Semantics of Programming Languages

Andrzej Murawski and Nikos Tzevelekos

Lecture 3: Sound models of PCF

Complete partial orders

- A **partially ordered set (poset)** is a structure $(P, \sqsubseteq)$ where $\sqsubseteq$ is a relation on the set P such that, for all $x, y \in P$,

Reflexivity	*Transitivity*	*Anti-symmetry*
$x \sqsubseteq x$	$x \sqsubseteq y \wedge y \sqsubseteq z \implies x \sqsubseteq z$	$x \sqsubseteq y \wedge y \sqsubseteq x \implies x = y$

- $\bot \in P$ is a **least element** of a poset P if $\bot \sqsubseteq x$ for all $x \in P$.

- Given $S \subseteq P$, an $x \in P$ is an **upper bound** of S if $y \sqsubseteq x$ for all $y \in S$.
  x is a **least upper bound (lub)** if, additionally, $x \sqsubseteq x'$ for any upper bound x' of S.

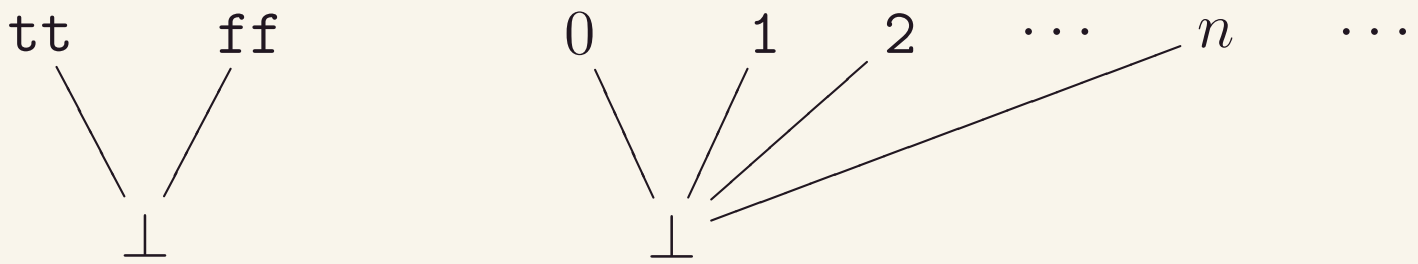
- A **chain** (or *ω-chain*) in P is a sequence of elements $(x_i)_{i \in \mathbb{N}}$ of P such that $x_i \sqsubseteq x_{i+1}$ for all $i \in \mathbb{N}$.
  A lub on $(x_i)_{i \in \mathbb{N}}$ is a lub of the set $\{x_i \mid i \in \mathbb{N}\}$. We write this as $\bigsqcup_{i \in \mathbb{N}} x_i$.

A **complete partial order (cpo)** is a poset $(D, \sqsubseteq)$ which has a least element, and a lub for every chain.

Examples

Flat cpo's Given a set X, we can form a cpo $X_\bot$ adjoining an element $\bot \notin X$ and defining the order by: $x \sqsubseteq y \iff x = \bot \vee x = y$

E.g. the cpos $\mathbb{B}_\bot$ (where $\mathbb{B} = \{\mathtt{tt}, \mathtt{ff}\}$) and $\mathbb{N}_\bot$:



Products Let D, E be cpo's. Their cartesian product $D \times E$ with the *componentwise* ordering:

$$(x, y) \sqsubseteq (x', y') \iff x \sqsubseteq x' \wedge y \sqsubseteq y'$$

is a cpo. Least element and lub's are also determined componentwise:

$$\bot_{D\times E} = (\bot_D, \bot_E) \qquad \bigsqcup_{i\in\mathbb{N}} (x_i, y_i) = (\bigsqcup_{i\in\mathbb{N}} x_i, \bigsqcup_{i\in\mathbb{N}} y_i)$$

Continuous functions

We consider an appropriate notion of function between cpo's.
Let D, E be cpo's. Consider a function $f : D \to E$.

- f is **monotonic** if, for all $x, y \in D$, $x \sqsubseteq y \implies f(x) \sqsubseteq f(y)$.
- f is **continuous** if it is monotonic and, additionally, for all chains $(x_i)_{i\in\mathbb{N}}$ in D,
$$f(\bigsqcup_{i\in\mathbb{N}} x_i) = \bigsqcup_{i\in\mathbb{N}} f(x_i)\,.$$

We write $D \Rightarrow E$ for the the set of continuous functions $f : D \to E$.
This is also a cpo, ordered componentwise:

$$f \sqsubseteq g \iff \forall x \in D.\, f(x) \sqsubseteq g(x)$$

The least element of $D \Rightarrow E$ is the function $\bot_{D\Rightarrow E} = x \mapsto \bot_E$.

Lub's are computed componentwise: $(\bigsqcup_{i\in\mathbb{N}} f_i)(x) = \bigsqcup_{i\in\mathbb{N}} f_i(x)$.

Notes

- We usually refer to a cpo $(D, \sqsubseteq)$ simply as D. We may use subscripts, like in $\sqsubseteq_D$ or $\bot_D$, to specify the cpo structure of D.
- Products between cpo's extend to finite products $D_1 \times \cdots \times D_n$ in a straightforward manner.
- Note that if $f : D \to E$ is monotonic and $(x_i)_{i \in \mathbb{N}}$ is a chain in D then $(f(x_i))_{i \in \mathbb{N}}$ is a chain in E.
  Moreover, $f(x_i) \sqsubseteq f(\bigsqcup_{i \in \mathbb{N}} x_i)$, for each i, and therefore

$$\bigsqcup_{i \in \mathbb{N}} f(x_i) \sqsubseteq f(\bigsqcup_{i \in \mathbb{N}} x_i)$$

  Hence, to prove that f is continuous it suffices to show:

$$f(\bigsqcup_{i \in \mathbb{N}} x_i) \sqsubseteq \bigsqcup_{i \in \mathbb{N}} f(x_i)$$

- Note that if D is finite then every monotonic $f : D \to E$ is continuous.

