $F(\sigma)$, is also, by composing with $F(\sigma \xrightarrow{f} \sigma')$ and using properties of pullbacks and of arrows in $\mathbb{C}[X_1, \ldots, X_n]$, a cone over the family of monics $\{F(t)^{\#}(B_i) : t \in S'_i, 1 \leq i \leq n\}$. There is therefore a unique induced arrow $F\langle \sigma, S \rangle \xrightarrow{\theta} F\langle \sigma', S' \rangle$ which is the value of $F(\langle \sigma, S \rangle \xrightarrow{f} \langle \sigma', S' \rangle)$. The details, and those of the proof of limit preservation, are left to the reader. □

We are interested in a category $\mathbb{D}$ with richer structure, in which case we are able to sharpen this result a bit.

Corollary

Assume the category $\mathbb{D}$ in the preceding theorem is $\mathrm{Set}^{\mathbb{C}^o}$ and that F is the Yoneda embedding. Choose the sequence of subobjects B_i of $Fb_i = Hom_{\mathbb{C}}(\_, b_i)$ to be canonical, *that is to say, pointwise subsets of Fb_i, and take limits in $\mathrm{Set}^{\mathbb{C}^o}$ to be given pointwise (not just up to isomorphism, but on the nose). Then the evaluation functor $F_{\vec{B}}$ of the preceding theorem is unique.*

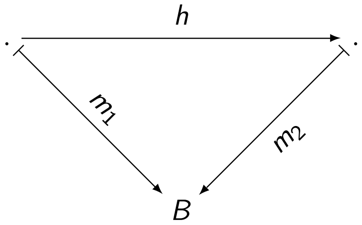
On the subobject lattices of $\mathbb{C}(\_, \sigma)$ in $\mathsf{Set}^{\mathbb{C}^{o}}$

Subobject structures

In any category a subobject of another object B is a monic m targeted at B. We can define a preorder on subobjects of B as follows:

$$m_1 \leq m_2$$

iff there is a (necessarily monic) arrow h such that



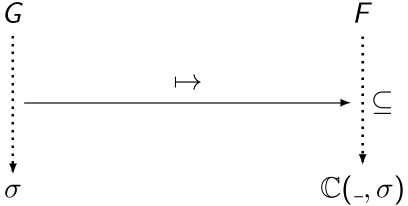
Two subobjects m_1 and m_2 are *equivalent* ($m_1 \simeq m_2$) if $m_1 \leq m_2$ and $m_2 \leq m_1$. This defines an equivalence relation on the collection of subobjects of B. The equivalence *classes* are usually what is meant by subobjects of B (rather than their members). The collection of $Sub_{\mathbb{C}}(B)$ of all such equivalence classes is called the subobject poset of B. Categories are often classified according to how much structure there is in $Sub_{\mathbb{C}}(B)$ (e.g. semilattice, lattice, complete lattice, Heyting algebra, Boolean algebra)

$\star$ Exercise: Equivalent subobjects are isomorphic.

Two abuses of language

- calling the individual monics subobjects (as we have already done).
- calling the *domain* A of a monic $A \stackrel{m}{\rightarrowtail} B$ a subobject of B. We sometimes do this in the category $\mathbf{Set}^{\mathbb{C}^o}$, especially when m is containment.

In our semantics we will be extending functors from $\mathbb{C}[X_1, \ldots, X_n]$ to $\mathsf{Set}^{\mathbb{C}^o}$ that send goal formulas G of type σ of $|\mathbb{C}|$ to subobjects (i.e. subfunctors) of the so-called *representable functors* $\mathbb{C}(\_, \sigma)$. We will refer to such subobjects F as *canonical* if $F \subset \mathbb{C}(\_, \sigma)$, i.e. if for every $\rho \in |\mathbb{C}|$ $F(\rho) \subset \mathbb{C}(\rho, \sigma)$



Functors and *co-sieves*[2]

We can think of any canonical subfunctor of the representable $\hat{\sigma} = \mathbb{C}(\_, \sigma)$ as being a class of arrows: namely a subclass of all the arrows targeted at σ.

[2] cribles

Lemma

A subclass F of $\mathbb{C}(\_, \sigma)$ *is a subfunctor iff it is a co-sieve: a class of arrows (targeted at* σ*) closed under left composition.*

i.e. $\rho \xrightarrow{f} \sigma \in F$ and $g \in \mathbb{C}(\alpha, \rho) \quad \Rightarrow \quad gf \in F$

Proof: $\star$exercise.

