Denotational Semantics

Ioannis Markakis

Lectures for Part II CST 2026/2027

Recap of Lecture 5

We introduced the simply typed language Pcf:

- We defined its types, terms, and typing contexts
- We defined inductively the typing relation $\Gamma \vdash t : \tau$
- We gave operational semantics using an evaluation relation $t \Downarrow_\tau v$
- We discussed divergence $t \Uparrow_\tau$ and showed $\Omega_\tau \stackrel{\text{def}}{=} \mathtt{fix}(\mathtt{fun}\, x{:}\tau.\, x) \Uparrow_\tau$

Determinism

Evaluation in PCF is deterministic: if both $t \Downarrow_\tau v$ and $t \Downarrow_\tau v'$ hold, then $v = v'$.

In other words, the relation $\Downarrow_\tau$ is a partial function for every type τ.

Determinism

Evaluation in PCF is deterministic: if both $t \Downarrow_\tau v$ and $t \Downarrow_\tau v'$ hold, then $v = v'$.

In other words, the relation $\Downarrow_\tau$ is a partial function for every type τ.

Proof by (rule) induction on evaluation $\Downarrow$:

$$P(t, \tau, v) \quad \stackrel{\text{def}}{=} \quad \forall v' \in \text{PCF}_\tau . (t \Downarrow_\tau v' \Rightarrow v = v')$$

Intuition: there is always exactly one rule which applies.

Contextual equivalence

Contextual equivalence – informal

Two phrases of a programming language are **contextually equivalent** if any occurrences of the first phrase in a **complete program** can be replaced by the second phrase without affecting the **observable results** of executing the program.

Contextual equivalence – informal

Two phrases of a programming language are **contextually equivalent** if any occurrences of the first phrase in a **complete program** can be replaced by the second phrase without affecting the **observable results** of executing the program.

The intuitive notion of **program equivalence** for programmers.

Contextual equivalence – informal

Two phrases of a programming language are **contextually equivalent** if any occurrences of the first phrase in a **complete program** can be replaced by the second phrase without affecting the **observable results** of executing the program.

The intuitive notion of **program equivalence** for programmers.

But what's a complete program? What's an observable result?

Evaluation contexts

"Term with a hole":

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

Evaluation contexts

"Term with a hole":

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

Typing extended to evaluation contexts: $\Gamma \vdash_{\Delta,\sigma} \mathcal{C} : \tau$, to be read as the evaluation context has type τin context Γ, assuming the hole has type σ in context Δ.

Evaluation contexts

"Term with a hole":

$$\begin{array}{rcl} \mathcal{C} & ::= & - \mid \texttt{succ}\,(\mathcal{C}) \mid \texttt{pred}\,(\mathcal{C}) \mid \texttt{zero?}\,(\mathcal{C}) \mid \\ & & \texttt{if}\ \mathcal{C}\ \texttt{then}\ t\ \texttt{else}\ t \mid \texttt{if}\ t\ \texttt{then}\ \mathcal{C}\ \texttt{else}\ t \mid \texttt{if}\ t\ \texttt{then}\ t\ \texttt{else}\ \mathcal{C} \mid \\ & & \texttt{fun}\, x{:}\tau.\, \mathcal{C} \mid \mathcal{C}\ t \mid t\ \mathcal{C} \mid \texttt{fix}(\mathcal{C}) \end{array}$$

Typing extended to evaluation contexts: $\Gamma \vdash_{\Delta,\sigma} \mathcal{C} : \tau$, to be read as the evaluation context has type τin context Γ, assuming the hole has type σ in context Δ.

$$\frac{}{\Gamma \vdash_{\Gamma,\tau} - : \tau} \qquad \frac{\Gamma \vdash_{\Delta,\sigma} \mathcal{C} : \tau_1 \to \tau_2 \qquad \Gamma \vdash u : \tau_1}{\Gamma \vdash_{\Delta,\sigma} \mathcal{C}\ u : \tau_2} \qquad \ldots$$

