

PCF syntax

Types

$$\tau ::= \mathit{nat} \mid \mathit{bool} \mid \tau \rightarrow \tau$$

Expressions

$$\begin{aligned} M ::= & \mathbf{0} \mid \mathbf{succ}(M) \mid \mathbf{pred}(M) \\ & \mid \mathbf{true} \mid \mathbf{false} \mid \mathbf{zero}(M) \\ & \mid x \mid \mathbf{if} \ M \ \mathbf{then} \ M \ \mathbf{else} \ M \\ & \mid \mathbf{fn} \ x : \tau . M \mid M \ M \mid \mathbf{fix}(M) \end{aligned}$$

where $x \in \mathbb{V}$, an infinite set of **variables**.

Technicality: We identify expressions up to α -conversion of bound variables (created by the **fn** expression-former): by definition a PCF **term** is an α -equivalence class of expressions.

PCF typing relation (sample rules)

$$(\cdot\text{fn}) \frac{\Gamma[x \mapsto \tau] \vdash M : \tau'}{\Gamma \vdash \mathbf{fn} \ x : \tau . M : \tau \rightarrow \tau'} \quad \text{if } x \notin \text{dom}(\Gamma)$$

$$(\cdot\text{app}) \frac{\Gamma \vdash M_1 : \tau \rightarrow \tau' \quad \Gamma \vdash M_2 : \tau}{\Gamma \vdash M_1 M_2 : \tau'}$$

$$(\cdot\text{fix}) \frac{\Gamma \vdash M : \tau \rightarrow \tau}{\Gamma \vdash \mathbf{fix}(M) : \tau}$$

Idea $F = \underline{\text{fix}}(\lambda f. \lambda x. \dots f \dots x \dots)$

The recursive function $F x = \dots F \dots x \dots$

Suppose $F: \text{nat} \rightarrow \text{nat}$ $G: \text{nat} \rightarrow \text{nat} \rightarrow \text{nat} \rightarrow \text{nat}$

$\boxed{?}$ $H: \text{nat} \rightarrow \text{nat} \rightarrow \text{nat} ?$

Partial recursive functions in PCF

- Primitive recursion.

Idea

$$\begin{cases} h(x, 0) = f(x) \\ h(x, y + 1) = g(x, y, h(x, y)) \end{cases}$$

$$H \ x \ y = \begin{cases} \text{if } (\underline{\text{zero}} \ y) \ \underline{\text{then}} \ F(x) \\ \underline{\text{else}} \ G \ x \ (\underline{\text{pred}} \ y) \ (H \ x \ (\underline{\text{pred}} \ y)) \end{cases}$$

Def:

$$H = \underline{\text{fix}} \ (\lambda h. \lambda x. \lambda y. \begin{cases} \text{if } (\underline{\text{zero}} \ y) \ \underline{\text{then}} \ F(x) \\ \underline{\text{else}} \ G \ x \ (\underline{\text{pred}} \ y) \ (h \ x \ (\underline{\text{pred}} \ y)) \end{cases})$$

Partial recursive functions in PCF

- Primitive recursion.

$$\begin{cases} h(x, 0) = f(x) \\ h(x, y + 1) = g(x, y, h(x, y)) \end{cases}$$

- Minimisation.

$$m(x) = \text{the least } y \geq 0 \text{ such that } k(x, y) = 0$$

Suppose $K: \text{nat} \rightarrow \text{nat} \rightarrow \text{nat}$.

Define $M: \text{nat} \rightarrow \text{nat}$?

def
 $Mx = Fx 0$

$$Fx y = \underline{\text{if}} \text{ zero}(Kxy) \underline{\text{then}} y$$

$$\underline{\text{else}} Fx (\underline{\text{succ}} y)$$

def
 $F = \underline{\text{fix}} (\underline{\lambda} f. \underline{\lambda} x. \underline{\lambda} y. \underline{\text{if}} (\text{zero } Kxy) \underline{\text{then}} y$
 $\underline{\text{else}} f x (\underline{\text{succ}} y))$

$$M = \underline{\text{fix}} x:\text{nat}. Fx 0$$

PCF evaluation relation

takes the form

$$M \Downarrow_{\tau} V$$

where

- τ is a PCF type
- $M, V \in \text{PCF}_{\tau}$ are closed PCF terms of type τ
- V is a **value**,

$$V ::= \mathbf{0} \mid \mathbf{succ}(V) \mid \mathbf{true} \mid \mathbf{false} \mid \mathbf{fn } x : \tau . M.$$

values of ground types = nat & bool.