Theorem

In $\mathrm{Set}^{\mathbb{C}^{o}}$ *the class* $Sub(\mathbb{C}(\_, \sigma))$ *of subobjects of any representable functor forms a complete Heyting algebra, i.e. a distributive lattice with*

- *suprema* $F_1 \cup F_2$ *and* $\bigcup S$ *(of finite and arbitrary sets of subobjects)*
- *infima* $F_1 \cap F_2$ *and* $\bigcap S$
- *exponents* $F_1 \Rightarrow F_2$

Let S be a class of arrows targeted at some object σ of $\mathbb{C}$. Define its *interior* $Int(S)$ to be the largest co-sieve contained in S. Then we can define implication in $Sub(\sigma)$ by

$$F_1 \Rightarrow F_2 \quad = \quad Int(\overline{F_1} \cup F_2)$$

$\star$Exercise:

$$F_1 \Rightarrow F_2 \quad = \quad \{f : gf \in F_1 \to gf \in F_2\}$$

Since $\cap$ and $\cup$ map pairs of co-sieves to co-sieves, the sup and inf operations are well defined.

$\star$ In general, complementation does *not* preserve co-sieve structure unless $\mathbb{C}$ has all maps isos (i.e. is a groupoid).

If we define *falsity* $\bot$ as the constantly empty-valued functor we can define negation (pseudo-complementation) of a subobject (using the canonical representatives, i.e. the co-sieves) via $F \Rightarrow \bot$, i.e. the interior of the complement.
Truth $\top$ is the full subobject of $\mathbb{C}(\_, \sigma)$. I.e., taking canonical representatives, it is $\mathbb{C}(\_, \sigma)$ itself.

Taking $\vee, \wedge, \Rightarrow$ as logical connectives, defining the *internal logic* of $\mathrm{Set}^{\mathbb{C}^o}$, we do **not** have

$$F \vee \neg F \simeq \top \text{ or } \neg\neg F \simeq F$$

i.e. the internal logic (if $\mathbb{C}$ is not a groupoid) is intuitionistic.

If $\mathbb{C}$ is a groupoid $\mathit{Sub}(\mathbb{C}(\_, \sigma))$ has only two elements.

Predicates via indexed categories

The dependency of predicates on sorts (and later on underlying programs or states) is nicely captured and generalized via *indexed category structure*. Indexed categories resolve logic and logic programing structure in a clean way into a (vertical) basic logical component (the structure in the fibers) and the (horizontal) predicate logic and substitution component, which, as we shall soon see, is a special case of state change in logic programming.

Definition

A strict indexed $\mathbb{C}$-category (or just a $\mathbb{C}$-category) is a functor

$$\mathbb{C} \xrightarrow{P} \mathbb{CAT}.$$

An **indexed functor** from one $\mathbb{C}$-category p to another q is just a natural transformation from p to q. The category $P\sigma$ associated to the object σ of the base category $\mathbb{C}$ is called the **fiber** at σ. To each arrow $\sigma \xrightarrow{f} \rho$ between objects in the base category, P associates a functor Pf between the fibers.

If we relax the conditions to *pseudofunctors* then we have a (non-strict) *indexed category*. A pseudofunctor $P : \mathbb{A} \to \mathbb{B}$ only preserves composition and identity *up to (coherent) isomorphism*:

$$P(fg) \simeq P(f)P(g) \qquad P(id_A) \simeq id_{P(A)}$$

Notice that *pullback* $f^{\#}$ along an arrow f defines a pseudofunctor since $(fg)^{\#} \simeq f^{\#}g^{\#}$. They are not necessarily equal. Often pullback is just defined up to isomorphism.

However we will assume we are working in a strictly associative product category, with canonical pullbacks. See τ categories [Freyd-Scedrov]. Thus our first examples of predicate (indexed) categories will be strict.

Generic predicates as an indexed category

Definition

Let $\mathbb{C}$ be a τ category and $\boldsymbol{b} = b_1 \cdots b_n$ be a sequence of objects of $\mathbb{C}$. Then

$$\Pi_{\boldsymbol{b}} : \mathbb{C} \longrightarrow \mathbb{CAT},$$

the indexed cartesian category of generic predicates with sort $\boldsymbol{b}$, is defined as follows. Each fiber $\Pi_{\boldsymbol{b}}(\sigma)$ has **objects** the members of

$$FinPow(\mathbb{C}(\sigma, b_1)) \times \cdots \times FinPow(\mathbb{C}(\sigma, b_n))$$

where *FinPow* denotes the finite power set, i.e. sequences $S = S_1 \cdots S_n$ where each S_i is a finite set of arrows from σ to b_i, further endowed with the poset operation of pointwise containment: $S \leq T$ iff for all $i \quad S_i \subseteq T_i$.

