Topic 8

Full Abstraction

Proof principle

For all types τ and closed terms $M_1, M_2 \in \mathrm{PCF}_\tau$,

$$[\![M_1]\!] = [\![M_2]\!] \text{ in } [\![\tau]\!] \implies M_1 \cong_{\mathrm{ctx}} M_2 : \tau \ .$$

Hence, to prove

$$M_1 \cong_{\mathrm{ctx}} M_2 : \tau$$

it suffices to establish

$$[\![M_1]\!] = [\![M_2]\!] \text{ in } [\![\tau]\!] \ .$$

Full abstraction

A denotational model is said to be *fully abstract* whenever denotational equality characterises contextual equivalence.

- The domain model of PCF is *not* fully abstract.

  In other words, there are contextually equivalent PCF terms with different denotations.

Failure of full abstraction, idea

We will construct two closed terms

$$T_1, T_2 \in \mathrm{PCF}_{(bool \to (bool \to bool)) \to bool}$$

such that

$$T_1 \cong_{\mathrm{ctx}} T_2$$

and

$$[\![T_1]\!] \neq [\![T_2]\!]$$

- Recall that

$$T_1 \simeq_{ctx} T_2 : (\text{bool} \to (\text{bool} \to \text{bool})) \to \text{bool}$$

iff

$$\forall M : \text{PCF}_{\text{bool} \to (\text{bool} \to \text{bool})}.\ \forall V : \text{PCF}_{\text{bool}}.$$

$$T_1\, M \Downarrow_{\text{bool}} V \iff T_2\, M \Downarrow_{\text{bool}} V$$

- In particular, we will achieve $T_1 \simeq_{ctx} T_2$ by making sure that

$$\forall M : \text{PCF}_{\text{bool} \to (\text{bool} \to \text{bool})}.$$

$$T_1\, M \not\Downarrow_{\text{bool}} \quad \text{and} \quad T_2\, M \not\Downarrow_{\text{bool}}$$

- We achieve $T_1 \cong_{\mathrm{ctx}} T_2$ by making sure that

$$\forall\, M \in \mathrm{PCF}_{bool\to(bool\to bool)}\,(\,T_1\, M \not\Downarrow_{bool}\ \&\ T_2\, M \not\Downarrow_{bool}\,)$$

Hence,

$$[\![T_1]\!]([\![M]\!]) = \bot = [\![T_2]\!]([\![M]\!])$$

for all $M \in \mathrm{PCF}_{bool\to(bool\to bool)}$.

- We achieve $T_1 \cong_{\mathrm{ctx}} T_2$ by making sure that

$$\forall\, M \in \mathrm{PCF}_{bool\to(bool\to bool)}\,(\,T_1\,M \not\Downarrow_{bool} \;\&\; T_2\,M \not\Downarrow_{bool}\,)$$

$$[\![T_1]\!] \neq [\![T_2]\!] : (\mathbb{B}_\perp \to (\mathbb{B}_\perp \to \mathbb{B}_\perp)) \to \mathbb{B}_\perp$$

$\exists f$

$$[\![T_1]\!](f) \neq [\![T_2]\!](f)$$

for some $f \in (\mathbb{B}_\perp \to (\mathbb{B}_\perp \to \mathbb{B}_\perp))$
that is necessarily not definable,
in the sense that

$$f \neq [\![M]\!] \quad \forall M \in PCF_{bool\to(bool\to bool)}$$

- We achieve $T_1 \cong_{\mathrm{ctx}} T_2$ by making sure that

$$\forall\, M \in \mathrm{PCF}_{bool \to (bool \to bool)} \,(\, T_1\, M \not\Downarrow_{bool} \;\&\; T_2\, M \not\Downarrow_{bool} \,)$$

  Hence,

$$[\![T_1]\!]([\![M]\!]) = \bot = [\![T_2]\!]([\![M]\!])$$

  for all $M \in \mathrm{PCF}_{bool \to (bool \to bool)}$.