Contextual equivalence

Definition

We say that terms $\Gamma \vdash t : \tau$ and $\Gamma \vdash t' : \tau$ of the same type in the same typing context are contextually equivalent and we write $\Gamma \vdash t \cong_{\mathrm{ctx}} t' : \tau$ when for every ground type $\gamma \in \{\mathtt{nat}, \mathtt{bool}\}$, every evaluation context $\cdot \vdash_{\Gamma,\tau} \mathcal{C} : \gamma$ and every value $v \in \mathrm{PCF}_\gamma$:

$$\mathcal{C}[t] \Downarrow_\gamma v \Leftrightarrow \mathcal{C}[t'] \Downarrow_\gamma v.$$

- Since evaluation is deterministic that means that $\mathcal{C}[t]$ converges if and only if $\mathcal{C}[t']$ converges, and they have the same value.
- When Γ is empty, we simply write $t \cong_{\mathrm{ctx}} t' : \tau$
- Divergence is implicitly covered by taking $\mathcal{C} = -$
- Hard to check due to quantifying over all evaluation contexts!

Denotational Semantics for Pcf

Denotational Semantics for Pcf

Introducing denotational semantics

The aims of denotational semantics

We want to approximate contextual equivalence by semantic equality.

The aims of denotational semantics

We want to approximate contextual equivalence by semantic equality.

Denotational semantics amounts to:

- a mapping of PCF types τ to domains $[\![\tau]\!]$;
- a mapping of PCF contexts Γ to domains $[\![\Gamma]\!]$;
- a mapping of closed, well-typed PCF terms $\cdot \vdash t : \tau$ to elements $[\![t]\!] \in [\![\tau]\!]$;
- a mapping of arbitrary, well-typed terms $\Gamma \vdash t : \tau$ to continuous functions $[\![\Gamma]\!] \to [\![\tau]\!]$

The aims of denotational semantics

We want to approximate contextual equivalence by semantic equality.

Denotational semantics amounts to:

- a mapping of PCF types τ to domains $[\![\tau]\!]$;
- a mapping of PCF contexts Γ to domains $[\![\Gamma]\!]$;
- a mapping of closed, well-typed PCF terms $\cdot \vdash t : \tau$ to elements $[\![t]\!] \in [\![\tau]\!]$;
- a mapping of arbitrary, well-typed terms $\Gamma \vdash t : \tau$ to continuous functions $[\![\Gamma]\!] \to [\![\tau]\!]$

Compositionality: $[\![t]\!] = [\![t']\!] \Rightarrow [\![\mathcal{C}[t]]\!] = [\![\mathcal{C}[t']]\!]$.

Soundness: $t \Downarrow_\tau v \Rightarrow [\![t]\!] = [\![v]\!]$ for any type τ,

Adequacy: $[\![t]\!] = [\![v]\!] \Rightarrow t \Downarrow_\gamma v$ for ground types $\gamma \in \{\mathtt{nat}, \mathtt{bool}\}$

Adequacy for function types?

$$v \stackrel{\text{def}}{=} \mathtt{fun}\, x\colon \mathtt{nat}.\, (\mathtt{fun}\, y\colon \mathtt{nat}.\, y)\, \mathtt{0} \quad \text{and} \quad v' \stackrel{\text{def}}{=} \mathtt{fun}\, x\colon \mathtt{nat}.\, \mathtt{0}.$$

Adequacy for function types?

$$v \stackrel{\text{def}}{=} \texttt{fun}\, x\colon \texttt{nat}.\, (\texttt{fun}\, y\colon \texttt{nat}.\, y)\, \texttt{0} \quad \text{and} \quad v' \stackrel{\text{def}}{=} \texttt{fun}\, x\colon \texttt{nat}.\, \texttt{0}.$$

They are distinct values, so $v \not\Downarrow_{\texttt{nat}\to\texttt{nat}} v'$. However:

- $(\texttt{fun}\, y\colon \texttt{nat}.\, y)\, \texttt{0} \Downarrow_{\texttt{nat}} \texttt{0}$ from the definition of evaluation
- $[\![\texttt{fun}\, y\colon \texttt{nat}.\, y)\, \texttt{0}]\!] = [\![\texttt{0}]\!]$ by soundness
- $[\![v]\!] = [\![v']\!]$ by compositionality for $\mathcal{C} = \texttt{fun}\, x\colon \texttt{nat}.\, -$

Adequacy for function types?

$$v \stackrel{\text{def}}{=} \texttt{fun}\, x\colon \texttt{nat}.\, (\texttt{fun}\, y\colon \texttt{nat}.\, y)\, \texttt{0} \quad \text{and} \quad v' \stackrel{\text{def}}{=} \texttt{fun}\, x\colon \texttt{nat}.\, \texttt{0}.$$

They are distinct values, so $v \not\Downarrow_{\texttt{nat}\to\texttt{nat}} v'$. However:

- $(\texttt{fun}\, y\colon \texttt{nat}.\, y)\, \texttt{0} \Downarrow_{\texttt{nat}} \texttt{0}$ from the definition of evaluation
- $[\![\texttt{fun}\, y\colon \texttt{nat}.\, y)\, \texttt{0}]\!] = [\![\texttt{0}]\!]$ by soundness
- $[\![v]\!] = [\![v']\!]$ by compositionality for $\mathcal{C} = \texttt{fun}\, x\colon \texttt{nat}.\, -$

Therefore $[\![v]\!] = [\![v']\!] \not\Rightarrow v \Downarrow_{\tau\to\sigma} v'$ in general

Proof principle: to show

$$\cdot \vdash t_1 \cong_{\mathrm{ctx}} t_2 : \tau$$

it suffices to establish

$$[\![t_1]\!] = [\![t_2]\!] \in [\![\tau]\!]$$

Proof principle: to show

$$\cdot \vdash t_1 \cong_{\text{ctx}} t_2 : \tau$$

it suffices to establish

$$[\![t_1]\!] = [\![t_2]\!] \in [\![\tau]\!]$$

For every evaluation context $\cdot \vdash_{\cdot,\tau} \mathcal{C} : \mathtt{nat}$ and value $v \in \text{PCF}_{\mathtt{nat}}$

$$\begin{aligned} \mathcal{C}[t_1] \Downarrow_{\mathtt{nat}} v &\Rightarrow [\![\mathcal{C}[t_1]]\!] = [\![v]\!] && \text{(soundness)} \\ &\Rightarrow [\![\mathcal{C}[t_2]]\!] = [\![v]\!] && \text{(compositionality on } [\![t_1]\!] = [\![t_2]\!]) \\ &\Rightarrow \mathcal{C}[t_2] \Downarrow_{\mathtt{nat}} v && \text{(adequacy)} \end{aligned}$$

Proof principle: to show

$$\cdot \vdash t_1 \cong_{\text{ctx}} t_2 : \tau$$

it suffices to establish

$$[\![t_1]\!] = [\![t_2]\!] \in [\![\tau]\!]$$

For every evaluation context $\cdot \vdash_{\cdot,\tau} \mathcal{C} : \mathtt{nat}$ and value $v \in \text{PCF}_{\mathtt{nat}}$

$$\begin{aligned} \mathcal{C}[t_1] \Downarrow_{\mathtt{nat}} v &\Rightarrow [\![\mathcal{C}[t_1]]\!] = [\![v]\!] && \text{(soundness)} \\ &\Rightarrow [\![\mathcal{C}[t_2]]\!] = [\![v]\!] && \text{(compositionality on } [\![t_1]\!] = [\![t_2]\!]) \\ &\Rightarrow \mathcal{C}[t_2] \Downarrow_{\mathtt{nat}} v && \text{(adequacy)} \end{aligned}$$

and symmetrically for $\mathcal{C}[t_2] \Downarrow_{\mathtt{nat}} v \Rightarrow \mathcal{C}[t_1] \Downarrow_{\mathtt{nat}} v$, and similarly for `bool`.