The Fixpoint Theorem

A **fixpoint** of a function $f : D \to D$ is an $x \in D$ such that $f(x) = x$.
x is a **least fixpoint** if, additionally, $x \sqsubseteq x'$ for every fixpoint x' of f.

Theorem. *Let D be a cpo and $f : D \to D$ a continuous function. Then f has a least fixpoint $\mathtt{lfp}(f)$, which is defined explicitly by:*

$$\mathtt{lfp}(f) = \bigsqcup_{i \in \mathbb{N}} f^i(\bot) \qquad (\textit{where } f^0(\bot) = \bot,\ f^{i+1}(\bot) = f(f^i(\bot)))$$

Proof. We first show that $(f^i(\bot))_{i \in \mathbb{N}}$ is a chain, i.e. $f^i(\bot) \sqsubseteq f^{i+1}(\bot)$, by induction on i: $\bot \sqsubseteq f(\bot)$ by leastness, and $f^i(\bot) \sqsubseteq f^{i+1}(\bot)$ implies $f^{i+1}(\bot) \sqsubseteq f^{i+2}(\bot)$ by monotonicity.

Next we show that $\mathtt{lfp}(f)$ is a fixpoint of f. By continuity of f:

$$f(\bigsqcup_{i \in \mathbb{N}} f^i(\bot)) = \bigsqcup_{i \in \mathbb{N}} f(f^i(\bot)) = \bigsqcup_{i \in \mathbb{N}} f^{i+1}(\bot) \overset{(Ex.)}{=} \bigsqcup_{i \in \mathbb{N}} f^i(\bot)$$

The Fixpoint Theorem

A **fixpoint** of a function $f : D \to D$ is an $x \in D$ such that $f(x) = x$.
x is a **least fixpoint** if, additionally, $x \sqsubseteq x'$ for every fixpoint x' of f.

Theorem. *Let D be a cpo and $f : D \to D$ a continuous function. Then f has a least fixpoint $\mathtt{lfp}(f)$, which is defined explicitly by:*

$$\mathtt{lfp}(f) = \bigsqcup_{i \in \mathbb{N}} f^i(\bot) \qquad (\textit{where } f^0(\bot) = \bot,\ f^{i+1}(\bot) = f(f^i(\bot)))$$

Proof. Finally, suppose that x is a fixpoint of f. We show by induction on i that $f^i(\bot) \sqsubseteq x$ for all $i \in \mathbb{N}$.
For $i = 0$ this is clear, and if $f^i(\bot) \sqsubseteq x$ then:

$$f^{i+1}(\bot) = f(f^i(\bot)) \sqsubseteq f(x) = x$$

Thus, x is an upper bound for $(f^i(\bot))_{i \in \mathbb{N}}$ and so $\bigsqcup_{i \in \mathbb{N}} f^i(\bot) \sqsubseteq x$. $\square$

The cpo model of PCF

We take $\mathcal{M}$ to be the collection (*category*) of cpo's and continuous functions. Each type A is translated to a cpo $[\![A]\!]$ in $\mathcal{M}$ as follows.

$$[\![\texttt{bool}]\!] = \mathbb{B}_\perp \quad [\![\texttt{nat}]\!] = \mathbb{N}_\perp \quad [\![A \to B]\!] = [\![A]\!] \Rightarrow [\![B]\!]$$

For any context $\Gamma = \{x_1 : A_1, \ldots, x_m : A_m\}$, we set:

$$[\![\Gamma]\!] = [\![A_1]\!] \times \cdots \times [\![A_m]\!]$$

In case $\Gamma = \emptyset$, we take $[\![\Gamma]\!] = \{\perp\}$.

We next extend the translation to typed terms so that each $\Gamma \vdash M : A$ is mapped to a continuous function $[\![\Gamma \vdash M : A]\!] : [\![\Gamma]\!] \to [\![A]\!]$.

In full formality, contexts Γ should be seen as lists, rather than sets, with the additional typing rule:

$$\frac{\Gamma \vdash M : A}{\Gamma' \vdash M : A}\ \Gamma' \text{ a permutation of } \Gamma$$

The cpo model of PCF (ctd)

The translation is defined by induction on the type derivations. For each typed constant $\Gamma \vdash \mathsf{c} : A$, we have ($\Gamma, \vdash, A$ are omitted for economy):

$$[\![\mathtt{t}]\!]\,\vec{z} = \mathtt{tt} \qquad [\![\mathtt{f}]\!]\,\vec{z} = \mathtt{ff}$$

$$[\![\mathtt{zero?}\ ]\!]\vec{z}\,y = \begin{cases} \mathtt{tt} & \text{if } y = 0 \\ \mathtt{ff} & \text{if } y > 0 \\ \bot & \text{if } y = \bot \end{cases} \qquad [\![\mathtt{succ}\,]\!]\vec{z}\,y = \begin{cases} y+1 & \text{if } y \neq \bot \\ \bot & \text{if } y = \bot \end{cases}$$

$$[\![\mathtt{cond}_{A_b}]\!]\vec{z}\,y y_1 y_2 = \begin{cases} y_1 & \text{if } y = \mathtt{tt} \\ y_2 & \text{if } y = \mathtt{ff} \\ \bot & \text{if } y = \bot \end{cases} \qquad [\![\mathtt{pred}\,]\!]\vec{z}\,y = \begin{cases} y-1 & \text{if } y > 0 \\ 0 & \text{if } y = 0 \\ \bot & \text{if } y = \bot \end{cases}$$

$$[\![n]\!]\,\vec{z} = n \qquad [\![\mathbf{Y}_A]\!]\vec{z}\,f = \mathtt{lfp}(f) = \bigsqcup_{i \in \mathbb{N}} f^i(\bot)$$

Note that each $[\![\Gamma \vdash \mathsf{c} : A]\!] : [\![\Gamma]\!] \to [\![A]\!]$, which is why the $\vec{z}$ input is needed ($\vec{z}$ ranges over elements of $[\![\Gamma]\!]$).

