Denotational Semantics

Ioannis Markakis

Lectures for Part II CST 2026/2027

Recap of Part I

We have studied domains and continuous functions:

Recap of Part I

We have studied domains and continuous functions:

- We have seen Kleene's fixed point theorem

Recap of Part I

We have studied domains and continuous functions:

- We have seen Kleene's fixed point theorem
- We have seen how to construct domains (flat,product,function) and continuous functions between them

Recap of Part I

We have studied domains and continuous functions:

- We have seen Kleene's fixed point theorem
- We have seen how to construct domains (flat,product,function) and continuous functions between them
- We described Scott induciton as a method for reasoning about least fixed points

Recap of Part I

We have studied domains and continuous functions:

- We have seen Kleene's fixed point theorem
- We have seen how to construct domains (flat,product,function) and continuous functions between them
- We described Scott induciton as a method for reasoning about least fixed points

Recap of Part I

We have studied domains and continuous functions:

- We have seen Kleene's fixed point theorem
- We have seen how to construct domains (flat,product,function) and continuous functions between them
- We described Scott induciton as a method for reasoning about least fixed points

We will now apply these tools to study PCF

The language Pcf

The language Pcf

Syntax

Types and terms of Pcf

Types:

$$\tau ::= \texttt{nat} \mid \texttt{bool} \mid \tau \to \tau$$

Types and terms of Pcf

Types:

$$\tau ::= \mathtt{nat} \mid \mathtt{bool} \mid \tau \to \tau$$

Terms:

$$\begin{aligned} t \quad ::= \quad & \mathtt{0} \mid \mathtt{succ}\,(t) \mid \mathtt{pred}\,(t) \mid \\ & \mathtt{true} \mid \mathtt{false} \mid \mathtt{zero?}\,(t) \mid \mathtt{if}\ t\ \mathtt{then}\ t\ \mathtt{else}\ t \\ & x \mid \mathtt{fun}\, x{:}\tau.\, t \mid t\ t \mid \mathtt{fix}(t) \end{aligned}$$

Types and terms of Pcf

Types:

$$\tau ::= \texttt{nat} \mid \texttt{bool} \mid \tau \to \tau$$

Terms:

$$\begin{aligned} t \quad ::= \quad & \texttt{0} \mid \texttt{succ}\,(t) \mid \texttt{pred}\,(t) \mid \\ & \texttt{true} \mid \texttt{false} \mid \texttt{zero?}\,(t) \mid \texttt{if}\ t\ \texttt{then}\ t\ \texttt{else}\ t \\ & x \mid \texttt{fun}\, x{:}\tau.\, t \mid t\ t \mid \texttt{fix}(t) \end{aligned}$$

- λ-calculus + base types/functions + `fix`
- tiny ML (without references, ADTs, polymorphism…)

Variables, substitutions and contexts

The constructor $\mathtt{fun}\, x{:}\tau.\, t$ binds the variable x.

- We consider terms up to α-equivalence of bound variables

$$\mathtt{fun}\, x{:}\tau.\, x = \mathtt{fun}\, y{:}\tau.\, y$$

Variables, substitutions and contexts

The constructor $\mathtt{fun}\, x{:}\tau.\, t$ binds the variable x.

- We consider terms up to α-equivalence of bound variables

$$\mathtt{fun}\, x{:}\tau.\, x = \mathtt{fun}\, y{:}\tau.\, y$$

- We denote by $t[u/x]$ the (capture-avoiding) substitution of u for x in t

Variables, substitutions and contexts

The constructor $\mathtt{fun}\, x{:}\tau.\, t$ binds the variable x.

- We consider terms up to α-equivalence of bound variables

$$\mathtt{fun}\, x{:}\tau.\, x = \mathtt{fun}\, y{:}\tau.\, y$$

- We denote by $t[u/x]$ the (capture-avoiding) substitution of u for x in t

To assign types to terms, we define typing contexts by the following grammar (with x fresh in Γ):

$$\Gamma ::= \cdot \mid \Gamma, x{:}\tau$$

Variables, substitutions and contexts

The constructor $\mathtt{fun}\, x{:}\,\tau.\, t$ binds the variable x.