The power of denotational semantics

Proof principle: to show

$$\cdot \vdash t_1 \cong_{\mathrm{ctx}} t_2 : \tau$$

it suffices to establish

$$[\![t_1]\!] = [\![t_2]\!] \in [\![\tau]\!]$$

Denotational equality is sound, but is it complete?
Does equality in the model imply contextual equivalence?

The power of denotational semantics

Proof principle: to show

$$\cdot \vdash t_1 \cong_{\mathrm{ctx}} t_2 : \tau$$

it suffices to establish

$$[\![t_1]\!] = [\![t_2]\!] \in [\![\tau]\!]$$

Denotational equality is sound, but is it complete?
Does equality in the model imply contextual equivalence?

Full abstraction.

Denotational Semantics for Pcf

Definition

Semantics of types

$$[\![\texttt{nat}]\!] \stackrel{\text{def}}{=} \mathbb{N}_\bot \qquad \text{(flat domain)}$$

$$[\![\texttt{bool}]\!] \stackrel{\text{def}}{=} \mathbb{B}_\bot \qquad \text{(flat domain)}$$

$$[\![\tau \to \tau']\!] \stackrel{\text{def}}{=} [\![\tau]\!] \to [\![\tau']\!] \qquad \text{(function domain)}$$

Semantics of contexts

$$[\![\Gamma]\!] \stackrel{\text{def}}{=} \prod_{x \in \text{dom}(\Gamma)} [\![\Gamma(x)]\!] \qquad \text{(environment)}$$

Semantics of contexts

$$[\![\Gamma]\!] \overset{\text{def}}{=} \prod_{x \in \mathrm{dom}(\Gamma)} [\![\Gamma(x)]\!] \qquad \text{(environment)}$$

- $[\![\cdot]\!] = \mathbb{1}$ (one element set)
- $[\![x : \tau]\!] = (\{x\} \to [\![\tau]\!]) \cong [\![\tau]\!]$
- $[\![x_1 : \tau_1, \ldots, x_n : \tau_n]\!] \cong [\![\tau_1]\!] \times \cdots \times [\![\tau_n]\!]$

To every typing judgement

$$\Gamma \vdash t : \tau$$

we associate a continuous function

$$[\![\Gamma \vdash t : \tau]\!] : [\![\Gamma]\!] \to [\![\tau]\!]$$

between domains. In other words,

$$[\![-]\!] : \mathsf{Pcf}_{\Gamma,\tau} \to [\![\Gamma]\!] \to [\![\tau]\!]$$

Denotation of operations on $\mathbb{B}$ and $\mathbb{N}$

$$\begin{array}{lrcl} \text{succ}: & \mathbb{N} & \to & \mathbb{N} \\ & n & \mapsto & n+1 \end{array}$$

$$\begin{array}{lrcl} \text{pred}: & \mathbb{N} & \rightharpoonup & \mathbb{N} \\ & n+1 & \mapsto & n \\ & 0 & & \text{undefined} \end{array}$$

$$\begin{array}{lrcl} \text{zero?}: & \mathbb{N} & \to & \mathbb{B} \\ & 0 & \mapsto & \text{true} \\ & n+1 & \mapsto & \text{false} \end{array}$$

Denotation of operations on $\mathbb{B}$ and $\mathbb{N}$

$$\begin{array}{lrcl} \mathrm{succ}_\perp : & \mathbb{N}_\perp & \to & \mathbb{N}_\perp \\ & n & \mapsto & n+1 \\ & \perp & \mapsto & \perp \end{array}$$

$$\begin{array}{lrcl} \mathrm{pred}_\perp : & \mathbb{N}_\perp & \to & \mathbb{N}_\perp \\ & n+1 & \mapsto & n \\ & 0 & \mapsto & \perp \\ & \perp & \mapsto & \perp \end{array}$$

$$\begin{array}{lrcl} \mathrm{zero?}_\perp : & \mathbb{N}_\perp & \to & \mathbb{B}_\perp \\ & 0 & \mapsto & \mathrm{true} \\ & n+1 & \mapsto & \mathrm{false} \\ & \perp & \mapsto & \perp \end{array}$$