The cpo model of PCF (ctd)

For the rest of the rules:

$$[\![\Gamma \vdash x_i : A]\!]\,\vec{z} \;=\; z_i$$

$$\frac{[\![\Gamma \vdash M : A \to B]\!] = f \quad [\![\Gamma \vdash N : A]\!] = g}{[\![\Gamma \vdash MN : B]\!]\,\vec{z} = (f(\vec{z}))(g(\vec{z}))}$$

$$\frac{[\![\Gamma, x : A \vdash M : B]\!] = f}{[\![\Gamma \vdash \lambda x^A.M : A \to B]\!]\,\vec{z}\,z' = f(\vec{z}, z')}$$

Example

We show that $[\![(\lambda x^{\mathtt{bool}}.\mathtt{cond}\ x\ 24\ 42)(\mathtt{zero?}\ 42)]\!] = 42$.

- $[\![\mathtt{zero?}\ 42]\!] = [\![\mathtt{zero?}\ ]\!][\![42]\!] = [\![\mathtt{zero?}\ ]\!]42 = \mathtt{ff}$
- $[\![\mathtt{cond}\ x\ 24\ 42]\!]z = ([\![\mathtt{cond}\ ]\!]z)([\![x]\!]z)([\![24]\!]z)([\![42]\!]z) =$
  $[\![\mathtt{cond}\ ]\!]\, z\, 24\, 42 = (\ 24 \text{ if } z = \mathtt{tt},\ 42 \text{ if } z = \mathtt{ff},\ \bot \text{ if } z = \bot\ )$
- $[\![\lambda x.\ \mathtt{cond}\ x\ 24\ 42]\!]z = [\![\mathtt{cond}\ x\ 24\ 42]\!]z$
- $[\![(\lambda x.\mathtt{cond}\ x\ 24\ 42)(\mathtt{zero?}\ 42)]\!] = [\![\lambda x.\ \mathtt{cond}\ x\ 24\ 42]\!][\![\mathtt{zero?}\ 42]\!] = 42$

Note that, although each $[\![\vdash M : A]\!]$ has domain $\{\bot\}$, by abuse of notation we may write $[\![\vdash M : A]\!]$ where we should be writing instead $[\![\vdash M : A]\!]\bot$.

Another example

$\mathtt{fact} \equiv \mathbf{Y}(\lambda f.\lambda x.\, N), \quad N \equiv \mathtt{cond}\ (\mathtt{zero?}\ x)\ 1\ (\mathtt{mult}\, x\, (f(\mathtt{pred}\, x))$

$(\Gamma = \{f : \mathtt{nat} \to \mathtt{nat}, x : \mathtt{nat}\})$

- $[\![f(\mathtt{pred}\, x)]\!]\, \phi\, y = \phi(\pi(y))$ (where $\pi : \mathbb{N}_\perp \to \mathbb{N}_\perp =$ predecessor)
- $[\![\mathtt{mult}\, x\, (f(\mathtt{pred}\, x))]\!]\, \phi\, y \overset{(Ex.)}{=} y \times \phi(\pi(y))$
- $[\![N]\!]\, \phi\, y = \begin{cases} \perp & \text{if } [\![\mathtt{zero?}\ x]\!]\phi y = \perp \\ 1 & \text{if } [\![\mathtt{zero?}\ x]\!]\phi y = \mathtt{tt}\ [\![N]\!]\, \phi\, y = \\ y \times \phi(\pi(y)) & \text{if } [\![\mathtt{zero?}\ x]\!]\phi y = \mathtt{ff} \end{cases}$

  $\begin{cases} \perp & \text{if } y = \perp \\ 1 & \text{if } y = 0 \\ y \times \phi(\pi(y)) & \text{if } y > 0 \end{cases}$

  $(\Gamma = \emptyset)$
- $[\![\lambda f.\lambda x.N]\!]\, \phi\, n = [\![N]\!]\, \phi\, n$

Another example

$\mathtt{fact} \equiv \mathbf{Y}(\lambda f.\lambda x.\, N), \quad N \equiv \mathtt{cond}\ (\mathtt{zero?}\ x)\ 1\ (\mathtt{mult}\, x\, (f(\mathtt{pred}\, x))$

$(\Gamma = \{f : \mathtt{nat} \to \mathtt{nat}, x : \mathtt{nat}\})$

- $[\![f(\mathtt{pred}\, x)]\!]\, \phi\, y = \phi(\pi(y))$ (where $\pi : \mathbb{N}_\perp \to \mathbb{N}_\perp =$ predecessor)
- $[\![\mathtt{mult}\, x\, (f(\mathtt{pred}\, x))]\!]\, \phi\, y \overset{(Ex.)}{=} y \times \phi(\pi(y))$
- $[\![N]\!]\, \phi\, y = \begin{cases} \perp & \text{if } y = \perp \\ 1 & \text{if } y = 0 \\ y \times \phi(\pi(y)) & \text{if } y > 0 \end{cases}$

  $(\Gamma = \emptyset)$
- $[\![\lambda f.\lambda x.N]\!]\, \phi\, n = [\![N]\!]\, \phi\, n$
- $[\![\mathtt{fact}\,]\!] = \mathtt{lfp}([\![\lambda f.\lambda x.N]\!]) = \bigsqcup_{i \in \mathbb{N}} [\![\lambda f.\lambda x.N]\!]^i(\perp_{\mathbb{N}_\perp \to \mathbb{N}_\perp})$

Another example (ctd)

$\texttt{fact} \equiv \mathbf{Y}(\lambda f.\lambda x.\, N), \quad N \equiv \texttt{cond}\ (\texttt{zero?}\ x)\ 1\ (\texttt{mult}\, x\, (f(\texttt{pred}\, x))$

- $[\![N]\!]\, \phi\, y = \begin{cases} \bot & \text{if } y = \bot \\ 1 & \text{if } y = 0 \\ n \times \phi(\pi(n)) & \text{if } y > 0 \end{cases}$
- $[\![\texttt{fact}\,]\!] = \texttt{lfp}([\![\lambda f.\lambda x.N]\!]) = \bigsqcup_{i \in \mathbb{N}} [\![\lambda f.\lambda x.N]\!]^i(\bot_{\mathbb{N}_\bot \to \mathbb{N}_\bot}) = \bigsqcup_{i \in \mathbb{N}} \chi^i(\bot)$

  where:

$$\chi^0(\bot)\, y = \bot$$

$$\chi^{i+1}(\bot)\, y = \begin{cases} \bot & \text{if } y = \bot \\ 1 & \text{if } y = 0 \\ y \times \chi^i(\bot)(\pi(y)) & \text{if } y > 0 \end{cases}$$

  which is the same as the defition of factorial we gave before, lifted from partial functions to total functions with $\bot$.