To indicate the fiber in question, we will sometimes write objects as pairs $\langle\sigma, S\rangle$. The action of $\Pi_{\boldsymbol{b}}$ on arrows is given by

$$\Pi_{\boldsymbol{b}}(\sigma \xrightarrow{f} \rho) = f^{\#} : \Pi_{\boldsymbol{b}}(\rho) \longrightarrow \Pi_{\boldsymbol{b}}(\sigma)$$

Building a category of formulas

Definition

Let $\mathbb{C}$ be a finite-product category and $\boldsymbol{b} = b_1 \cdots b_n$ a finite sequence of objects of $\mathbb{C}$. A generalized first-order category of formulas (FOCF) $\mathbb{F}$ over $\mathbb{C}$ with signature $\stackrel{\text{def}}{\equiv} d_1 \cdots d_n$ of sort $\boldsymbol{b}$ is a predicate category with the following additional structure:

1. Every fiber $\mathbb{F}(\sigma)$ has an object $\top_\sigma$.
2. there are $\mathbb{C}$-indexed covariant bi-functors

$$\vee, \wedge : \mathbb{F} \times \mathbb{F} \longrightarrow \mathbb{F}$$

   and a bi-functor

$$\Rightarrow : \mathbb{F} \times \mathbb{F} \longrightarrow \mathbb{F}$$

   contravariant in its first coordinate and covariant in its second.
3. for every $\sigma \xrightarrow{f} \rho \in \mathbb{C}$ there are functors

$$\exists_f, \forall_f : \mathbb{F}(\sigma) \longrightarrow \mathbb{F}(\rho)$$

Definition

A *categorical signature* is a triple $(\mathbb{K}, \mathcal{D}, \mathbf{B})$ where $\mathbb{K}$ is a finite product category, $\mathcal{D}$ a family of arrows in $\mathbb{K}$ and $\mathbf{B}$ a distinguished class of *monics* in $\mathbb{K}$ satisfying the following condition: the pullback of any m in $\mathbf{B}$ along any coterminal arrow in $\mathcal{D}$ exists. $\mathcal{D}$ is usually taken closed under composition.

The *generic predicate category* $\mathbb{C}[X_1, \ldots, X_n]$ gives rise to the following signature:

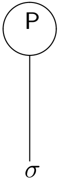
$$(\mathbb{C}[X_1, \ldots, X_n], \iota(\mathbb{C}), \{X_1, \ldots, X_n\})$$

Objects ocurring as sources or targets of members of $\mathcal{D}$ or targets of members of **B** are called *sorts*.
Arrows $f : \sigma \longrightarrow \rho$ of $\mathcal{D}$ are called *terms* of *insort* σ and *outsort* ρ. An arrow whose source is the terminal object and whose target is a sort σ is called a *constant* of sort σ. Members of B are called *predicate tokens*. The target of a predicate token is its sort.

We will say that a sort (i.e. an object of $\mathbb{K}$ ocurring as a source or target of a member of $\mathcal{D}$ or **B**) is an "object of $\mathbb{D}$".

An example: formula diagrams

Let $(\mathbb{K}, \mathcal{D}, \mathbf{B})$ be a categorical signature. A **formula diagram** P of sort σ over $(\mathbb{K}, \mathcal{D}, \mathbf{B})$ is a labelled diagram with a distinguished object σ of $\mathcal{D}$. For the purposes of this definition, such diagrams will be displayed as a bubble over a distinguished sort, as follows

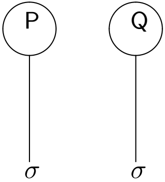


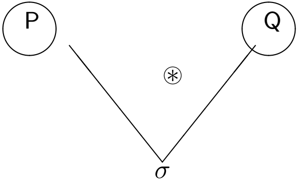
The class $\mathbb{F}(\mathbb{K}, \mathcal{D}, \mathbf{B})$ of formula diagrams over $(\mathbb{K}, \mathcal{D}, \mathbf{B})$ is given by the following inductive definition.

- For any object σ in $\mathcal{K}$, the identity arrow $\sigma = \!= \!= \sigma$ is a formula diagram of sort σ, called $\top_\sigma$.
- The pullback along an arrow $\sigma \xrightarrow{t} \rho$ in $\mathcal{D}$ of any predicate token X of sort ρ is a formula diagram $X(t)$ of sort σ. It is (a fortiori) monic, and is called an **atomic** formula diagram.

$$\begin{array}{c} X(t) \\ \big\downarrow \\ \sigma \end{array}$$

If P and Q are formula diagrams of sort σ (shown on the left), then so is the labelled diagram $P \circledast Q$ (shown on the right) below,





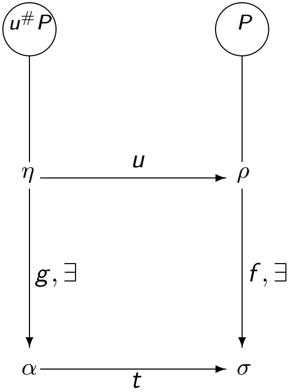
where $*$ is either the label $\Rightarrow$, or $\vee$ or $\wedge$.