- We consider terms up to α-equivalence of bound variables

$$\mathtt{fun}\, x{:}\,\tau.\, x = \mathtt{fun}\, y{:}\,\tau.\, y$$

- We denote by $t[u/x]$ the (capture-avoiding) substitution of u for x in t

To assign types to terms, we define typing contexts by the following grammar (with x fresh in Γ):

$$\Gamma ::= \cdot \mid \Gamma, x{:}\,\tau$$

- finite partial functions from variable to types
- finite list of pairs $x_1{:}\,\tau_1, \ldots, x_n{:}\,\tau_n$ with x_i distinct variables

Typing for Pcf (I)

$\boxed{\Gamma \vdash t : \tau}$ The term t has type τ in context Γ

$$\textsc{Zero}\ \frac{}{\Gamma \vdash \mathtt{0} : \mathtt{nat}} \qquad \textsc{Succ}\ \frac{\Gamma \vdash t : \mathtt{nat}}{\Gamma \vdash \mathtt{succ}\,(t) : \mathtt{nat}} \qquad \textsc{Pred}\ \frac{\Gamma \vdash t : \mathtt{nat}}{\Gamma \vdash \mathtt{pred}\,(t) : \mathtt{nat}}$$

Typing for Pcf (I)

$\boxed{\Gamma \vdash t : \tau}$ The term t has type τ in context Γ

$$\textsc{Zero}\ \frac{}{\Gamma \vdash \mathtt{0} : \mathtt{nat}} \qquad \textsc{Succ}\ \frac{\Gamma \vdash t : \mathtt{nat}}{\Gamma \vdash \mathtt{succ}\,(t) : \mathtt{nat}} \qquad \textsc{Pred}\ \frac{\Gamma \vdash t : \mathtt{nat}}{\Gamma \vdash \mathtt{pred}\,(t) : \mathtt{nat}}$$

$$\textsc{True}\ \frac{}{\Gamma \vdash \mathtt{true} : \mathtt{bool}} \qquad \textsc{False}\ \frac{}{\Gamma \vdash \mathtt{false} : \mathtt{bool}} \qquad \textsc{IsZ}\ \frac{\Gamma \vdash t : \mathtt{nat}}{\Gamma \vdash \mathtt{zero?}\,(t) : \mathtt{bool}}$$

$$\textsc{If}\ \frac{\Gamma \vdash b : \mathtt{bool} \qquad \Gamma \vdash t : \tau \qquad \Gamma \vdash t' : \tau}{\Gamma \vdash \mathtt{if}\ b\ \mathtt{then}\ t\ \mathtt{else}\ t' : \tau}$$

Typing for Pcf (II)

$$\textsc{Var}\ \frac{\Gamma(x) = \tau}{\Gamma \vdash x : \tau} \qquad \textsc{Fun}\ \frac{\Gamma, x{:}\sigma \vdash t : \tau}{\Gamma \vdash \mathtt{fun}\, x{:}\sigma.\, t : \sigma \to \tau}$$

$$\textsc{App}\ \frac{\Gamma \vdash f : \sigma \to \tau \qquad \Gamma \vdash u : \sigma}{\Gamma \vdash f\, u : \tau} \qquad \textsc{Fix}\ \frac{\Gamma \vdash f : \tau \to \tau}{\Gamma \vdash \mathtt{fix}(f) : \tau}$$