- We achieve $[\![T_1]\!] \neq [\![T_2]\!]$ by making sure that

$$[\![T_1]\!](por) \neq [\![T_2]\!](por)$$

  for some *non-definable* continuous function

$$por \in (\mathbb{B}_\bot \to (\mathbb{B}_\bot \to \mathbb{B}_\bot))\ .$$

Parallel-or function

is the unique continuous function $por : \mathbb{B}_\perp \to (\mathbb{B}_\perp \to \mathbb{B}_\perp)$ such that

$$\begin{array}{lcl} por\ \ true\ \ \perp & = & true \\ por\ \ \perp\ \ true & = & true \\ por\ \ false\ \ false & = & false \end{array}$$

In which case, it necessarily follows by monotonicity that

$$\begin{array}{lcl} por\ \ true\ \ true & = & true \\ por\ \ true\ \ false & = & true \\ por\ \ false\ \ true & = & true \end{array} \qquad \begin{array}{lcl} por\ \ false\ \ \perp & = & \perp \\ por\ \ \perp\ \ false & = & \perp \\ por\ \ \perp\ \ \perp & = & \perp \end{array}$$

Undefinability of parallel-or

Proposition. *There is no closed PCF term*

$$P : bool \to (bool \to bool)$$

satisfying

$$[\![P]\!] = por : \mathbb{B}_\perp \to (\mathbb{B}_\perp \to \mathbb{B}_\perp) \ .$$

Parallel-or test functions

NB: One may define a program $T \in PCF_{(bool \to (bool \to bool)) \to bool}$ That tests whether its input behaves as por and loops otherwise.

```
T = fn f : bool → (bool → bool)
    if (f true Ω)
    then if (f Ω true)
         then if (f false false)
              then Ω
              else .... input behaves like por ....
         else Ω
    else Ω
```

In particular,

$$T M \not\models_{bool} \quad \forall M : bool \to (bool \to bool)$$

and we may define two versions of T, say T_1 and T_2, that are contextually equivalent but for which

$$[\![T_1]\!](por) \neq [\![T_2]\!](por)$$

by giving different outputs when the test succeds.

Parallel-or test functions

For $i = 1, 2$ define

```
$T_i \overset{\text{def}}{=}$  $\mathbf{fn}\, f : \mathit{bool} \to (\mathit{bool} \to \mathit{bool})\, .$
          $\mathbf{if}\ (f\ \mathbf{true}\ \Omega)\ \mathbf{then}$
            $\mathbf{if}\ (f\ \Omega\ \mathbf{true})\ \mathbf{then}$
              $\mathbf{if}\ (f\ \mathbf{false}\ \mathbf{false})\ \mathbf{then}\ \Omega\ \mathbf{else}\ B_i$
            $\mathbf{else}\ \Omega$
          $\mathbf{else}\ \Omega$
```

where $B_1 \overset{\text{def}}{=} \mathbf{true}$, $B_2 \overset{\text{def}}{=} \mathbf{false}$,
and $\Omega \overset{\text{def}}{=} \mathbf{fix}(\mathbf{fn}\, x : \mathit{bool}\, .\, x)$.

Failure of full abstraction

Proposition.

$$T_1 \cong_{\mathrm{ctx}} T_2 : (bool \to (bool \to bool)) \to bool$$

$$[\![T_1]\!] \neq [\![T_2]\!] \in (\mathbb{B}_\perp \to (\mathbb{B}_\perp \to \mathbb{B}_\perp)) \to \mathbb{B}_\perp$$

PCF+por

Expressions $M ::= \cdots \mid \mathbf{por}(M, M)$

Typing $$\frac{\Gamma \vdash M_1 : bool \quad \Gamma \vdash M_2 : bool}{\Gamma \vdash \mathbf{por}(M_1, M_2) : bool}$$

Evaluation

$$\frac{M_1 \Downarrow_{bool} \mathbf{true}}{\mathbf{por}(M_1, M_2) \Downarrow_{bool} \mathbf{true}} \qquad \frac{M_2 \Downarrow_{bool} \mathbf{true}}{\mathbf{por}(M_1, M_2) \Downarrow_{bool} \mathbf{true}}$$

$$\frac{M_1 \Downarrow_{bool} \mathbf{false} \quad M_2 \Downarrow_{bool} \mathbf{false}}{\mathbf{por}(M_1, M_2) \Downarrow_{bool} \mathbf{false}}$$

Plotkin’s full abstraction result

The denotational semantics of PCF+por is given by extending that of PCF with the clause

$$[\![\Gamma \vdash \mathbf{por}(M_1, M_2)]\!](\rho) \overset{\text{def}}{=} por\big([\![\Gamma \vdash M_1]\!](\rho)\big)\big([\![\Gamma \vdash M_2]\!](\rho)\big)$$

This denotational semantics is fully abstract for contextual equivalence of PCF+por terms:

$$\Gamma \vdash M_1 \cong_{\text{ctx}} M_2 : \tau \;\Leftrightarrow\; [\![\Gamma \vdash M_1]\!] = [\![\Gamma \vdash M_2]\!].$$

PoR is not definable

- Domains nat bool σ→τ. σ×τ

⇝ stable

ωcpo meets of [consistent / bounded] elements ⇝ are continuous

$$\left(\bigsqcup_i x_i\right) \wedge y = \bigsqcup_i (x_i \wedge y)$$

⇝ bounded.