If P is a formula diagram of sort ρ and $\rho \xrightarrow{f} \sigma$ is an arrow in $\mathcal{D}$, then the diagrams

are formula diagrams of sort σ, referred to as $\exists_f P$ and $\forall_f P$ respectively.

If P is a formula diagram of sort ρ and if $\alpha \xrightarrow{t} \rho$ is an arrow in $\mathcal{D}$, then the **formal pullback** $(t)^{\#}(P)$ is a formula diagram of sort σ, given by the following inductive definition:

1. if A is a predicate token then $(t)^{\#}(A)$ is just the normal pullback of A along t in $\mathbb{K}$.
2. $(t)^{\#}(P \circledast Q) = (t)^{\#}(P) \circledast (t)^{\#}(Q)$
3. $(t)^{\#}(\exists_f P) = \exists_g((u)^{\#}(P))$ where the bottom square in the following diagram is a (labelled) pullback:



For the instance to exist, the lower pullback must exist (in $\mathbb{C}$).

Program Syntax

We recall that uniform programming languages are given by the following data: a set of program formulas $\mathcal{P}$ and a set of goal formulas $\mathcal{G}$ recursively interdefined, as well as a notion of operational derivation $\vdash_o$ of sequents $P \vdash_o G$ where P is a finite subset of $\mathcal{P}$ and $G \in \mathbb{G}$. The sets $\mathcal{P}$ and $\mathcal{G}$ for Horn($\mathbb{C}$) and WHH($\mathbb{C}$) are defined below.

In each of the cases, *formula* will mean *formula diagram* over the categorical signature $\mathbb{C}[X]$. In particular *atomic formulas* will be of the form $A = (u)^{\#}(X_i)$ for some $X_i \in X$. $\top_\alpha$ will mean the identity $\alpha =\!=\!= \alpha$ for any object α.

Definition

- Horn($\mathbb{C}$) program formulae D and goal formulae G over a categorical signature are given by:

$$\begin{array}{rcl} G & ::= & \top \mid A \mid G \wedge G \mid G \vee G \\ D & ::= & A \mid G \Rightarrow A \mid D \wedge D \mid \forall_{x:\alpha} D \end{array}$$

- WHH ($\mathbb{C}$) program formulae and goal formulae over a categorical signature are given by:

$$\begin{array}{rcl} G & ::= & \top \mid A \mid G \wedge G \mid G \vee G \mid D \Rightarrow G \mid \exists_{x:\alpha} D \\ D & ::= & A \mid G \Rightarrow A \mid D \wedge D \mid \forall_{x:\alpha} D \end{array}$$

Given a base category $\mathbb{C}$ define a **WHH structure** to be a triple of $\mathbb{C}$-indexed categories

$$\mathsf{Goal}, \mathsf{Atom}, \mathsf{Prog} : \mathbb{C} \longrightarrow \mathbb{CAT}$$

endowed with the following indexed category morphisms:

- $\Rightarrow$: Goal $\times$ Atom $\longrightarrow$ Prog
- $\wedge$: Prog $\times$ Prog $\longrightarrow$ Prog
- $\wedge, \vee$: Goal $\times$ Goal $\longrightarrow$ Goal
- $\rightarrow$: Prog $\times$ Goal $\longrightarrow$ Goal

satisfying

1. Atom $\subseteq$ Goal
2. Atom $\subseteq$ Prog

Furthermore, for each $\sigma \xrightarrow{\theta} \rho$ in $\mathbb{C}$, there are functors

1. $\exists_\theta : \mathsf{Goal}(\sigma) \longrightarrow \mathsf{Goal}(\rho)$
2. $\forall_\theta : \mathsf{Prog}(\sigma) \longrightarrow \mathsf{Prog}(\rho)$

Clauses

In order to recapture the familiar notion of program as a set of *clauses* or formal sequents, with a head and tail, we carry out the translation described below. This translation process yields constituent clauses while cumulatively computing the sort extension that is taking place as quantifiers are removed. The effect of the translation is to replace outermost conjunctions with (finite) sets of formulae, and further translate the formulae by

- removing outer occurences of universal quantification, and
- replacing atoms A by the equivalent clause $\top \Rightarrow A$, where $\top = \top_\sigma$ has the same sort as the atom A.