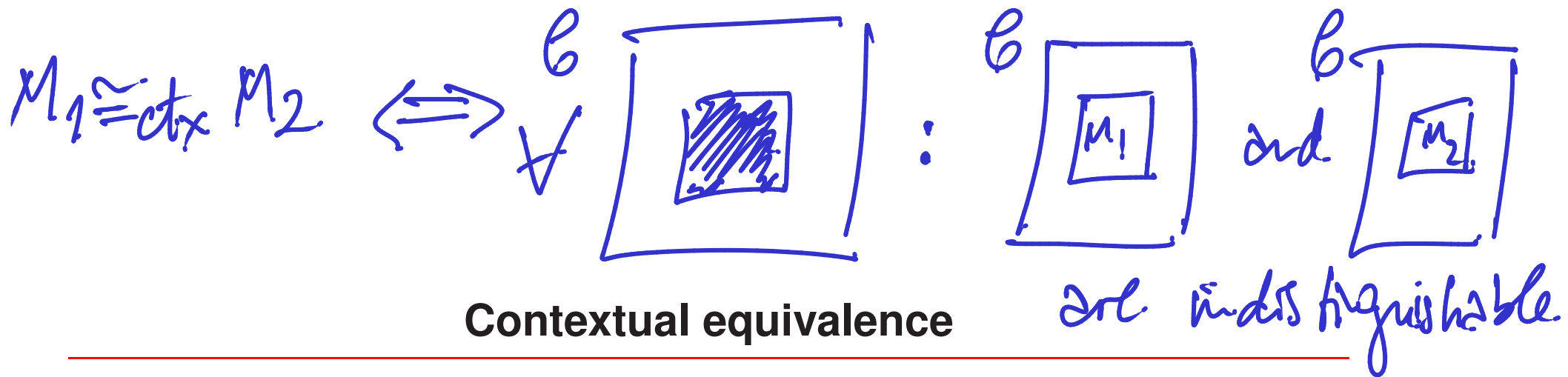
closure

PCF evaluation (sample rules)

$(\Downarrow_{\text{val}})$ $V \Downarrow_{\tau} V$ (V a value of type τ)

$(\Downarrow_{\text{cbn}})$
$$\frac{M_1 \Downarrow_{\tau \rightarrow \tau'} \mathbf{fn} x : \tau . M'_1 \quad M'_1[M_2/x] \Downarrow_{\tau'} V}{M_1 M_2 \Downarrow_{\tau'} V}$$

$(\Downarrow_{\text{fix}})$
$$\frac{M(\mathbf{fix}(M)) \Downarrow_{\tau} V}{\mathbf{fix}(M) \Downarrow_{\tau} V}$$



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.

Remark $(nat \rightarrow nat) \rightarrow (nat \rightarrow nat)$

fn $F: nat \rightarrow nat. F$
fn $F: nat \rightarrow nat. fix: nat.$

Contextual equivalence of PCF terms

$F(x)$
two values
for the
identity
function.

Given PCF terms M_1, M_2 , PCF type τ , and a type environment Γ , the relation $\Gamma \vdash M_1 \cong_{\text{ctx}} M_2 : \tau$ is defined to hold iff

- Both the typings $\Gamma \vdash M_1 : \tau$ and $\Gamma \vdash M_2 : \tau$ hold.
- For all PCF contexts C for which $C[M_1]$ and $C[M_2]$ are closed terms of type γ , where $\gamma = nat$ or $\gamma = bool$, and for all values $V : \gamma$,

$$C[M_1] \Downarrow_{\gamma} V \Leftrightarrow C[M_2] \Downarrow_{\gamma} V.$$

ground types



PCF denotational semantics — aims

- PCF types $\tau \mapsto$ domains $\llbracket \tau \rrbracket$.