Denotation of operations on $\mathbb{B}$ and $\mathbb{N}$

$$
\begin{aligned}
[\![\texttt{0}]\!](\rho) &\stackrel{\text{def}}{=} 0 && \in \mathbb{N}_\perp \\
[\![\texttt{true}]\!](\rho) &\stackrel{\text{def}}{=} \text{true} && \in \mathbb{B}_\perp \\
[\![\texttt{false}]\!](\rho) &\stackrel{\text{def}}{=} \text{false} && \in \mathbb{B}_\perp
\end{aligned}
$$

Denotation of operations on $\mathbb{B}$ and $\mathbb{N}$

$$
\begin{aligned}
[\![\mathtt{0}]\!](\rho) &\stackrel{\text{def}}{=} 0 && \in \mathbb{N}_\perp \\
[\![\mathtt{true}]\!](\rho) &\stackrel{\text{def}}{=} \text{true} && \in \mathbb{B}_\perp \\
[\![\mathtt{false}]\!](\rho) &\stackrel{\text{def}}{=} \text{false} && \in \mathbb{B}_\perp \\
\\
[\![\mathtt{succ}\,(t)]\!](\rho) &\stackrel{\text{def}}{=} \text{succ}_\perp([\![t]\!](\rho)) && \in \mathbb{N}_\perp \\
[\![\mathtt{pred}\,(t)]\!](\rho) &\stackrel{\text{def}}{=} \text{pred}_\perp([\![t]\!](\rho)) && \in \mathbb{N}_\perp \\
[\![\mathtt{zero?}\,(t)]\!](\rho) &\stackrel{\text{def}}{=} \text{zero?}_\perp([\![t]\!](\rho)) && \in \mathbb{B}_\perp
\end{aligned}
$$

$$[\![\mathtt{succ}\,(t)]\!] = \text{succ}_\perp \circ [\![t]\!]$$

Denotation of operations on $\mathbb{B}$ and $\mathbb{N}$

$$
\begin{array}{rcll}
[\![\mathtt{0}]\!](\rho) & \stackrel{\text{def}}{=} & 0 & \in \mathbb{N}_\perp \\
[\![\mathtt{true}]\!](\rho) & \stackrel{\text{def}}{=} & \text{true} & \in \mathbb{B}_\perp \\
[\![\mathtt{false}]\!](\rho) & \stackrel{\text{def}}{=} & \text{false} & \in \mathbb{B}_\perp \\
 & & & \\
[\![\mathtt{succ}\,(t)]\!](\rho) & \stackrel{\text{def}}{=} & \text{succ}_\perp([\![t]\!](\rho)) & \in \mathbb{N}_\perp \\
[\![\mathtt{pred}\,(t)]\!](\rho) & \stackrel{\text{def}}{=} & \text{pred}_\perp([\![t]\!](\rho)) & \in \mathbb{N}_\perp \\
[\![\mathtt{zero?}\,(t)]\!](\rho) & \stackrel{\text{def}}{=} & \text{zero?}_\perp([\![t]\!](\rho)) & \in \mathbb{B}_\perp \\
 & & & \\
[\![\mathtt{if}\ b\ \mathtt{then}\ t\ \mathtt{else}\ t']\!] & \stackrel{\text{def}}{=} & \text{if}\,([\![b]\!](\rho), [\![t]\!](\rho), [\![t']\!](\rho)) & \in [\![\tau]\!]
\end{array}
$$

$$
[\![\mathtt{if}\ b\ \mathtt{then}\ t\ \mathtt{else}\ t']\!] = \text{if} \circ \langle [\![b]\!], \langle [\![t]\!], [\![t']\!] \rangle \rangle
$$

