Full Abstraction
Proof principle

For all types \( \tau \) and closed terms \( M_1, M_2 \in \text{PCF}_\tau \),

\[ [\tau] \text{ in } [M_2] = [M_1] \]

Hence, to prove \( M_1 \sim_{\text{ctx}} M_2 : \tau \) it suffices to establish

\[ \tau : \text{ctx} M_2 \cong M_1 \]

For all types \( \tau \) and closed terms \( M_1, M_2 \in \text{PCF}_\tau \),

Proof Principle
with different denotations.

In other words, there are contextually equivalent $\text{PCF}$ terms with different denotations.

The domain model of $\text{PCF}$ is not fully abstract.

A denotational model is said to be fully abstract whenever denotation equality characterises contextual equivalence.
Failure of full abstraction, idea

We will construct two closed terms $T_1, T_2 \in \text{PCF}$ such that

\[ [T_2] \not= [T_1] \]

and

\[ T_2 \overset{\text{ctx}}{\approx} T_1 \]

such that

\[ \text{let } l_2 = \text{let } l_1 = \text{let } b_1 = \text{let } b_2 = \text{let } l_0 = \text{let } \text{PCF} \text{ in } l_1, l_2 \]

We will construct two closed terms

Fault of full abstraction, idea
We achieve $T_1 \equiv T_2$ by making sure that $\forall M \in \text{PCF} \rightarrow \text{bool} \rightarrow \text{bool}$

$(\text{por})[\downarrow L] \not= (\text{por})[\uparrow L]$

Hence, we achieve $[T_1] = [T_2]$ for some non-definable continuous function $\text{por} \in (\downarrow \bot \rightarrow (\downarrow \bot \rightarrow \uparrow \bot))$.

For all $M \in \text{PCF}$, we achieve $[W][\downarrow L] = \top = ([W])'[\uparrow L]$
Parallel-or function isthe unique continuous function por:

<table>
<thead>
<tr>
<th>⊥</th>
<th>⊥</th>
<th>⊥</th>
</tr>
</thead>
<tbody>
<tr>
<td>⊥</td>
<td>⊥</td>
<td>⊥</td>
</tr>
<tr>
<td>⊥</td>
<td>⊥</td>
<td>⊥</td>
</tr>
</tbody>
</table>

In which case, it necessarily follows by monotonicity that

<table>
<thead>
<tr>
<th>⊥</th>
<th>⊥</th>
<th>⊥</th>
</tr>
</thead>
<tbody>
<tr>
<td>⊥</td>
<td>⊥</td>
<td>⊥</td>
</tr>
<tr>
<td>⊥</td>
<td>⊥</td>
<td>⊥</td>
</tr>
<tr>
<td>⊥</td>
<td>⊥</td>
<td>⊥</td>
</tr>
</tbody>
</table>

that

<table>
<thead>
<tr>
<th>⊥</th>
<th>⊥</th>
<th>⊥</th>
</tr>
</thead>
<tbody>
<tr>
<td>⊥</td>
<td>⊥</td>
<td>⊥</td>
</tr>
<tr>
<td>⊥</td>
<td>⊥</td>
<td>⊥</td>
</tr>
<tr>
<td>⊥</td>
<td>⊥</td>
<td>⊥</td>
</tr>
</tbody>
</table>

is the unique continuous function por: por

Parall-or function
Proposition. Undefinability of parallel-or

There is no closed PCF term $P : \text{bool} \rightarrow (\text{bool} \rightarrow \text{bool})$ satisfying $\llbracket P \rrbracket = \text{par} \circ \bot \rightarrow (\bot \rightarrow \bot)$.
Parallel-or test functions

For $i = 1, 2$, define

Parallelor test functions
106

\[ \tau_1 \leftarrow \left( (\tau_1 \leftarrow \tau_1) \leftarrow \tau_1 \right) \in \llbracket \mathcal{L} \rrbracket \neq \llbracket \mathcal{L} \rrbracket \]

100q \leftarrow \left( (100q \leftarrow 100q) \leftarrow 100q \right) : \llbracket \mathcal{L} \rrbracket^x_{\text{ctx}} \approx \llbracket \mathcal{L} \rrbracket

Proposition.

Failure of full abstraction
Expressions

\[ P ::= \cdots \mid \text{por} (M, M) \]

Typing

\[ \Gamma \vdash M_1 : \text{bool} \]
\[ \Gamma \vdash M_2 : \text{bool} \]

\[ \Gamma \vdash \text{por} (M_1, M_2) : \text{bool} \]

Evaluation

\[ M_1 \Downarrow \text{false} \]
\[ \text{por} (M_1, M_2) \Downarrow \text{false} \]
\[ M_2 \Downarrow \text{false} \]
\[ \text{por} (M_1, M_2) \Downarrow \text{false} \]

\[ M_1 \Downarrow \text{true} \]
\[ \text{por} (M_1, M_2) \Downarrow \text{true} \]
\[ M_2 \Downarrow \text{true} \]
\[ \text{por} (M_1, M_2) \Downarrow \text{true} \]

Typing

\[ \text{por} (M, M) : \text{bool} \]
\[ \text{por} (M, M) : \text{bool} \]

Expressions

\[ (M, M) : \text{por} | \cdots = : M \]

PCF + por
Plotkin's full abstraction result

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

\[ \llbracket □ M \longrightarrow J \rrbracket = \llbracket □ M \longrightarrow J \rrbracket \iff \exists \rho : □ M \cong □ M \]

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

\[ \Gamma \Downarrow M_1 \leadsto \text{ctx} \Downarrow M_2 : \tau \iff \llbracket □ M_1 \rrbracket = \llbracket □ M_2 \rrbracket \]