Correctness

Theorem. *For any term* $\vdash M : A$*, if* $M \Downarrow V$ *then* $[\![M]\!] = [\![V]\!]$.

We need a lemma...

Substitution Lemma

Lemma. *For all* $\Gamma, x : A \vdash M : B$ *and* $\Gamma \vdash N : A$*, and all* $\vec{z} \in [\![\Gamma]\!]$*,*

$$[\![\Gamma \vdash M[N/x] : B]\!]\,\vec{z} \;=\; [\![\Gamma, x : A \vdash M : B]\!]\,\vec{z}\,([\![\Gamma \vdash N : A]\!]\,\vec{z})$$

Proof. By induction on the structure of M.

- $M \equiv \mathsf{c}$. Then, $[\![M[N/x]]\!]\vec{z} = [\![\Gamma \vdash \mathsf{c}]\!]\vec{z} = [\![\Gamma, x : A \vdash \mathsf{c}]\!]\vec{z}y$, for all $y \in [\![A]\!]$, so taking $y = [\![N]\!]\vec{z}$ we obtain the required equality.

- $M \equiv x_i \not\equiv x$. Then, $[\![M[N/x]]\!]\vec{z} = z_i = [\![\Gamma, x : A \vdash \mathsf{c}]\!]\vec{z}y$, for all $y \in [\![A]\!]$, so taking $y = [\![N]\!]\vec{z}$ we obtain the required equality.

- $M \equiv x$. Then, $[\![M]\!]\,\vec{z}\,([\![N]\!]\vec{z}) = [\![N]\!]\vec{z} = [\![M[N/x]]\!]\,\vec{z}$.

Substitution Lemma (ctd)

Lemma. *For all* $\Gamma, x : A \vdash M : B$ *and* $\Gamma \vdash N : A$, *and all* $\vec{z} \in [\![\Gamma]\!]$,

$$[\![\Gamma \vdash M[N/x] : B]\!]\,\vec{z} \;=\; [\![\Gamma, x : A \vdash M : B]\!]\,\vec{z}\,([\![\Gamma \vdash N : A]\!]\,\vec{z})$$

Proof. By induction on the structure of M.

- $M \equiv M_1 M_2$. Then, $[\![M[N/x]]\!]\vec{z} = ([\![M_1[N/x]]\!]\vec{z})([\![M_2[N/x]]\!]\vec{z})$
  $\overset{IH}{=} ([\![M_1]\!]\,\vec{z}([\![N]\!]\vec{z}))\,([\![M_2]\!]\,\vec{z}([\![N]\!]\vec{z})) = [\![M_1 M_2]\!]\,\vec{z}([\![N]\!]\vec{z})$.

- $M \equiv \lambda y.M'$. Then, $[\![M[N/x]]\!]\vec{z} = [\![\lambda y.M'[N/x]]\!]\vec{z} = [\![M'[N/x]]\!]\vec{z}$
  $\overset{IH}{=} [\![M']\!]\,\vec{z}\,([\![N]\!]\vec{z}) = [\![\lambda y.M']\!]\,\vec{z}\,([\![N]\!]\vec{z})$. □

Correctness

Theorem. *For any term* $\vdash M : A$*, if* $M \Downarrow V$ *then* $[\![M]\!] = [\![V]\!]$.

Proof. By induction on the derivation of $M \Downarrow V$.
The base case is obvious ($M \equiv V$).

For the inductive step, we have $M \equiv M_1 M_2$ and either of the following.

- M_1 is not a value. In this case, $M \Downarrow V$ ends in:

$$\frac{M_1 \Downarrow V' \quad V' M_2 \Downarrow V}{M_1 M_2 \Downarrow V}$$

  By the IH, $[\![M_1]\!] = [\![V']\!]$ and $[\![V' M_2]\!] = [\![V]\!]$.

  Thus, by compositionality, $[\![M_1 M_2]\!] = [\![V]\!]$.

Correctness (ctd)

Theorem. *For any term* $\vdash M : A$, *if* $M \Downarrow V$ *then* $[\![M]\!] = [\![V]\!]$.

Proof. By induction on the derivation of $M \Downarrow V$.
The base case is obvious ($M \equiv V$).

For the inductive step, we have $M \equiv M_1 M_2$ and either of the following.

- $M_1 \equiv \mathsf{c}$, some constant c. In this case, $M \Downarrow V$ ends in either:

$$\frac{M_2 \Downarrow U}{\mathsf{c}\, M_2 \Downarrow V}\ \mathsf{c} \not\equiv \mathbf{Y} \qquad \frac{M_2(\mathbf{Y} M_2) \Downarrow V}{\mathbf{Y}\, M_2 \Downarrow V}$$

  By the IH, $[\![M_2]\!] = [\![U]\!]$ or $[\![M_2(\mathbf{Y} M_2)]\!] = [\![V]\!]$.

  In the former case, $[\![\mathsf{c} M_2]\!] = [\![\mathsf{c}]\!][\![M_2]\!] = [\![\mathsf{c}]\!][\![U]\!]$ and this equals $[\![V]\!]$, by definition of $[\![\mathsf{c}]\!]$.