Typing for Pcf (II)

$$\textsc{Var}\ \frac{\Gamma(x) = \tau}{\Gamma \vdash x : \tau} \qquad \textsc{Fun}\ \frac{\Gamma, x{:}\sigma \vdash t : \tau}{\Gamma \vdash \mathtt{fun}\, x{:}\sigma.\, t : \sigma \to \tau}$$

$$\textsc{App}\ \frac{\Gamma \vdash f : \sigma \to \tau \qquad \Gamma \vdash u : \sigma}{\Gamma \vdash f\, u : \tau} \qquad \textsc{Fix}\ \frac{\Gamma \vdash f : \tau \to \tau}{\Gamma \vdash \mathtt{fix}(f) : \tau}$$

Notation:

$$\textsc{Pcf}_{\Gamma,\tau} \stackrel{\text{def}}{=} \{\, t \mid \Gamma \vdash t : \tau \,\}$$

$$\textsc{Pcf}_{\tau} \stackrel{\text{def}}{=} \textsc{Pcf}_{\cdot,\tau}$$

Typing for Pcf (II)

$$\textsc{Var}\ \frac{\Gamma(x) = \tau}{\Gamma \vdash x : \tau} \qquad \textsc{Fun}\ \frac{\Gamma, x{:}\sigma \vdash t : \tau}{\Gamma \vdash \mathtt{fun}\, x{:}\sigma.\, t : \sigma \to \tau}$$

$$\textsc{App}\ \frac{\Gamma \vdash f : \sigma \to \tau \qquad \Gamma \vdash u : \sigma}{\Gamma \vdash f\, u : \tau} \qquad \textsc{Fix}\ \frac{\Gamma \vdash f : \tau \to \tau}{\Gamma \vdash \mathtt{fix}(f) : \tau}$$

Notation:

$$\textsc{Pcf}_{\Gamma,\tau} \stackrel{\text{def}}{=} \{\, t \mid \Gamma \vdash t : \tau \,\}$$

$$\textsc{Pcf}_{\tau} \stackrel{\text{def}}{=} \textsc{Pcf}_{\cdot,\tau}$$

We **only** care about well-typed programs!

We will **not** give semantics for other ones.

Lemma

If $\Gamma \vdash t : \tau$ and $\Gamma, x : \tau \vdash t' : \tau'$ both hold, then so does $\Gamma \vdash t'[t/x] : \tau'$.

Lemma

If $\Gamma \vdash t : \tau$ and $\Gamma, x : \tau \vdash t' : \tau'$ both hold, then so does $\Gamma \vdash t'[t/x] : \tau'$.

- The proof is by induction on the typing derivations.

The language Pcf

Operational semantics

PCF VALUES

We give operational semantics for PCF in terms of an evaluation relation between closed terms and values(canonical forms):

PCF VALUES

We give operational semantics for PCF in terms of an evaluation relation between closed terms and values(canonical forms):

Values:

$$v ::= \underbrace{\mathtt{0} \mid \mathtt{succ}\,(v)}_{\underline{n}} \mid \mathtt{true} \mid \mathtt{false} \mid \underbrace{\mathtt{fun}\, x{:}\tau.\, t}_{\text{All functions}}$$

Pcf values

We give operational semantics for Pcf in terms of an evaluation relation between closed terms and values(canonical forms):

Values:

$$v ::= \underbrace{\mathtt{0} \mid \mathtt{succ}\,(v)}_{\underline{n}} \mid \mathtt{true} \mid \mathtt{false} \mid \underbrace{\mathtt{fun}\, x{:}\tau.\, t}_{\text{All functions}}$$

Evaluation is a relation $t \Downarrow_\tau v$ defined inductively.

