Denotation of λ-Terms

$$[\![e]\!]\rho \in D$$

λ-term $e \in \Lambda$ environment $\rho \in D^V$

defined by recursion on the structure of e:

- $[\![x]\!]\rho = \rho(x)$
- $[\![\lambda x.e]\!]\rho = \mathrm{fun}(d \in D \mapsto [\![e]\!](\rho[x \mapsto d]))$
- $[\![e\,e']\!]\rho = \mathrm{app}([\![e]\!]\rho, [\![e']\!]\rho)$

updated environment, maps x to d and otherwise acts like ρ

Properties of $[\![-]\!]$:

Support

$$(\forall x \in fv(e).\ \rho(x) = \rho'(x)) \supset [\![e]\!]\rho = [\![e]\!]\rho'$$

Properties of $[\![-]\!]$:

Support

$$(\forall x \in fv(e).\ \rho(x) = \rho'(x)) \supset [\![e]\!]\rho = [\![e]\!]\rho'$$

$fv(e)$: set of free variables of e

(proved by induction on structure of e)

So for closed expressions ($fv(e) = \emptyset$) $[\![e]\!]\rho$ is independent of which ρ we use — just write $\boxed{[\![e]\!]}$ for $[\![e]\!]\rho$ in this case.

Properties of $[\![-]\!]$:

Support

$$(\forall x \in fv(e).\ \rho(x) = \rho'(x)) \supset [\![e]\!]\rho = [\![e]\!]\rho'$$

Compositionality

$$[\![e[e'/x]]\!]\rho = [\![e]\!](\rho[x \mapsto [\![e']\!]\rho])$$

(proved by induction on the structure of e, using the support property in case $e = \lambda x. e_1$)

Properties of $[\![-]\!]$:

Support

$$(\forall x \in fv(e).\ \rho(x) = \rho'(x)) \supset [\![e]\!]\rho = [\![e]\!]\rho'$$

Compositionality

$$[\![e[e'/x]]\!]\rho = [\![e]\!](\rho[x \mapsto [\![e']\!]\rho])$$

Soundness

$$e \Rightarrow c \quad \supset \quad [\![e]\!] = [\![c]\!]$$

proved by induction on the derivation of $e \Rightarrow c$

Eg. induction step for $\dfrac{e_1 \Rightarrow \lambda x.e \quad e[e_2/x] \Rightarrow c}{e_1 e_2 \Rightarrow c}$:

If $[\![e_1]\!]\rho = [\![\lambda x.e]\!]\rho$ & $[\![e[e_2/x]]\!]\rho = [\![c]\!]\rho$
then

$$\begin{aligned}
[\![e_1 e_2]\!]\rho &= app([\![e_1]\!]\rho, [\![e_2]\!]\rho) \\
&= app([\![\lambda x.e]\!]\rho, [\![e_2]\!]\rho) \\
&= app(fun(d \mapsto [\![e]\!]\rho[x \mapsto d]), [\![e_2]\!]\rho) \\
&= [\![e]\!]\rho[x \mapsto [\![e_2]\!]\rho] \\
&= [\![e[e_2/x]]\!]\rho \quad \text{by substitution prop.} \\
&= [\![c]\!]\rho \quad \checkmark
\end{aligned}$$

$\boxed{\begin{aligned}&app(fun(f), d)\\ &= f(d)\end{aligned}}$

N.B. converse of Soundness need not hold

E.g. $[\![\lambda y.(\lambda x.x)y]\!] = [\![\lambda y.y]\!]$ (see above),

but $\lambda y.(\lambda x.x)y \not\Rightarrow \lambda y.y$.

N.B. converse of Soundness need not hold

However, we can hope for

$$[\![e]\!] \neq \bot \supset e\Downarrow$$

termination

$$\triangleq \exists c.\, e \Rightarrow c$$

and hence $[\![e]\!] \neq \bot \equiv e\Downarrow$

since $e\Downarrow \supset \exists c.\, e \Rightarrow c$

$\supset \exists c.\, [\![e]\!] = [\![c]\!] \neq \bot$

denotation of c is fun$(\cdots) \neq \bot$

N.B. converse of Soundness need not hold

However, we can hope for

$$[\![e]\!] \neq \bot \supset e\Downarrow$$

This property implies

Computational Adequacy

$$[\![e_1]\!] \sqsubseteq [\![e_2]\!] \supset e_1 \leq_{ctx} e_2$$

Contextual pre-order: $\forall e.\ e[e_1/x]\Downarrow \supset e[e_2/x]\Downarrow$