We obtain clausal formulae of the form

$$tl_{cl} \Rightarrow hd_{cl}(tm_{cl})$$

accompanied by a sort-extending substitution (i.e. a projection).

Definition

We inductively define the translation κ by

- $\kappa(\varphi, A) = \{(\varphi, \top \Rightarrow A)\}$, $\top$ of the same sort as A.
- $\kappa(\varphi, G \Rightarrow A) = \{(\varphi, G \Rightarrow A)\}$
- $\kappa(\varphi, P_1 \wedge P_2) = \kappa(\varphi, P_1) \cup \kappa(\varphi, P_2)$
- $\kappa(\varphi, \forall_{f:\alpha\to\beta} P) = \kappa(f\varphi, P)$.

States and state vectors

Definition

Let $\mathbb{C}$ be an FP category and σ an object in $\mathbb{C}$. A σ-**state** is a pair $\langle P \mid A \rangle_\sigma$ where P is a program diagram over $\mathbb{C}$ of sort σ and A a goal diagram over the same category and sort. When clear from context, mention of the sort σ may be omitted.

A **state vector** is a finite sequence $\langle P_1 \mid A_1 \rangle \;\&\; \cdots \;\&\; \langle P_i \mid A_i \rangle \;\&\; \cdots \;\&\; \langle P_n \mid A_n \rangle$ of σ-states of the same sort and signature.

Resolution reduction rules

backchain

$$\langle P_1 \mid A_1\rangle \;\&\; \cdots \;\&\; \langle P_i \mid A_i\rangle \;\&\; \cdots \;\&\; \langle P_n \mid A_n\rangle \overset{\theta\pi,(G \Rightarrow A_i')}{\rightsquigarrow} \langle(\pi^{\#}P_1)\theta \mid (\pi^{\#}A_1)\theta\rangle \;\&\; \cdots \;\&\; \langle(\pi^{\#}P_i)\theta \mid G\theta\rangle \;\&\; \cdots \;\&\; \langle(\pi^{\#}P_n)\theta \mid (\pi^{\#}A_n)\theta\rangle$$

for atomic formula diagrams A_i, clause diagrams $(G \Rightarrow A_i')$ and substitution arrows $\theta\pi$, where

- $(\pi, G \Rightarrow A_i') \in \kappa(P_i)$
- θ is a unifier of the (sort-extended) atomic goal diagram $\pi^{\#}A_i$ and the head A_i' of the selected clause.

augment:

$$\langle P_1 \mid A_1 \rangle \;\&\; \cdots \;\&\; \langle P_i \mid A \Rightarrow B \rangle \;\&\; \cdots \;\&\; \langle P_n \mid A_n \rangle \overset{A}{\rightsquigarrow}$$
$$\langle P_1 \mid A_1 \rangle \;\&\; \cdots \;\&\; \langle P_i \wedge A \mid B \rangle \;\&\; \cdots \;\&\; \langle P_n \mid A_n \rangle$$

instance $\langle P_1 \mid A_1 \rangle \ \& \ \cdots \ \& \ \langle P_i \mid \exists_{x:\alpha} A_i \rangle \ \& \ \cdots \ \& \ \langle P_n \mid A_n \rangle \stackrel{\pi}{\leadsto}$
$\langle \pi^{\#} P_1 \mid \pi^{\#} A_1 \rangle \ \& \ \cdots \ \& \ \langle \pi^{\#} P_i \mid A_i \rangle \ \& \ \cdots \ \& \ \langle \pi^{\#} P_n \mid \pi^{\#} A_n \rangle$
where π is the projection $\sigma \times \alpha \to \sigma$.

and:

$$\langle P_1 \mid A_1 \rangle \;\&\; \cdots \;\&\; \langle P_i \mid A \wedge B \rangle \;\&\; \cdots \;\&\; \langle P_n \mid A_n \rangle \stackrel{\wedge}{\rightsquigarrow}$$
$$\langle P_1 \mid A_1 \rangle \;\&\; \cdots \;\&\; \langle P_i \mid A \rangle \;\&\; \langle P_i \mid B \rangle \;\&\; \cdots \;\&\; \langle P_n \mid A_n \rangle$$

or-right:

$$\langle P_1 \mid A_1 \rangle \;\&\; \cdots \;\&\; \langle P_i \mid A \vee B \rangle \;\&\; \cdots \;\&\; \langle P_n \mid A_n \rangle \overset{\vee_r}{\rightsquigarrow}$$
$$\langle P_1 \mid A_1 \rangle \;\&\; \cdots \;\&\; \langle P_i \mid B \rangle \;\&\; \cdots \;\&\; \langle P_n \mid A_n \rangle$$

or-left:
$\langle P_1 \mid A_1 \rangle \; \& \; \cdots \; \& \; \langle P_i \mid A \vee B \rangle \; \& \; \cdots \; \& \; \langle P_n \mid A_n \rangle \overset{\vee_l}{\rightsquigarrow}$
$\langle P_1 \mid A_1 \rangle \; \& \; \cdots \; \& \; \langle P_i \mid A \rangle \; \& \; \cdots \; \& \; \langle P_n \mid A_n \rangle$