PCF EVALUATION

$$\textsc{Val}\ \frac{\vdash v : \tau}{v \Downarrow_\tau v}$$

PCF evaluation

$$\textsc{Val}\ \frac{\vdash v : \tau}{v \Downarrow_\tau v} \qquad \textsc{Succ}\ \frac{t \Downarrow_{\mathsf{nat}} v}{\mathtt{succ}\,(t) \Downarrow_{\mathsf{nat}} \mathtt{succ}\,(v)} \qquad \textsc{Pred}\ \frac{t \Downarrow_{\mathsf{nat}} \mathtt{succ}\,(v)}{\mathtt{pred}\,(t) \Downarrow_{\mathsf{nat}} v}$$

PCF evaluation

$$\textsc{Val}\ \frac{\vdash v : \tau}{v \Downarrow_\tau v} \qquad \textsc{Succ}\ \frac{t \Downarrow_{\mathtt{nat}} v}{\mathtt{succ}\,(t) \Downarrow_{\mathtt{nat}} \mathtt{succ}\,(v)} \qquad \textsc{Pred}\ \frac{t \Downarrow_{\mathtt{nat}} \mathtt{succ}\,(v)}{\mathtt{pred}\,(t) \Downarrow_{\mathtt{nat}} v}$$

$$\textsc{ZeroZ}\ \frac{t \Downarrow_{\mathtt{nat}} \mathtt{0}}{\mathtt{zero?}\,(t) \Downarrow_{\mathtt{bool}} \mathtt{true}} \qquad \textsc{ZeroS}\ \frac{t \Downarrow_{\mathtt{nat}} \mathtt{succ}\,(v)}{\mathtt{zero?}\,(t) \Downarrow_{\mathtt{bool}} \mathtt{false}}$$

PCF evaluation

$$\textsc{Val}\ \frac{\vdash v : \tau}{v \Downarrow_\tau v} \qquad \textsc{Succ}\ \frac{t \Downarrow_{\mathtt{nat}} v}{\mathtt{succ}\,(t) \Downarrow_{\mathtt{nat}} \mathtt{succ}\,(v)} \qquad \textsc{Pred}\ \frac{t \Downarrow_{\mathtt{nat}} \mathtt{succ}\,(v)}{\mathtt{pred}\,(t) \Downarrow_{\mathtt{nat}} v}$$

$$\textsc{ZeroZ}\ \frac{t \Downarrow_{\mathtt{nat}} \mathtt{0}}{\mathtt{zero?}\,(t) \Downarrow_{\mathtt{bool}} \mathtt{true}} \qquad \textsc{ZeroS}\ \frac{t \Downarrow_{\mathtt{nat}} \mathtt{succ}\,(v)}{\mathtt{zero?}\,(t) \Downarrow_{\mathtt{bool}} \mathtt{false}}$$

$$\textsc{IfT}\ \frac{b \Downarrow_{\mathtt{bool}} \mathtt{true} \qquad t_1 \Downarrow_\tau v}{\mathtt{if}\ b\ \mathtt{then}\ t_1\ \mathtt{else}\ t_2 \Downarrow_\tau v} \qquad \textsc{IfF}\ \frac{b \Downarrow_{\mathtt{bool}} \mathtt{false} \qquad t_2 \Downarrow_\tau v}{\mathtt{if}\ b\ \mathtt{then}\ t_1\ \mathtt{else}\ t_2 \Downarrow_\tau v}$$

PCF evaluation

$$\textsc{Val}\ \frac{\vdash v : \tau}{v \Downarrow_\tau v} \qquad \textsc{Succ}\ \frac{t \Downarrow_{\mathtt{nat}} v}{\mathtt{succ}\,(t) \Downarrow_{\mathtt{nat}} \mathtt{succ}\,(v)} \qquad \textsc{Pred}\ \frac{t \Downarrow_{\mathtt{nat}} \mathtt{succ}\,(v)}{\mathtt{pred}\,(t) \Downarrow_{\mathtt{nat}} v}$$

$$\textsc{ZeroZ}\ \frac{t \Downarrow_{\mathtt{nat}} \mathtt{0}}{\mathtt{zero?}\,(t) \Downarrow_{\mathtt{bool}} \mathtt{true}} \qquad \textsc{ZeroS}\ \frac{t \Downarrow_{\mathtt{nat}} \mathtt{succ}\,(v)}{\mathtt{zero?}\,(t) \Downarrow_{\mathtt{bool}} \mathtt{false}}$$