N.B. converse of Soundness need not hold

However, we can hope for

$$[\![e]\!] \neq \bot \supset e\Downarrow$$

This property implies

Computational Adequacy

$$[\![e_1]\!] \sqsubseteq [\![e_2]\!] \supset e_1 \leq_{ctx} e_2$$

because if $[\![e_1]\!] \sqsubseteq [\![e_2]\!]$, then

$$e[e_1/x]\Downarrow \supset [\![e[e_1/x]]\!] \neq \bot \supset [\![e[e_2/x]]\!] = [\![e]\!][x \mapsto [\![e_2]\!]]$$

$$\sqsupseteq [\![e]\!][x \mapsto [\![e_1]\!]] \neq \bot$$

$$\supset e[e_2/x]\Downarrow$$

N.B. Converse of Soundness need not hold

However, we can hope for

$[\![e]\!] \neq \bot \supset e\Downarrow$

This property implies

Computational Adequacy

$$\boxed{[\![e_1]\!] \sqsubseteq [\![e_2]\!] \supset e_1 \leq_{ctx} e_2}$$

this holds when $i : (D \to D)_\bot \cong D$ is the "minimal invariant" for $((-) \to (-))_\bot$.

Locally continuous functors

Categories of domains

$\mathcal{Dom}$ = category whose objects are domains (ω-chain complete cpos with least elements) and whose morphisms are continuous functions.

$\mathcal{Dom}_\perp$ = category whose objects are domains and whose morphisms are strict continuous functions.

As usual (in category theory) $\mathcal{Dom}_\perp^{op}$ is the opposite of $\mathcal{Dom}_\perp$—same objects and morphisms given by:

$$\mathcal{Dom}_\perp^{op}(D,E) = \mathcal{Dom}_\perp(E,D)$$

$\mathrm{Dom}_\perp$, $\mathrm{Dom}_\perp^{op}$ & $\mathrm{Dom}_\perp^{op} \times \mathrm{Dom}$ are examples of

cpo-enriched category

- an ordinary category $\mathcal{C}$, plus
- cpo structure on each hom $\mathcal{C}(A,B)$

such that composition

$$\mathcal{C}(A,B) \times \mathcal{C}(B,C) \longrightarrow \mathcal{C}(A,C)$$
$$(f,g) \longmapsto g \circ f$$

is a continuous function

Functors $F : \mathcal{C} \to \mathcal{C}'$ are cpo-enriched, or locally continuous, if each function

$$\mathcal{C}(A,B) \longrightarrow \mathcal{C}'(FA,FB) \qquad f \longmapsto F(f)$$

is continuous.

All the constructions on domains determine locally continuous functors:

$$\boxed{(-)_\perp : \mathbf{Dom}_\perp \longrightarrow \mathbf{Dom}_\perp}$$

$$f \in D \multimap E \mapsto f_\perp \in D_\perp \multimap E_\perp$$

$$f_\perp(x) \triangleq \begin{cases} f(d) & \text{if } x = d \in D \\ \perp & \text{if } x = \perp \end{cases}$$

All the constructions on domains determine locally continuous functors:

$$\boxed{(-)\times(-) : \mathbf{Dom}_\perp \times \mathbf{Dom}_\perp \longrightarrow \mathbf{Dom}_\perp}$$

$$\begin{matrix} f_1 \in D_1 \multimap E_1 \\ f_2 \in D_2 \multimap E_2 \end{matrix} \mapsto f_1 \times f_2 \in D_1 \times D_2 \multimap E_1 \times E_2$$

$$(f_1 \times f_2)(d_1, d_2) \triangleq (f_1(d_1), f_2(d_2))$$

All the constructions on domains determine locally continuous functors:

$$\boxed{(-)\otimes(-) : \mathbf{Dom}_\perp \times \mathbf{Dom}_\perp \longrightarrow \mathbf{Dom}_\perp}$$

$$\begin{array}{l} f_1 \in D_1 \multimap E_1 \\ f_2 \in D_2 \multimap E_2 \end{array} \mapsto f_1 \otimes f_2 \in D_1 \otimes D_2 \multimap E_1 \otimes E_2$$

$$\begin{cases} (f_1 \otimes f_2)(\perp) \triangleq \perp \\ (f_1 \otimes f_2)(d_1, d_2) \triangleq \begin{cases} (f_1(d_1), f_2(d_2)) & \text{if } f_1(d_1) \neq \perp \ \& \ f_2(d_2) \neq \perp \\ \perp & \text{otherwise} \end{cases} \end{cases}$$