Denotation of the λ-calculus operations

$$[\![x]\!](\rho) \overset{\text{def}}{=} \rho(x) \qquad \in [\![\Gamma(x)]\!]$$

$$[\![x]\!](\rho) = \pi_x(\rho)$$

Denotation of the λ-calculus operations

$$\begin{aligned} [\![x]\!](\rho) &\stackrel{\text{def}}{=} \rho(x) && \in [\![\Gamma(x)]\!] \\ [\![t_1\ t_2]\!](\rho) &\stackrel{\text{def}}{=} ([\![t_1]\!](\rho))\ ([\![t_2]\!](\rho)) \end{aligned}$$

$$[\![t_1\ t_2]\!] = \mathrm{eval} \circ \langle [\![t_1]\!], [\![t_2]\!] \rangle$$

Denotation of the λ-calculus operations

$$
\begin{aligned}
[\![x]\!](\rho) &\stackrel{\text{def}}{=} \rho(x) && \in [\![\Gamma(x)]\!] \\
[\![t_1\ t_2]\!](\rho) &\stackrel{\text{def}}{=} ([\![t_1]\!](\rho))\ ([\![t_2]\!](\rho)) \\
[\![\mathtt{fun}\ x{:}\tau.\ t]\!](\rho) &\stackrel{\text{def}}{=} \lambda d \in [\![\tau]\!].\ [\![t]\!](\rho, d)
\end{aligned}
$$

$$
[\![\mathtt{fun}\ x{:}\tau.\ t]\!] = \mathrm{cur}([\![t]\!])
$$

Denotation of fixed points

$$[\![\texttt{fix}\ f]\!](\rho) \stackrel{\text{def}}{=} \mathrm{fix}([\![f]\!](\rho))$$

Denotation of Pcf terms

For any Pcf term t such that $\Gamma \vdash t : \tau$, the object $[\![t]\!]$
is well-defined and a continuous function $[\![t]\!] : [\![\Gamma]\!] \to [\![\tau]\!]$.

Denotation of Pcf terms

For any Pcf term t such that $\Gamma \vdash t : \tau$, the object $[\![t]\!]$
is well-defined and a continuous function $[\![t]\!] : [\![\Gamma]\!] \to [\![\tau]\!]$.

If $t \in \mathrm{Pcf}_\tau$: $\quad [\![t]\!] \quad \in \quad [\![\cdot]\!] \to [\![\tau]\!] \quad = \quad \mathbb{1} \to [\![\tau]\!] \quad \cong \quad [\![\tau]\!]$

Denotational Semantics for Pcf

Compositionality

Compositionality

Suppose $t, u \in \mathrm{PCF}_{\Delta,\sigma}$, such that

$$[\![t]\!] = [\![u]\!] : [\![\Delta]\!] \to [\![\sigma]\!]$$

Suppose moreover that $\mathcal{C}[-]$ is a PCF context such that $\Gamma \vdash_{\Delta,\sigma} \mathcal{C} : \tau$. Then

$$[\![\mathcal{C}[t]]\!] = [\![\mathcal{C}[u]]\!] : [\![\Gamma]\!] \to [\![\tau]\!].$$

A denotation for evaluation contexts

If $\Gamma \vdash_{\Delta,\sigma} \mathcal{C} : \tau$, then define $[\![\mathcal{C}]\!]$ such that

$$[\![\mathcal{C}]\!] : ([\![\Delta]\!] \to [\![\sigma]\!]) \to [\![\Gamma]\!] \to [\![\tau]\!]$$

A denotation for evaluation contexts

If $\Gamma \vdash_{\Delta,\sigma} \mathcal{C} : \tau$, then define $[\![\mathcal{C}]\!]$ such that

$$[\![\mathcal{C}]\!] : ([\![\Delta]\!] \to [\![\sigma]\!]) \to [\![\Gamma]\!] \to [\![\tau]\!]$$

$$\begin{aligned} [\![-]\!](d) &= d \\ [\![\mathcal{C}\ t]\!](d)(\rho) &= ([\![\mathcal{C}]\!](d)(\rho))([\![t]\!](\rho)) \\ &\vdots \end{aligned}$$