$$\textsc{IfT}\ \frac{b \Downarrow_{\mathtt{bool}} \mathtt{true} \qquad t_1 \Downarrow_\tau v}{\mathtt{if}\ b\ \mathtt{then}\ t_1\ \mathtt{else}\ t_2 \Downarrow_\tau v} \qquad \textsc{IfF}\ \frac{b \Downarrow_{\mathtt{bool}} \mathtt{false} \qquad t_2 \Downarrow_\tau v}{\mathtt{if}\ b\ \mathtt{then}\ t_1\ \mathtt{else}\ t_2 \Downarrow_\tau v}$$

$$\textsc{Fun}\ \frac{t \Downarrow_{\sigma\to\tau} \mathtt{fun}\,x{:}\sigma.\,t' \qquad t'[u/x] \Downarrow_\tau v}{t\,u \Downarrow_\tau v} \qquad \textsc{Fix}\ \frac{t\,(\mathtt{fix}(t)) \Downarrow_\tau v}{\mathtt{fix}(t) \Downarrow_\tau v}$$

Example: Addition

Addition can be defined by recursion on the second argument.
It must satisfy the fixed point equation for $x, y : \mathtt{nat}$:

$$\mathtt{plus}\ x\ y = \mathtt{if}\ \mathtt{zero?}(y)\ \mathtt{then}\ x\ \mathtt{else}\ \mathtt{succ}(\mathtt{plus}\ x\ \mathtt{pred}(y))$$

Example: Addition

Addition can be defined by recursion on the second argument.
It must satisfy the fixed point equation for $x, y : \mathtt{nat}$:

$$\mathtt{plus}\ x\ y = \mathtt{if}\ \mathtt{zero?}\,(y)\ \mathtt{then}\ x\ \mathtt{else}\ \mathtt{succ}\,(\mathtt{plus}\ x\ \mathtt{pred}\,(y))$$

We may then define in context $x : \mathtt{nat}$:

$$\begin{aligned}\mathtt{plus}_x \stackrel{\text{def}}{=}\ & \mathtt{fix}(\mathtt{fun}(p{:}\,\mathtt{nat} \to \mathtt{nat})(y{:}\,\mathtt{nat}).\\ & \quad \mathtt{if}\ \mathtt{zero?}\,(y)\ \mathtt{then}\ x\ \mathtt{else}\ \mathtt{succ}\,(p\ \mathtt{pred}\,(y)))\end{aligned}$$

and then bind the variable x to get:

$$\mathtt{plus} = \mathtt{fun}\,x{:}\,\mathtt{nat}.\,\mathtt{plus}_x \tag{1}$$

Evaluation (I)

$$\textsc{Fun}\ \frac{\texttt{plus} \Downarrow \texttt{fun}\,\texttt{x}\!:\!\texttt{nat}.\,\texttt{plus}_{\texttt{x}} \qquad \texttt{plus}_{\underline{3}}\ \underline{1} \Downarrow \underline{4}}{\texttt{plus}\ \underline{3}\ \underline{1} \Downarrow_{\texttt{nat}} \underline{4}}$$

Evaluation (I)

$$\textsc{Fun}\ \frac{\mathtt{plus} \Downarrow \mathtt{fun}\,\mathtt{x}\colon \mathtt{nat}.\,\mathtt{plus}_{\mathtt{x}} \qquad \mathtt{plus}_{\underline{3}}\ \underline{1} \Downarrow \underline{4}}{\mathtt{plus}\ \underline{3}\ \underline{1} \Downarrow_{\mathtt{nat}} \underline{4}}$$