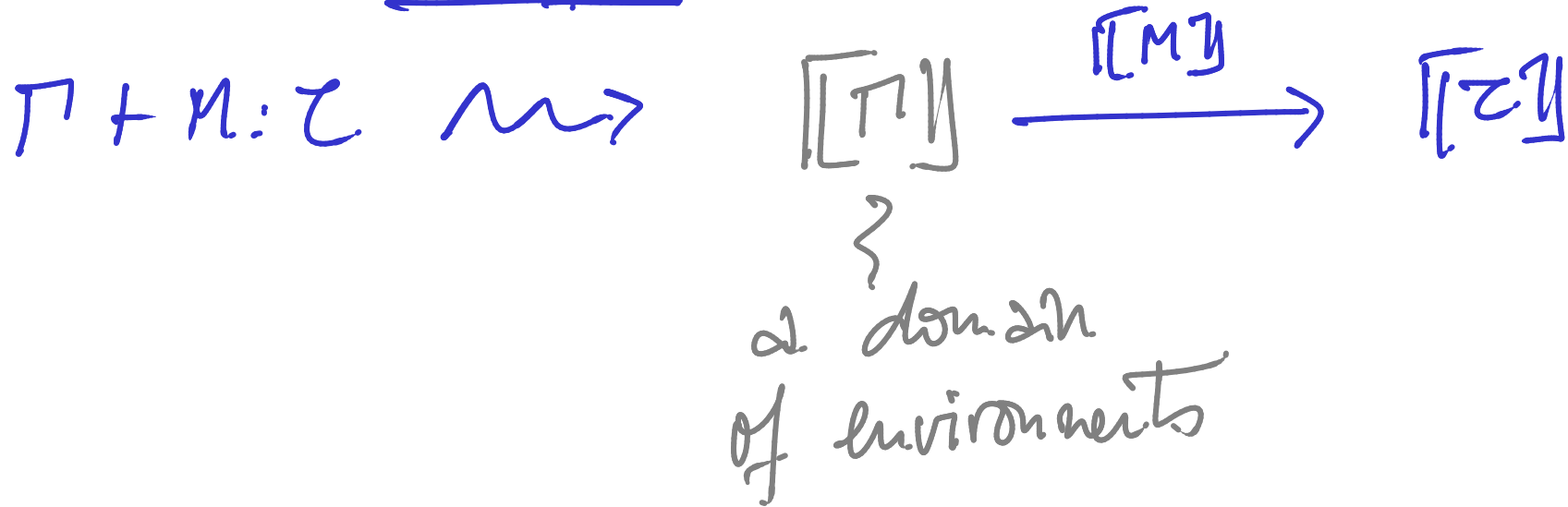
$$\llbracket \text{bool} \rrbracket = \left(\begin{array}{c} \text{true} \quad \text{false} \\ \quad \searrow \quad \swarrow \\ \quad \perp \end{array} \right)$$

$$\llbracket \text{nat} \rrbracket = \left(\begin{array}{c} 0 \quad 1 \quad 2 \quad \dots \quad n \quad \dots \\ \quad \searrow \quad \downarrow \quad \swarrow \quad \nearrow \\ \quad \perp \end{array} \right) \text{ new}$$

$$\llbracket \tau_1 \rightarrow \tau_2 \rrbracket = \left(\llbracket \tau_1 \rrbracket \rightarrow \llbracket \tau_2 \rrbracket \right)$$

PCF denotational semantics — aims

- PCF types $\tau \mapsto$ domains $\llbracket \tau \rrbracket$.
 - Closed PCF terms $M : \tau \mapsto$ elements $\llbracket M \rrbracket \in \llbracket \tau \rrbracket$.
- Denotations of open terms will be continuous functions.
- a term in the empty context.*



The domain associated to a context.

$$\Gamma \equiv (x_1 : \tau_1, x_2 : \tau_2, \dots, x_n : \tau_n)$$

\Downarrow
 $[[\Gamma]]$ a domain of environments

$$\rho \in ([\tau_1] \times [\tau_2] \times \dots \times [\tau_n])$$

$$\parallel$$
$$(d_1, d_2, \dots, d_n) \quad d_i \in [[\tau_i]]$$

An equivalent definition:

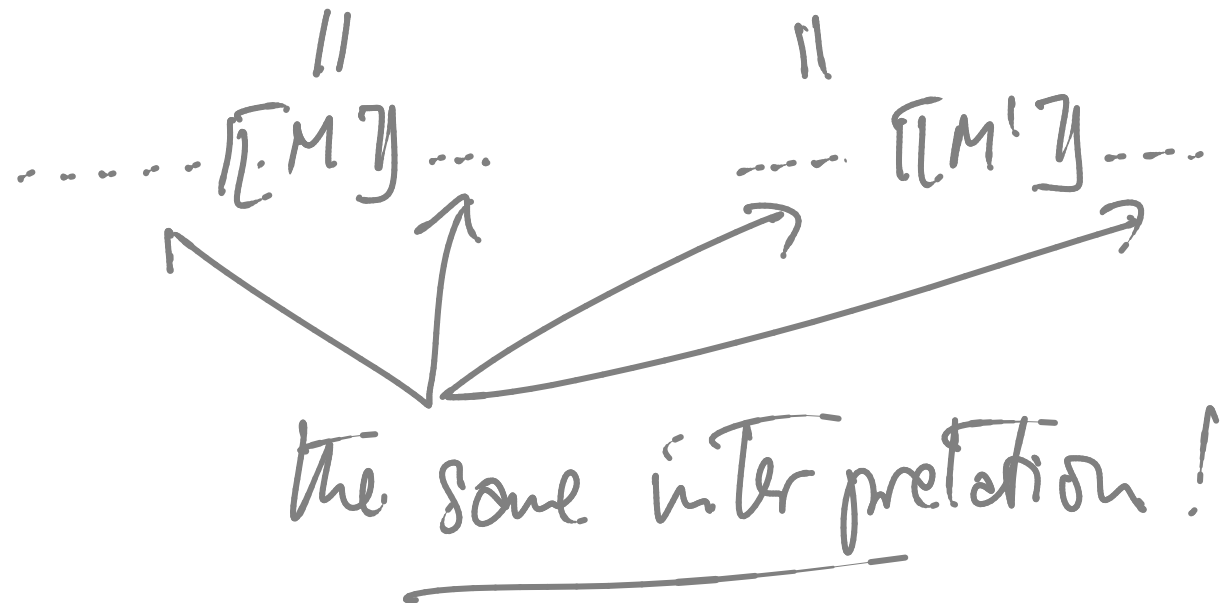
environments are functions that to each x_i assign a $d_i \in [[\tau_i]]$.