All the constructions on domains determine locally continuous functors:

$$\boxed{(-)\oplus(-) : \mathbf{Dom}_\perp \times \mathbf{Dom}_\perp \longrightarrow \mathbf{Dom}_\perp}$$

$$\begin{array}{l} f_1 \in D_1 \multimap E_1 \\ f_2 \in D_2 \multimap E_2 \end{array} \mapsto f_1 \oplus f_2 \in D_1 \oplus D_2 \multimap E_1 \oplus E_2$$

$$(f_1 \oplus f_2)(x) \triangleq \begin{cases} f_1(d_1) & \text{if } x = d_1 \in D_1{\downarrow} \ \& \ f_1(d_1) \neq \perp \\ f_2(d_2) & \text{if } x = d_2 \in D_2{\downarrow} \ \& \ f_2(d_2) \neq \perp \\ \perp & \text{otherwise} \end{cases}$$

All the constructions on domains determine locally continuous functors:

N.B.

$$\boxed{(-)\to(-) : \mathbf{Dom}_\perp^{op} \times \mathbf{Dom}_\perp \to \mathbf{Dom}_\perp}$$

$$\begin{array}{l} f_1 \in E_1 \multimap D_1 \\ f_2 \in D_2 \multimap E_2 \end{array} \mapsto f_1 \to f_2 \in (D_1 \to D_2) \multimap (E_1 \to E_2)$$

$$(f_1 \to f_2)(f) \triangleq f_2 \circ f \circ f_1$$

(Note that $f_1 \to f_2$ is a strict function, because...)

All the constructions on domains determine locally continuous functors:

$$\boxed{(-)\multimap(-) : \mathbf{Dom}_\perp^{op} \times \mathbf{Dom}_\perp \to \mathbf{Dom}_\perp}$$

$$\begin{array}{l} f_1 \in E_1 \multimap D_1 \\ f_2 \in D_2 \multimap E_2 \end{array} \mapsto f_1 \multimap f_2 \in (D_1 \multimap D_2) \multimap (E_1 \multimap E_2)$$

$$(f_1 \multimap f_2)(f) \triangleq f_2 \circ f \circ f_1$$

Positive & negative occurrences

An occurrence of X in $\Phi(X)$ is negative if one passes through an odd number of left-hand branches of $\to$ or $\multimap$ constructions between the occurrence and the root of the parse tree; the occurrence is positive otherwise.

E.g.

$$(\overset{-ve}{X} \to \overset{+ve}{X})_\perp$$

$$(\underset{+ve}{X} \multimap \mathbb{Z}_\perp) \to \mathbb{Z}_\perp$$

Given a domain construction $\Phi(X)$, by separating +ve & −ve occurrences of X, we get a locally continuous functor

$$F : \mathbf{Dom}_\perp^{op} \times \mathbf{Dom}_\perp \longrightarrow \mathbf{Dom}_\perp$$

such that $\Phi(D) = F(D, D)$ for all $D \in \mathbf{Dom}_\perp$

E.g. from $\Phi(X) = (X \to X)_\perp$ we get

$$F(-,+) = ((-) \to (+))_\perp$$

from $\Phi(X) = (X \multimap \mathbb{Z}_\perp) \to \mathbb{Z}_\perp$

$$F(-,+) = ((+) \multimap \mathbb{Z}_\perp) \to \mathbb{Z}_\perp$$

Given a domain construction $\Phi(X)$, by separating +ve & −ve occurrences of X, we get a locally continuous functor

$$F : \mathrm{Dom}_\perp^{op} \times \mathrm{Dom}_\perp \longrightarrow \mathrm{Dom}_\perp$$

such that $\Phi(D) = F(D, D)$ for all $D \in \mathrm{Dom}_\perp$

solutions $\longleftrightarrow$ "invariants"

$D \cong \Phi(D)$ $\qquad\qquad$ $D \cong F(D, D)$