$$\textsc{Fix}\ \frac{\textsc{Fun}\ \dfrac{\textsc{Val}\ \dfrac{}{(\mathtt{fun}\ p\colon \mathtt{nat} \to \mathtt{nat}.\ \ldots) \Downarrow \ldots} \qquad \textsc{Val}\ \dfrac{}{(\mathtt{fun}\ y\colon \mathtt{nat}.\ldots)[\mathtt{plus}_x/p] \Downarrow r_x}}{(\mathtt{fun}(p\colon \mathtt{nat} \to \mathtt{nat})(y\colon \mathtt{nat}).\ \ldots)\ \mathtt{plus}_x \Downarrow r_x}}{\mathtt{plus}_x \Downarrow \underbrace{\mathtt{fun}\,y\colon \mathtt{nat}.\,\mathtt{if}\ \mathtt{zero?}\,(y)\ \mathtt{then}\ x\ \mathtt{else}\ \mathtt{succ}\,(\mathtt{plus}_x\ \mathtt{pred}\,(y))}_{r_x}}$$

Evaluation (II)

$$\textsc{Fun}\;\frac{\texttt{plus}_{\underline{3}} \Downarrow r_{\underline{3}} \qquad \textsc{Iff}\;\dfrac{\textsc{ZeroS}\;\dfrac{\textsc{Val}\;\dfrac{}{\underline{1} \Downarrow \underline{1}}}{\texttt{zero?}\,(\underline{1}) \Downarrow \texttt{false}} \qquad \textsc{Succ}\;\dfrac{\dfrac{\textsc{ZeroZ}\;\dfrac{\textsc{Pred}\;\dfrac{\ldots}{\texttt{pred}\,(\underline{1}) \Downarrow \texttt{0}}}{\texttt{zero?}\,(\texttt{pred}\,(\underline{1})) \Downarrow \texttt{true}} \qquad \ldots}{\texttt{plus}_{\underline{3}}\ \texttt{pred}\,(\underline{1}) \Downarrow \underline{3}}}{\texttt{succ}\,(\texttt{plus}_{\underline{3}}\ \texttt{pred}\,(\underline{1})) \Downarrow \underline{4}}}{\texttt{if zero?}\,(\underline{1})\ \texttt{then}\ \underline{3}\ \texttt{else succ}\,(\texttt{plus}_{\underline{3}}\ \texttt{pred}\,(\underline{1})) \Downarrow \underline{4}}}{\texttt{plus}_{\underline{3}}\ \underline{1} \Downarrow_{\texttt{nat}} \underline{4}}$$

Divergence

Divergence ($t \Uparrow_\tau$):

$$t : \tau \quad \wedge \quad \nexists v.\, t \Downarrow_\tau v$$

Divergence

Divergence ($t \Uparrow_\tau$):

$$t : \tau \quad \wedge \quad \nexists v.\, t \Downarrow_\tau v$$

$$\Omega_\tau \stackrel{\text{def}}{=} \texttt{fix}(\texttt{fun}\, x{:}\tau.\, x)$$

$$\Omega_\tau \Uparrow_\tau \qquad \text{(diverges)}$$

Divergence

Divergence ($t \Uparrow_\tau$):

$$t : \tau \quad \wedge \quad \nexists v.\, t \Downarrow_\tau v$$

$$\Omega_\tau \overset{\text{def}}{=} \texttt{fix}(\texttt{fun}\, x{:}\tau.\, x)$$

$$\Omega_\tau \Uparrow_\tau \qquad \text{(diverges)}$$

$$\dfrac{\dfrac{\texttt{fun}\, x{:}\tau.\, x \Downarrow \texttt{fun}\, x{:}\tau.\, x \qquad \overset{\mathcal{P}}{\texttt{fix}(\texttt{fun}\, x{:}\tau.\, x) \Downarrow v}}{(\texttt{fun}\, x{:}\tau.\, x)\,(\texttt{fix}(\texttt{fun}\, x{:}\tau.\, x)) \Downarrow v}}{\texttt{fix}(\texttt{fun}\, x{:}\tau.\, x) \Downarrow v}$$