  In the latter, $[\![V]\!] = [\![M_2(\mathbf{Y} M_2)]\!] = [\![M_2]\!][\![\mathbf{Y} M_2]\!]$
  $= [\![M_2]\!](\mathtt{lfp}([\![M_2]\!])) = \mathtt{lfp}([\![M_2]\!]) = [\![\mathbf{Y} M_2]\!]$.

Correctness (ctd)

Theorem. *For any term* $\vdash M : A$*, if* $M \Downarrow V$ *then* $[\![M]\!] = [\![V]\!]$.

Proof. By induction on the derivation of $M \Downarrow V$.
The base case is obvious ($M \equiv V$).

For the inductive step, we have $M \equiv M_1 M_2$ and either of the following.

- $M_1 \equiv \lambda y.M'$. In this case, $M \Downarrow V$ ends in:

$$\frac{M'[M_2/y] \Downarrow V}{(\lambda y.M')M_2 \Downarrow V}$$

  By the IH, $[\![M'[M_2/y]]\!] = [\![V]\!]$.

  By the Substitution Lemma, $[\![M'[M_2/y]]\!] = [\![(\lambda y.M')M_2]\!]$. □

Soundness

Theorem. *$\mathcal{M}$ is a sound model of PCF, i.e. $[\![M]\!] = [\![N]\!] \implies M \cong N$ for all terms M, N.*

To prove the above we need another result, called **Computational Adequacy**:

Theorem. *For all terms $\vdash M : \mathtt{nat}$, if $[\![M]\!] = [\![V]\!]$ then $M \Downarrow V$.*

Proof of Soundness. Suppose $[\![M]\!] = [\![N]\!]$ and let $C[M] \Downarrow 0$, for some context $C[X]$.

By correctness, we have $[\![C[M]]\!] = [\![0]\!]$.

By the fact that the semantic translation is defined compositionally, and $[\![M]\!] = [\![N]\!]$, we have $[\![C[M]]\!] = [\![C[N]]\!] = [\![0]\!]$.

Thus, by computational adequacy, $C[N] \Downarrow 0$.

This shows $M \precsim N$, and symmetrically we show $N \precsim M$. $\square$

Examples

Here are some example (in)equivalences.

1. $\texttt{cond}\ x\ \texttt{t}\ \texttt{f}\ \cong?\ x$
2. $\texttt{cond}\ x\ \texttt{t}\ \texttt{t}\ \cong?\ \texttt{t}$
3. $\texttt{cond}\ x\ x\ \texttt{f}\ \cong?\ x$
4. $\texttt{cond}\ x\ x\ x\ \cong?\ x$

- $(\lambda x.\, \texttt{cond}\ (\texttt{zero}?\ x)\ 24\ 42)\, 42\ \cong\ 42$
- $\Omega_{\texttt{nat}\to\texttt{nat}}\ \cong\ \lambda x^{\texttt{nat}}.\Omega_{\texttt{nat}}$
- $x\ \ncong\ y$
- $\lambda x.\lambda y.\, \texttt{plus}\, x\, y\ \cong\ \lambda x.\lambda y.\, \texttt{plus}\, y\, x$

Computability predicates

We want to prove that, for all $\vdash M : \mathtt{nat}$, if $[\![M]\!] = [\![V]\!]$ then $M \Downarrow V$.

For that, we will need to establish something stronger, namely that every term is **computable**. We define the following **computability** predicate.

- A term $\vdash M : A_b$, where $A_b \in \{\mathtt{bool}, \mathtt{nat}\}$, is computable if $[\![M]\!] = [\![V]\!]$ implies $M \Downarrow V$ for all values V.
- A term $\vdash M : A_1 \to A_2$ is computable if $\vdash MN : A_2$ is computable for all computable $\vdash N : A_1$.
- A term $\{x_1 : A_1, \ldots, x_m : A_m\} \vdash M : A$ is computable if $\vdash M[N_1/x_1, \ldots, N_m/x_m] : A$ is computable for all computable $\vdash N_i : A_i$.

Thus, a term $\{x_1 : B_1, \ldots, x_m : B_m\} \vdash M : A_1 \to \cdots \to A_n \to A_b$ is computable iff $\vdash M[K_1/x_1, \ldots, K_m/x_m]\, N_1 \cdots N_n : A_b$ is computable for all computable $\vdash K_j : B_j$ and $\vdash N_i : A_i$.

Proof of computability

Proposition. *Every PCF term* $\Gamma \vdash M : A$ *is computable.*

Proof. Suppose $\Gamma = \{x_1 : A_1, \cdots, x_n : A_n\}$. We do induction on M.

- $M \equiv x_i$. From the definition: x_i is computable if $x_i[\vec{N}/\vec{x}] \equiv N_i$ is computable for all computable $\vdash N_1 : A_1, \ldots, N_n : A_n$.

- $M \equiv \mathsf{c} \not\equiv \mathbf{Y}$. If A is base type then we just need to show that $M \Downarrow \mathsf{c}$, which is an axiom.

  Otherwise, say $M \equiv \mathtt{cond}_{A_b}$ and take any computable $\vdash N : \mathtt{bool}$ and $\vdash N_i : A_b$, $i = 1, 2$, and assume that $[\![\mathtt{cond}\ N\ N_1\ N_2]\!] = [\![V]\!]$.

  By definition of $[\![\_]\!]$, either $[\![N]\!] = \mathtt{tt}$ and $[\![V]\!] = [\![N_1]\!]$, or $[\![N]\!] = \mathtt{ff}$ and $[\![V]\!] = [\![N_2]\!]$ – assume WLOG the former.

  Since N, N_1 are computable, $N \Downarrow \mathtt{t}$ and $N_1 \Downarrow V$, and thus $\mathtt{cond}\ N\ N_1\ N_2 \Downarrow V$.

  The other cases for c are treated similarly.

