Types

 $\tau ::= nat \mid bool \mid \tau \to \tau$

Expressions

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, $\Gamma \vdash M : au$

Lecture 5

PCF

- Γ is a type environment, *i.e.* a finite partial function mapping variables to types (whose domain of definition is denoted dom(Γ))
- *M* is a term
- au is a type.

Notation:

$$\begin{split} M &: \tau \text{ means } M \text{ is closed and } \emptyset \vdash M : \tau \text{ holds.} \\ \mathrm{PCF}_\tau \stackrel{\mathrm{def}}{=} \{ M \mid M : \tau \}. \end{split}$$

PCF typing relation (sample rules)

2

4

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

$$(:_{app}) \quad \frac{\Gamma \vdash M_1 : \tau \to \tau' \quad \Gamma \vdash M_2 : \tau}{\Gamma \vdash M_1 M_2 : \tau'}$$
$$(:_{fix}) \quad \frac{\Gamma \vdash M : \tau \to \tau}{\Gamma \vdash fix(M) : \tau}$$

3

1

Partial recursive functions in PCF

• Primitive recursion.

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

• Minimisation.

$$m(x) =$$
 the least $y \ge 0$ such that $k(x,y) = 0$

takes the form

 $M \Downarrow_{\tau} V$

where

- au is a PCF type
- $M,V \in \mathrm{PCF}_{ au}$ are closed PCF terms of type au

• V is a value,

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

PCF evaluation (sample rules)

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

 $(\Downarrow_{cbn}) \quad \frac{M_1 \Downarrow_{\tau \to \tau'} \mathbf{fn} \, x : \tau \, . \, M_1' \qquad M_1' [M_2/x] \Downarrow_{\tau'} V}{M_1 \, M_2 \Downarrow_{\tau'} V}$

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

Contextual equivalence

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

5

6

Contextual equivalence of PCF terms

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 $V : \gamma$,

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

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

Proof.

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

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

(compositionality on $\llbracket M_1 \rrbracket = \llbracket M_2 \rrbracket$)

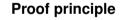
 $\Rightarrow \mathcal{C}[M_2] \Downarrow_{nat} V$ (adequacy)

uequacy)

and symmetrically.

- PCF types $\tau \mapsto$ domains $[\tau]$.
- 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 $\tau, M \Downarrow_{\tau} V \Rightarrow \llbracket M \rrbracket = \llbracket V \rrbracket$.
- Adequacy. For $\tau = bool$ or nat, $\llbracket M \rrbracket = \llbracket V \rrbracket \in \llbracket \tau \rrbracket \Rightarrow M \Downarrow_{\tau} V.$

10



To prove

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

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?

9