PCF denotational semantics — aims

- PCF types $\tau \mapsto$ domains $[[\tau]]$.
- Closed PCF terms $M : \tau \mapsto$ elements $[[M]] \in [[\tau]]$.
Denotations of open terms will be continuous functions.

- **Compositionality.**

In particular: $[[M]] = [[M']] \Rightarrow \forall c \quad [[c[M]]] = [[c[M']]]$.



PCF denotational semantics — aims

- PCF types $\tau \mapsto$ domains $\llbracket \tau \rrbracket$.
- Closed PCF terms $M : \tau \mapsto$ elements $\llbracket M \rrbracket \in \llbracket \tau \rrbracket$.
Denotations of open terms will be continuous functions.
- **Compositionality**.
In particular: $\llbracket M \rrbracket = \llbracket M' \rrbracket \Rightarrow \llbracket \mathcal{C}[M] \rrbracket = \llbracket \mathcal{C}[M'] \rrbracket$.
- **Soundness**.
For any type τ , $M \Downarrow_{\tau} V \Rightarrow \llbracket M \rrbracket = \llbracket V \rrbracket$.
- **Adequacy**.
For $\tau = \mathit{bool}$ or nat , $\llbracket M \rrbracket = \llbracket V \rrbracket \in \llbracket \tau \rrbracket \implies M \Downarrow_{\tau} V$.

Theorem. For all types τ and closed terms $M_1, M_2 \in \text{PCF}_\tau$, if $\llbracket M_1 \rrbracket$ and $\llbracket M_2 \rrbracket$ are equal elements of the domain $\llbracket \tau \rrbracket$, then $M_1 \cong_{\text{ctx}} M_2 : \tau$.

Proof principle:

$$\llbracket M_1 \rrbracket = \llbracket M_2 \rrbracket$$

$$M_1 \cong_{\text{ctx}} M_2$$

Theorem. For all types τ and closed terms $M_1, M_2 \in \text{PCF}_\tau$, if $\llbracket M_1 \rrbracket$ and $\llbracket M_2 \rrbracket$ are equal elements of the domain $\llbracket \tau \rrbracket$, then $M_1 \cong_{\text{ctx}} M_2 : \tau$.

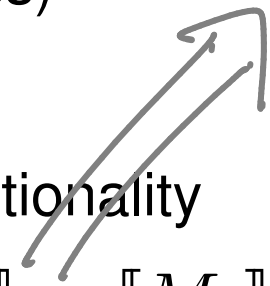
Proof. $\forall V$.

$$\mathcal{C}[M_1] \Downarrow_{\text{nat}} V \Rightarrow \llbracket \mathcal{C}[M_1] \rrbracket = \llbracket V \rrbracket \quad (\text{soundness})$$

$$\Rightarrow \llbracket \mathcal{C}[M_2] \rrbracket = \llbracket V \rrbracket \quad (\text{compositionality on } \llbracket M_1 \rrbracket = \llbracket M_2 \rrbracket)$$

$$\Rightarrow \mathcal{C}[M_2] \Downarrow_{\text{nat}} V \quad (\text{adequacy})$$

$$\llbracket \mathcal{C}[M_1] \rrbracket = \llbracket \mathcal{C}[M_2] \rrbracket$$



and symmetrically. □

Proof principle

To prove

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

it suffices to establish

$$\llbracket M_1 \rrbracket = \llbracket M_2 \rrbracket \text{ in } \llbracket \tau \rrbracket$$

- ? The proof principle is sound, but is it complete? That is, is equality in the denotational model also a necessary condition for contextual equivalence?