Proof of computability (ctd)

Proposition. *Every PCF term* $\Gamma \vdash M : A$ *is computable.*

Proof. Suppose $\Gamma = \{x_1 : A_1, \cdots, x_n : A_n\}$. We do induction on M.

- $M \equiv M_1 M_2$. We want to show that $M[\vec{K}/\vec{x}] \equiv M_1[\vec{K}/\vec{x}]M_2[\vec{K}/\vec{x}]$ is computable for all computable $\vec{K}$.

  By IH, M_i is computable, thus $M_i[\vec{K}/\vec{x}]$ is computable, for $i = 1, 2$. But computablility of $M_1[\vec{K}/\vec{x}]$ implies that of $M_1[\vec{K}/\vec{x}]M_2[\vec{K}/\vec{x}]$.

- $M \equiv \lambda y.M'$. We want to show that $\vdash ((\lambda y.M')[\vec{K}/\vec{x}])N_0\vec{N} : A_b$ is computable for all computable $\vec{K}, N_0, \vec{N}$.

  The term can be written as $M'' \equiv (\lambda y.M'[\vec{K}/\vec{x}])N_0\vec{N} : A_b$, and let us assume $[\![M'']\!] = [\![V]\!]$.

  We have that $[\![M'']\!] = [\![(M'[\vec{K}/\vec{x}, N_0/y])\vec{N}]\!]$ (by Substitution).

  Thus, by the IH on M', $(M'[\vec{K}/\vec{x}, N_0/y])\vec{N} \Downarrow V$, which implies $M'' \Downarrow V$.

Proof of computability (ctd)

- $M \equiv \mathbf{Y}_B$. We need to argue via *approximants*. We introduce auxiary PCF terms $\mathbf{Y}_{B,l}$, for each $l \in \mathbb{N}$ and each type B, and rules:

$$\frac{\Gamma \vdash N : B \to B}{\Gamma \vdash \mathbf{Y}_{B,l}\, N : B} \qquad \frac{N(\mathbf{Y}_l N) \Downarrow V}{\mathbf{Y}_{l+1} N \Downarrow V}$$

We also set $[\![\mathbf{Y}_{B,l}]\!]\vec{z}\, f = f^l(\bot_{[\![B]\!]})$. We claim the following (Exercise):

1. For any $\vdash N : B$, $[\![\mathbf{Y} N]\!] = \bigsqcup_{i\in\mathbb{N}} [\![\mathbf{Y}_i N]\!]$.
2. If some $\mathbf{Y}_l\, N_0 \vec{N} \Downarrow V$ then $\mathbf{Y}\, N_0 \vec{N} \Downarrow V$.

Now let $[\![\mathbf{Y}\, N_0 \vec{N}]\!] = [\![V]\!]$ for computable terms $N_0, \vec{N}$. By 1:

$$[\![V]\!] = [\![\mathbf{Y}\, N_0 \vec{N}]\!] = [\![\mathbf{Y}\, N_0]\!]([\![\vec{N}]\!]) = (\bigsqcup_{i\in\mathbb{N}} [\![\mathbf{Y}_i\, N_0]\!])([\![\vec{N}]\!])$$

And, by definition of lub's for chains of cts functions,

$$(\bigsqcup_{i\in\mathbb{N}} [\![\mathbf{Y}_i\, N_0]\!])([\![\vec{N}]\!]) = \bigsqcup_{i\in\mathbb{N}} [\![\mathbf{Y}_i\, N_0]\!]([\![\vec{N}]\!]) = \bigsqcup_{i\in\mathbb{N}} [\![\mathbf{Y}_i\, N_0 \vec{N}]\!]$$

Proof of computability (ctd)

- $M \equiv \mathbf{Y}_B$. We need to argue via *approximants*. We introduce auxiary PCF terms $\mathbf{Y}_{B,l}$, for each $l \in \mathbb{N}$ and each type B, and rules:

$$\frac{\Gamma \vdash N : B \to B}{\Gamma \vdash \mathbf{Y}_{B,l}\, N : B} \qquad \frac{N(\mathbf{Y}_l N) \Downarrow V}{\mathbf{Y}_{l+1} N \Downarrow V}$$

We also set $[\![\mathbf{Y}_{B,l}]\!]\vec{z}\, f = f^l(\bot_{[\![B]\!]})$. We claim the following (Exercise):

1. For any $\vdash N : B$, $[\![\mathbf{Y} N]\!] = \bigsqcup_{i\in\mathbb{N}} [\![\mathbf{Y}_i N]\!]$.
2. If some $\mathbf{Y}_l\, N_0 \vec{N} \Downarrow V$ then $\mathbf{Y}\, N_0 \vec{N} \Downarrow V$.

Now let $[\![\mathbf{Y}\, N_0 \vec{N}]\!] = [\![V]\!]$ for computable terms $N_0, \vec{N}$. We obtain:

$$[\![V]\!] = \bigsqcup_{i\in\mathbb{N}} [\![\mathbf{Y}_i\, N_0 \vec{N}]\!]$$

Now, $\vdash V : A_b$ so $[\![V]\!] \in [\![A_b]\!]$, where the latter is a flat cpo.

Thus, it must be $[\![\mathbf{Y}_l\, N_0 \vec{N}]\!] = [\![V]\!]$ for some l.
Hence, by 2, if $\mathbf{Y}_l$ is computable then we are done.

Proof of computability (ctd)

We need to argue via *approximants*. We introduce auxiary PCF terms $\mathbf{Y}_{B,l}$, for each $l \in \mathbb{N}$ and each type B, and rules:

$$\frac{\Gamma \vdash N : B \to B}{\Gamma \vdash \mathbf{Y}_{B,l}\, N : B} \qquad \frac{N(\mathbf{Y}_l N) \Downarrow V}{\mathbf{Y}_{l+1} N \Downarrow V}$$

We also set $[\![\mathbf{Y}_{B,l}]\!]\vec{z}\, f = f^l(\bot_{[\![B]\!]})$.