Call-by-name and call-by-value

$$\textsc{Fun-CBN}\ \frac{t \Downarrow_{\sigma\to\tau} \mathsf{fun}\, x{:}\sigma.\, t' \qquad t'[u/x] \Downarrow_\tau v}{t\ u \Downarrow_\tau v}$$

$$\textsc{Fun-CBV}\ \frac{t \Downarrow_{\sigma\to\tau} \mathsf{fun}\, x{:}\sigma.\, t' \qquad u \Downarrow_\sigma v' \qquad t'[v'/x] \Downarrow_\tau v}{t\ u \Downarrow_\tau v}$$

Call-by-name and call-by-value

$$\textsc{Fun-CBN}\ \frac{t \Downarrow_{\sigma\to\tau} \mathtt{fun}\, x{:}\sigma.\, t' \qquad t'[u/x] \Downarrow_\tau v}{t\ u \Downarrow_\tau v}$$

$$\textsc{Fun-CBV}\ \frac{t \Downarrow_{\sigma\to\tau} \mathtt{fun}\, x{:}\sigma.\, t' \qquad u \Downarrow_\sigma v' \qquad t'[v'/x] \Downarrow_\tau v}{t\ u \Downarrow_\tau v}$$

What does $(\mathtt{fun}\, x{:}\mathtt{nat}.\ \mathtt{0})\ \Omega_{\mathtt{nat}}$ denote?

Call-by-name and call-by-value

$$\textsc{Fun-CBN}\ \frac{t \Downarrow_{\sigma\to\tau} \mathtt{fun}\, x{:}\sigma.\, t' \qquad t'[u/x] \Downarrow_\tau v}{t\ u \Downarrow_\tau v}$$

$$\textsc{Fun-CBV}\ \frac{t \Downarrow_{\sigma\to\tau} \mathtt{fun}\, x{:}\sigma.\, t' \qquad u \Downarrow_\sigma v' \qquad t'[v'/x] \Downarrow_\tau v}{t\ u \Downarrow_\tau v}$$

What does $(\mathtt{fun}\, x{:}\mathtt{nat}.\ \mathtt{0})\ \Omega_{\mathtt{nat}}$ denote?

In call-by-value, all functions are strict… but the least-fixed points of a strict function is always $\bot$!

Small-step semantic

Small-step $t \rightsquigarrow_\tau u$:

$$\frac{}{(\mathsf{fun}\, x{:}\sigma.\, t)\; u \rightsquigarrow_\tau t[u/x]} \qquad \frac{t \rightsquigarrow_{\sigma\to\tau} t'}{t\; u \rightsquigarrow_\tau t'\; u} \qquad \ldots$$

Small-step semantic

Small-step $t \rightsquigarrow_\tau u$:

$$\frac{}{(\mathsf{fun}\, x{:}\sigma.\, t)\, u \rightsquigarrow_\tau t[u/x]} \qquad \frac{t \rightsquigarrow_{\sigma\to\tau} t'}{t\, u \rightsquigarrow_\tau t'\, u} \qquad \ldots$$

We have $t \Downarrow_\tau v$ iff $t \rightsquigarrow^\star_\tau u$.

Turing-completeness

PCF is Turing-complete: for every partial recursive function ϕ, there is a PCF term $\underline{\phi} \in \text{PCF}_{\mathtt{nat} \to \mathtt{nat}}$ such that for all $n \in \mathbb{N}$, if $\phi(n)$ is defined then $\underline{\phi}\ \underline{n} \Downarrow_{\mathtt{nat}} \underline{\phi(n)}$.

Turing-completeness

PCF is Turing-complete: for every partial recursive function ϕ, there is a PCF term $\underline{\phi} \in \text{PCF}_{\mathtt{nat}\to\mathtt{nat}}$ such that for all $n \in \mathbb{N}$, if $\phi(n)$ is defined then $\underline{\phi}\ \underline{n} \Downarrow_{\mathtt{nat}} \underline{\phi(n)}$.

(Later on: $\phi = [\![\underline{\phi}]\!]$).