A **null** resolution vector is one of the form
$\langle P_1 \mid \top \rangle \,\&\, \cdots \,\&\, \langle P_n \mid \top \rangle$

Definition

Let P be a program diagram and G a goal diagram over a category $\mathbb{C}$, that is to say, a categorical signature $(\mathbb{C}[X_1, \ldots, X_n], \iota(\mathbb{C}), \langle \mathbf{X_1}, \ldots, \mathbf{X_n} \rangle)$. Then an **SLD derivation** is a sequence of reductions starting with (singleton) state vector $\langle P \mid G \rangle$.

Definition

An **operational (SLD) proof** is a (finite) sequence of reductions $\langle P \mid G \rangle \rightsquigarrow \cdots \rightsquigarrow$ NULL where NULL is a null resolution vector.

Definition

A **computed answer substitution** θ is the composition of all the substitutions occurring in the backchain and instance steps of an SLD-proof.

We are now in a position to define *operational inference* $\vdash_o$ based on the notion of resolution. We must be careful, however, to distinguish between the universal role played by open formulas (formulas of non-terminator sort, corresponding to those containing free variables) in a *sequent* and the existential character they have (as formulas with *logic variables*) in a resolution sequence

$$\langle P \mid G \rangle \leadsto^{\theta} \leadsto \cdots \leadsto \mathrm{NULL}.$$

The intended meaning of such a derivation is that θ has successfully instantiated an existential query and that any variables remaining free after application of θ (i.e. if the source of θ is other than $\mathbf{1}$) are universal. Thus, for example, the existence of the derivation above should be equivalent to the assertion $P\theta \vdash_o G\theta$.

Definition

We will say that G is operationally derivable from P and write $P \vdash_o G$ iff there is a program $\tilde{P}$ and a formula $\tilde{G}$ such that $\langle \tilde{P} \mid \tilde{G} \rangle \leadsto^{\theta} \leadsto \cdots \leadsto$ NULL, with computed answer substitution θ, $P = \tilde{P}\theta$ and $G = \tilde{G}\theta$.

Lemma

If

$$\langle P(t) \mid G(t)\rangle \rightsquigarrow^{\theta} \rightsquigarrow \cdots \rightsquigarrow NULL$$

then

$$\langle P \mid G\rangle \rightsquigarrow^{(\theta t)} \rightsquigarrow \cdots \rightsquigarrow NULL.$$

Lemma

If

$$\langle P \mid G\rangle \rightsquigarrow^{(\theta t)} \rightsquigarrow \cdots \rightsquigarrow NULL$$

then

$$\langle P(t) \mid G(t)\rangle \rightsquigarrow^{\theta} \rightsquigarrow \cdots \rightsquigarrow NULL.$$

The preceding lemmas give us that

$$\langle P(t) \mid G(t)\rangle \leadsto^{\theta} \leadsto \cdots \leadsto NULL \text{ iff } \langle P \mid G\rangle \leadsto^{(\theta t)} \leadsto \cdots \leadsto NULL$$

Semantics

Indexing over state information

Let $\mathsf{Prog}_0 : \mathbb{C} \longrightarrow \mathbb{CAT}$ be the indexed category given by:

- $\mathsf{Prog}_0(\sigma) = \{P : P \text{ is a program of sort } \sigma\}$ with arrows generated by the identity arrows, and, for each pair of programs P, A, the *right-augment* arrows

$$P \wedge A \xleftarrow{\pi_A} P,$$

and

- $\mathsf{Prog}_0(\sigma \xrightarrow{\theta} \rho) = \mathsf{Prog}_0(\rho) \xrightarrow{\theta^{\#}} \mathsf{Prog}_0(\sigma)$

...more formally...

For each σ we can view $\text{Prog}_0(\sigma)$ as a preorder by ordering $P \leq Q$ iff $\kappa(id, P) \subseteq \kappa(id, Q)$ (the set of clauses of P is contained in the set of clauses of Q). We will label the containments $P \leq P \wedge A$ using the notation α_A so we can talk about the behavior of natural transformations with respect to these arrows.