- $M \equiv \mathbf{Y}_{B,l}$. We do induction on l. The base case is trivial as $[\![\mathbf{Y}_0]\!] = \bot$ and therefore $[\![\mathbf{Y}_0\, N_0 \vec{N}]\!] = \bot \neq [\![V]\!]$, for all values V.

  For the inductive step, suppose $[\![\mathbf{Y}_{l+1}\, N_0 \vec{N}]\!] = [\![V]\!]$

  We then have $[\![\mathbf{Y}_{l+1}\, N_0 \vec{N}]\!] = [\![(N_0(\mathbf{Y}_l\, N_0))\vec{N}]\!] = [\![V]\!]$.

  By IH for l and the fact that $N_0, \vec{N}$ are computable, we have that $(N_0(\mathbf{Y}_l\, N_0))\vec{N}$ is computable and hence evaluates to V.

  Thus, $\mathbf{Y}_{l+1}\, N_0 \vec{N} \Downarrow V$. □

Soundness OK, completeness?

We have therefore proved the following for our cpo model $\mathcal{M}$.

Theorem. *$\mathcal{M}$ is a sound model of PCF, i.e. $[\![M]\!] = [\![N]\!] \implies M \cong N$ for all terms M, N.*

What about the converse?

- Does our model verify all PCF-equivalences?
- Or are there terms M_1, M_2 such that $M_1 \cong M_2$ but $[\![M_1]\!] \neq [\![M_2]\!]$?

Actually, the latter is the case...

- The semantic model makes *too many* distinctions between terms.
- The reason is that it is *too expressive*, i.e. it contains elements which are not the denotations of PCF terms.

Parallel OR

Consider the function $por : \mathbb{B}_\bot \to \mathbb{B}_\bot \to \mathbb{B}_\bot$ defined by:

inputs	`tt`	`ff`	$\bot$
`tt`	`tt`	`tt`	`tt`
`ff`	`tt`	`ff`	$\bot$
$\bot$	`tt`	$\bot$	$\bot$

Note that por is continuous:

- it is monotone in each argument,
- and the domain is finite in both arguments.

and thus it is an element of $[\![\texttt{bool} \to \texttt{bool} \to \texttt{bool}]\!]$.

However, there is no PCF term $\vdash M : \texttt{bool} \to \texttt{bool} \to \texttt{bool}$ such that:

$$[\![M]\!] = por$$

But, I thought PCF was Turing complete...

Consider the function $por : \mathbb{B}_\perp \to \mathbb{B}_\perp \to \mathbb{B}_\perp$ defined by:

inputs	`tt`	`ff`	$\perp$
`tt`	`tt`	`tt`	`tt`
`ff`	`tt`	`ff`	$\perp$
$\perp$	`tt`	$\perp$	$\perp$

Note that por is continuous: ...

However, there is no PCF term $\vdash M : \texttt{bool} \to \texttt{bool} \to \texttt{bool}$ such that:

$$[\![M]\!] = por$$

- The function por is not recursive, in the sense we saw before. In particular, it is defined on a lifted domain (containing $\perp$).

Why not definable?

Consider the function $por : \mathbb{B}_\perp \to \mathbb{B}_\perp \to \mathbb{B}_\perp$ defined by:

inputs	tt	ff	$\perp$
tt	tt	tt	tt
ff	tt	ff	$\perp$
$\perp$	tt	$\perp$	$\perp$

A PCF term corresponding to por,

- would need to be non-constant,
- so it would need to examine its inputs.

Well, one input would need to be examined first – there is no parallelism!

Multi-holed contexts

Contexts can be straightforwardly extended to many "holes", into which other terms can be plugged. E.g.

$$C[X, Y] \equiv \lambda x.xXY$$

Here, X, Y are both hole variables.

For terms M, N, we write $C[M, N]$ to mean the result of copying M for X, and N for Y, in a "blind" (not capture-avoiding manner) in $C[X, Y]$. E.g:

$$\begin{aligned} C[z, z] &\equiv \lambda x.xzz \\ C[\lambda z.z, y] &\equiv \lambda x.x(\lambda z.z)y \\ C[\mathtt{pred}, \Omega] &\equiv \lambda x.x(\mathtt{pred})\Omega \\ C[xyy, y] &\equiv \lambda x.x(xyy)y \end{aligned}$$

Undefinability of por

Lemma. *Let $C[X_1, X_2]$ be a context and M_1, M_2 be closed terms such that $\vdash C[M_1, M_2] : A$, and suppose that $C[M_1, M_2] \Downarrow V$. Then,*

- *either $V \equiv C'[M_1, M_2]$ and for all closed terms N_1, N_2 such that $\vdash C[N_1, N_2] : A$, we have $C[N_1, N_2] \Downarrow C'[N_1, N_2]$,*
- *or there is $i \in \{1, 2\}$ such that, for all closed terms N_1, N_2 such that $\vdash C[N_1, N_2] : A$, if N_i is not a value and $C[N_1, N_2] \Downarrow V'$ for some value V' then the derivation of the latter contains a node $N_i \Downarrow V_i$.*

Proposition. *For no term $\vdash M : \mathtt{bool} \to \mathtt{bool} \to \mathtt{bool}$, $[\![M]\!] = por$.*

Proof. If such a term M existed then, by correctness and adequacy,

$$M\,\mathtt{t}\,\Omega \Downarrow \mathtt{t} \qquad M\,\mathtt{f}\,\mathtt{f} \Downarrow \mathtt{f} \qquad M\,\Omega\,\mathtt{t} \Downarrow \mathtt{t}$$

But now apply the lemma to $M\,\mathtt{t}\,\Omega \Downarrow \mathtt{t}$, and given that $M\,\mathtt{f}\,\mathtt{f} \Downarrow \mathtt{f}$, to get an $i \in \{1, 2\}$ such that the second part of the lemma holds.