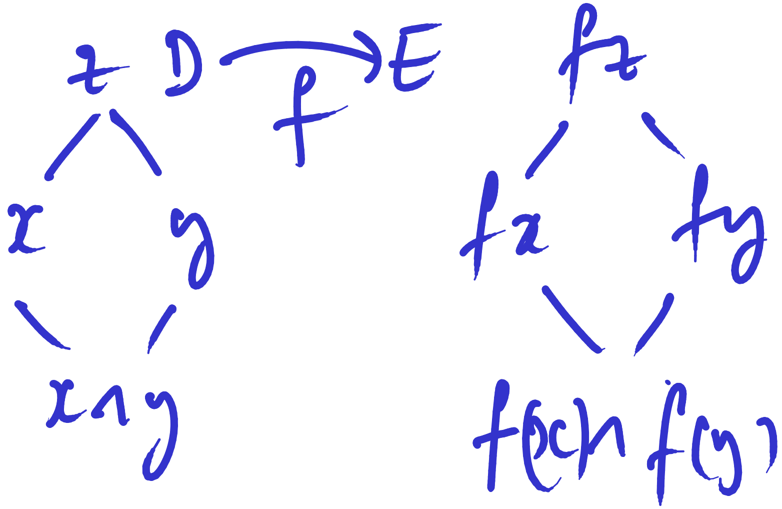
Example: $(X \Rightarrow Y)$ domain

$f, g \in (X \Rightarrow Y)$ consistent iff def $\exists h \in (X \Rightarrow Y)$.

bounded meet ⇝

$$\begin{tikzcd}
 & h & \\
f \arrow[ur, phantom, "\sqsubseteq" sloped] & & g \arrow[ul, phantom, "\sqsupseteq" sloped] \\
 & f \wedge g \arrow[ul, phantom, "\sqsubseteq" sloped] \arrow[ur, phantom, "\sqsupseteq" sloped] &
\end{tikzcd}$$

- $[\![\text{nat}]\!] = \mathbb{N}_\perp$ $\qquad [\![\text{bool}]\!] = \mathbb{B}_\perp$
- D, E stable domains ⇒ D × E stable domain.
  with bounded meets given pointwise.
- D, E stable domains
  ⇒ There is a domain of stable functions.
  monotone
  continuous
  bounded-meet preserving



$f(x) \wedge f(y) = f(x \wedge y)$

$f, g : D \to E$ stable

$f \sqsubseteq g$ iff. $\forall d \in D \;\; f(d) \sqsubseteq g(d)$ in E.

$\forall d_1, d_2 \in D.$

$$d_1 \sqsubseteq d_2$$

$$\Downarrow$$

$$f(d_1) = g(d_1) \wedge f(d_2).$$

$$\begin{tikzcd}
 & g(d_2) & \\
g(d_1) \arrow[ur, phantom, "\sqsubseteq" sloped] & & f(d_2) \arrow[ul, phantom, "\sqsupseteq" sloped] \\
 & f(d_1) \arrow[ul, phantom, "\sqsupseteq" sloped] \arrow[ur, phantom, "\sqsubseteq" sloped] &
\end{tikzcd}$$

$\rightsquigarrow$ gives a stable domain.

$\underline{\text{eval}} : (D \to E) \times D \xrightarrow[\underline{\text{stable}}]{} E$

given

$$\frac{D \times E \xrightarrow[\text{stable}]{f} F}{D \xrightarrow[\text{stable.}]{\text{curry}(f)} (E \to F)}$$

domain of stable functions.

$$(D \to D) \xrightarrow[\text{stable}]{\text{fix}} D$$

$$f \longmapsto \bigsqcup_n f^n(\bot).$$

- $por : \mathbb{B}_\perp \to \mathbb{B}_\perp \to \mathbb{B}_\perp$

  is <u>not stable!</u>

  $\Downarrow$

  is not PCF definable

- Still the stable model is not fully abstract.