Let GP be the category (the domain of the fibration) yielded by the Grothendieck construction $G(\mathbb{C}, \mathrm{Prog}_0)$, namely the category with objects pairs (P, σ) with P a program of sort σ and arrows

$$(P, \sigma) \xrightarrow{(\theta, \alpha_A)} (Q, \rho)$$

where $\sigma \xrightarrow{\theta} \rho$ in $\mathbb{C}$ and $\theta^{\#}(Q) \xleftarrow{\alpha_A} P$, whenever $\theta^{\#}(Q) = P \wedge A$.

We now define categories of goals and models *indexed over* GP:

$$\mathcal{G}\mathsf{l}, \mathcal{M} : \mathsf{GP} \to \mathbb{CAT}$$

as follows:

- $\mathcal{M}(P,\sigma) = Sub(\mathbb{C}(\_\_,\sigma))$
- $\mathcal{M}[(Q,\rho) \xrightarrow{(\theta,\alpha_A)} (P,\sigma)]$ by pullback along θ

and

- $\mathcal{G}\mathsf{l}(P,\sigma) = \{(G,P,\sigma) : G \text{ is a goal over } \sigma\}$
- $\mathcal{G}\mathsf{l}(P \wedge A,\sigma) \xrightarrow{\mathcal{G}\mathsf{l}(\theta,\alpha_A)} \mathcal{G}\mathsf{l}(\theta^{\#}(P),\rho)$ via

$$\boxed{(G, P \wedge A, \sigma) \mapsto (\theta^{\#}(A \Rightarrow G), \theta^{\#}(P), \rho)}$$

Recall $\langle P | A \Rightarrow G\rangle \rightsquigarrow \langle P \wedge A | G\rangle$

Then an operational interpretation may be defined as a GP-indexed functor

$$[\![\,]\!] : \mathcal{G}\mathsf{I} \to \mathcal{M}.$$

satisfying conditions 1,2,3,4,5 below.

1. triples $(G, P, \sigma) \in \mathcal{T}_\sigma$ are mapped to monics with target $Hom_{\mathbb{C}}(\_, \sigma)$
2. $[\![\top]\!]_P$ is mapped to the identity arrow on $Hom_{\mathbb{C}}(\_, \sigma)$.
3. $[\![A \wedge B]\!]_P = [\![A]\!]_P \cap [\![B]\!]_P$
4. $[\![A \vee B]\!]_P = [\![A]\!]_P \cup [\![B]\!]_P$
5. $[\![\exists_f A]\!]_P = \mathrm{Im}_{[\![f]\!]} [\![A]\!]_{P(f)}$

where $\mathrm{Im}_{[\![f]\!]}$ is the image along f, also denoted $\exists_{[\![f]\!]}$.

By naturality of $[\![\,]\!]$ we have, for every arrow $\sigma \xrightarrow{t} \rho$ in $\mathbb{C}$

$$\begin{array}{ccc}
\mathcal{G}\mathsf{l}(P,\rho) & \xrightarrow{[\![\,]\!]} & \mathcal{M}(P,\rho) \\
\Big\downarrow{\scriptstyle \mathcal{G}\mathsf{l}(id,t)} & & \Big\downarrow{\scriptstyle \mathcal{M}(id,t)} \\
\mathcal{G}\mathsf{l}(P(t),\sigma) & \xrightarrow[{[\![\,]\!]}]{} & \mathcal{M}(P(t),\sigma)
\end{array}$$

i.e.

$$[\![t]\!]^{\#}([\![G]\!]_P) = [\![G(t)]\!]_{P(t)}$$

The fact that the semantics must respect implication is also guaranteed by naturality of $[\![\,]\!]$ over the base category GP. If we fix the sort σ and vary programs, the commutativity of

$$\begin{array}{ccc}
\mathcal{G}\mathsf{I}(P \wedge A, \sigma) & \xrightarrow{[\![\,]\!]} & \mathcal{M}(P \wedge A, \sigma) \\
\Big\downarrow \mathcal{G}\mathsf{I}(\alpha_A) & & \Big\downarrow \mathcal{M}(\alpha_A) \\
\mathcal{G}\mathsf{I}(P, \sigma) & \xrightarrow[{[\![\,]\!]}]{} & \mathcal{M}(P, \sigma)
\end{array}$$

implies $[\![A \Rightarrow G\,]\!]_P = [\![G\,]\!]_{P \wedge A}$.

Soundness and Completeness

There is a natural partial order on interpretations.