For $i = 1$, $M\,\Omega\,\mathtt{t} \Downarrow \mathtt{t}$ breaks the argument; for $i = 2$, $M\,\mathtt{t}\,\Omega \Downarrow \mathtt{t}$. □

Failure of completeness

We exploit undefinability of por to prove non-completeness of $\mathcal{M}$. Consider the following term.

$$\begin{aligned}\texttt{test\_por} \equiv \lambda f^{\texttt{bool}\to\texttt{bool}\to\texttt{bool}}.\texttt{cond}\ (f\,\texttt{f}\,\texttt{f})\ \Omega \\ (\ \texttt{cond}\ (f\,\texttt{t}\,\Omega) \\ (\texttt{cond}\ (f\,\Omega\,\texttt{t})\ \texttt{t}\ \Omega)\ \Omega\ )\end{aligned}$$

Lemma. $\texttt{test\_por} \cong \lambda f.\Omega.$

Proof. By the context lemma, it suffices to check that, for all terms $\vdash M : \texttt{bool} \to \texttt{bool} \to \texttt{bool}$ and values V,

$$\texttt{test\_por}\, M \Downarrow V \iff (\lambda f.\Omega) M \Downarrow V$$

i.e. that $\texttt{test\_por}\, M \Uparrow$ for all terms M.

But $\texttt{test\_por}\, M \Downarrow$ iff $M\,\texttt{t}\,\Omega \Downarrow \texttt{t}$, $M\,\texttt{f}\,\texttt{f} \Downarrow \texttt{f}$ and $M\,\Omega\,\texttt{t} \Downarrow \texttt{t}$, which is not possible, due to the (proof of the) previous result. □

Failure of completeness (ctd)

We exploit undefinability of por to prove non-completeness of $\mathcal{M}$. Consider the following term.

$$\begin{aligned}\mathtt{test\_por} \equiv \lambda f^{\mathtt{bool}\to\mathtt{bool}\to\mathtt{bool}}.\mathtt{cond}\ (f\,\mathtt{f}\,\mathtt{f})\ \Omega \\ (\ \mathtt{cond}\ (f\,\mathtt{t}\,\Omega) \\ (\mathtt{cond}\ (f\,\Omega\,\mathtt{t})\ \mathtt{t}\ \Omega)\ \Omega\ )\end{aligned}$$

Lemma. $\mathtt{test\_por} \cong \lambda f.\Omega.$

Theorem. *The cpo model $\mathcal{M}$ of PCF is not complete.*

Proof. It suffices to show that $[\![\mathtt{test\_por}]\!] \neq [\![\lambda f.\Omega]\!]$.

But this is now straightforward:

$$[\![\mathtt{test\_por}]\!]\, por = \mathtt{tt} \neq \bot = [\![\lambda f.\Omega]\!]\, por$$

□

Proof of the lemma

Lemma. *Let $C[X_1, X_2]$ be a context and M_1, M_2 be closed terms such that $\vdash C[M_1, M_2] : A$, and suppose that $C[M_1, M_2] \Downarrow V$. Then,*

- *either $V \equiv C'[M_1, M_2]$ and for all closed terms N_1, N_2 such that $\vdash C[N_1, N_2] : A$, we have $C[N_1, N_2] \Downarrow C'[N_1, N_2]$,*
- *or there is $i \in \{1, 2\}$ such that, for all closed terms N_1, N_2 such that $\vdash C[N_1, N_2] : A$, if N_i is not a value and $C[N_1, N_2] \Downarrow V'$ for some value V' then the derivation of the latter contains a node $N_i \Downarrow V_i$.*

Proof. Suppose that $C[M_1, M_2] \Downarrow_n V$. We do induction on n.

For the base case, either $C[X_1, X_2] \equiv V$, or $C[X_1, X_2] \equiv \lambda x.C''[X_1, X_2]$, or $C[X_1, X_2] \equiv X_i$ and $M_i \equiv V$ for some i. In the former two cases, the first part of the claim holds, in the latter the second (for the same i).

Proof of the lemma

Lemma. *Let $C[X_1, X_2]$ be a context and M_1, M_2 be closed terms such that $\vdash C[M_1, M_2] : A$, and suppose that $C[M_1, M_2] \Downarrow V$. Then,*

- *either $V \equiv C'[M_1, M_2]$ and for all closed terms N_1, N_2 such that $\vdash C[N_1, N_2] : A$, we have $C[N_1, N_2] \Downarrow C'[N_1, N_2]$,*
- *or there is $i \in \{1, 2\}$ such that, for all closed terms N_1, N_2 such that $\vdash C[N_1, N_2] : A$, if N_i is not a value and $C[N_1, N_2] \Downarrow V'$ for some value V' then the derivation of the latter contains a node $N_i \Downarrow V_i$.*

Proof. Suppose that $C[M_1, M_2] \Downarrow_n V$. We do induction on n.

For the inductive step, it must be that $C[X_1, X_2] \equiv C_1[X_1, X_2]C_2[X_1, X_2]$.

If $C_1[M_1, M_2]$ is a value then . . . (Exercise).

Finally, if $C_1[M_1, M_2]$ is not a value then $C_1[M_1, M_2] \Downarrow_m V''$ with $m < n$ so we can apply the IH. If the second part of the claim holds (for C_1), then we are done. Otherwise, $V'' \equiv C''[M_1, M_2]$ and $C_1[N_1, N_2] \Downarrow C''[N_1, N_2]$. Moreover, we have $C''[M_1, M_2]C_2[M_1, M_2] \Downarrow V$, for which we apply the IH and obtain the claim for the original $C[M_1, M_2]$. □

Exercises

1. Let D be a cpo and $(x_i)_{i \in \mathbb{N}}$ a chain in D. Show that, for any $k \in \mathbb{N}$, $\bigsqcup_{i \in \mathbb{N}} x_i = \bigsqcup_{i \geq k} x_i$.
2. Compute $[\![\texttt{plus}]\!]$ and $[\![\texttt{mult}]\!]$.
3. Complete the remaining part of the proof of computability.
4. Complete the remaining part of the proof of the last lemma (aka *the Activity Lemma*).