Definition

$[\![\,]\!] \subseteq [\![\,]\!]'$ iff for all goal formulae A and every $P \in \mathcal{K}$ of the same sort, $[\![A]\!]_P \subseteq [\![A]\!]'_P$.

It suffices to check the order on atoms.

Lemma

If $[\![X_i(t)]\!]_P \subseteq [\![X_i(t)]\!]'_P$ *for all atoms* $X_i(t)$ *and all* $P \in \mathcal{K}$ *of the same sort, then* $[\![\,]\!] \subseteq [\![\,]\!]'$

Proof.

By a straightforward induction on the structure of goals. □

Definition

We shall say that an interpretation $[\![\,]\!]$ is a **model** of a program Q of sort σ if for every clause $(\varphi, tl_{cl} \Rightarrow X_i(tm_{cl})) \in \kappa(Q)$, we have $[\![tl_{cl}]\!]_{Q(\varphi)} \subseteq [\![X_i(tm_{cl})]\!]_{Q(\varphi)}$.

Proposition

If $[\![\,]\!]$ *is a model of a program* Q*, and* G *is a goal such that* $Q \vdash_o G$ *then* $[\![G]\!]_Q$ *is an isomorphism.*

Proof: By induction on length of the derivation.
Let $[\![\,]\!]$ be a model of Q, and let G be a goal such that $Q \vdash_o G$. Consider the first resolution rule of the proof of $Q \vdash_o G$: whose computed substitution (the composition of the substitutions along the way) is the identity.

$$\langle Q \mid G \rangle \overset{\theta_1\pi,(tl_{cl} \Rightarrow \tilde{G})}{\rightsquigarrow} \langle (\pi^{\#} Q)\theta_1 \mid tl_{cl}\theta_1 \rangle \rightsquigarrow \cdots \rightsquigarrow^{\varphi} \rightsquigarrow \ NULL$$

Backchain

By the induction hypothesis then, $[\![(tl_{cl}\theta_1)\varphi]\!]_Q$ is an isomorphism. But, $[\![(tl_{cl}\theta_1)\varphi]\!]_Q \subseteq [\![(\tilde{G}\theta_1)\varphi]\!]_Q = [\![((\pi^{\#}G)\theta_1)\varphi]\!]_Q = [\![G]\!]_Q$ which must then also be an isomorphism.

Conjunction

$$\langle Q \mid A \wedge B \rangle \overset{\wedge}{\rightsquigarrow} \langle Q \mid A \rangle \;\&\; \langle Q \mid B \rangle \rightsquigarrow \cdots \rightsquigarrow^{id} \rightsquigarrow \; NULL$$

By the induction hypothesis then we know that $[\![A]\!]_Q$ and $[\![B]\!]_Q$ are isomorphisms, as then is $[\![A \wedge B]\!]_Q$.

Disjunction

$$\langle Q \mid A_1 \vee A_2 \rangle \overset{\vee}{\leadsto} \langle Q \mid A_i \rangle \leadsto \cdots \leadsto^{id} \leadsto \ \ NULL.$$

By the induction hypothesis then we know that $[\![A_i]\!]_Q$ is an isomorphism, as then is $[\![A_1 \vee A_2]\!]_Q$.

Augment

$$\langle Q \mid A \Rightarrow B \rangle \overset{A}{\leadsto} \langle Q \wedge A \mid B \rangle \leadsto \cdots \leadsto^{id} \leadsto \; NULL.$$

By the induction hypothesis then, we know that $[\![A \Rightarrow B]\!]_Q = [\![B]\!]_{Q \wedge \{A\}}$ is an isomorphism.

Instance

Suppose the first step of the resolution sequence was an instance rule:

$$\langle Q \mid \exists_{x:\alpha} A\rangle \overset{\pi}{\rightsquigarrow} \langle (\pi^{\#})Q \mid A\rangle \rightsquigarrow \cdots \rightsquigarrow^{\psi} \rightsquigarrow \; NULL.$$

Instance (cont.)

By the induction hypothesis then, we know that $[\![(\psi)^{\#}A]\!]_Q$ is an isomorphism, i.e., $[\![\top_{\alpha\times\sigma'}]\!] \subseteq ([\![\psi]\!])^{\#}[\![A]\!]_{Q\pi}$ for some type σ'. Now using the fact that image is left-adjoint to pullback, we have $\mathrm{Im}_{[\![\psi]\!]}[\![\top_{\alpha\times\sigma'}]\!] \subseteq [\![A]\!]_{Q\pi}$. This is equivalent to $[\![\top_\sigma]\!] \subseteq [\![\exists_{x:\alpha}A]\!]_Q$ as we wanted to show.