# On Randomization in (Higher-Order) Programming

## Part II

***Ugo Dal Lago***




*Escuela de Ciencias Informáticas*, Buenos Aires, July 2023

# A Foundation for Higher-Order Probabilistic Programming

- We are interested in a better understanding of some crucial questions about higher-order probabilistic programs, e.g.:
  - How could we *formalise* and *prove* programs to have certain desireable **properties**, like being terminating or consuming a bounded amount of resources?
  - How could we prove a pair of programs to be **equivalent**, or one to be a **refinement** of the other?

# A Foundation for Higher-Order Probabilistic Programming

- We are interested in a better understanding of some crucial questions about higher-order probabilistic programs, e.g.:
  - How could we *formalise* and *prove* programs to have certain desireable **properties**, like being terminating or consuming a bounded amount of resources?
  - How could we prove a pair of programs to be **equivalent**, or one to be a **refinement** of the other?
- We could in principle answer these questions directly in a programming language like OCAML.

# A Foundation for Higher-Order Probabilistic Programming

- We are interested in a better understanding of some crucial questions about higher-order probabilistic programs, e.g.:
  - How could we *formalise* and *prove* programs to have certain desireable **properties**, like being terminating or consuming a bounded amount of resources?
  - How could we prove a pair of programs to be **equivalent**, or one to be a **refinement** of the other?
- We could in principle answer these questions directly in a programming language like OCAML.
- It is methodologically better to distill some paradigmatic calculi which expose all the essential features, but which are somehow agnostic to many unimportant details.
- We will introduce and study two such calculi, and variations of them:
  - PCF, a calculus for higher-order functional programming.
  - $\mathsf{PCF}_{\oplus}$, a calculus for *randomized* higher-order programming.

# PCF: Types, Terms, Values

**Types** $\quad \tau, \rho \ ::= \ \textsc{Unit} \mid \textsc{Num} \mid \tau \to \rho$

# PCF: Types, Terms, Values

**Types** $\quad \tau, \rho \ ::= \ \textsc{Unit} \mid \textsc{Num} \mid \tau \to \rho$

**Terms** $\quad M, N \ ::= \ V \mid V\ W \mid \texttt{let}\ M = x\ \texttt{in}\ N$
$\qquad\qquad \mid \texttt{if}\ V\ \texttt{then}\ M\ \texttt{else}\ N \mid f_n(V_1, \ldots, V_n)$

**Values** $\quad V, W \ ::= \ \star \mid x \mid r \mid \lambda x.M \mid \texttt{fix}\ x.\lambda y.M$

# PCF: Type Assignment Rules

## Value Typing Rules

$$\frac{}{\Gamma \vdash \star : \textsc{Unit}}\ \mathsf{S} \qquad \frac{}{\Gamma, x : \tau \vdash x : \tau}\ \mathsf{V} \qquad \frac{}{\Gamma \vdash r : \textsc{Num}}\ \mathsf{R}$$

$$\frac{\Gamma, x : \tau \vdash M : \rho}{\Gamma \vdash \lambda x.M : \tau \to \rho}\ \lambda \qquad \frac{\Gamma, x : \tau \to \rho, y : \tau \vdash M : \rho}{\Gamma \vdash \texttt{fix}\, x.\lambda y.M : \tau \to \rho}\ \mathsf{X}$$

## Term Typing Rules

$$\frac{\Gamma \vdash V : \tau \to \rho \quad \Gamma \vdash W : \tau}{\Gamma \vdash V\ W : \rho}\ @ \qquad \frac{\Gamma \vdash M : \tau \quad \Gamma, x : \tau \vdash N : \rho}{\Gamma \vdash \texttt{let}\ M = x\ \texttt{in}\ N : \rho}\ \mathsf{L}$$

$$\frac{\Gamma \vdash V : \textsc{Num} \quad \Gamma \vdash M : \tau \quad \Gamma \vdash N : \tau}{\Gamma \vdash \texttt{if}\ V\ \texttt{then}\ M\ \texttt{else}\ N : \tau}\ \mathsf{I} \qquad \frac{\Gamma \vdash V_1 : \textsc{Num} \quad \cdots \quad \Gamma \vdash V_n : \textsc{Num}}{\Gamma \vdash f_n(V_1, \ldots, V_n) : \textsc{Num}}\ \mathsf{F}$$

# PCF: Type Assignment Rules

**Value Typing Rules**

$$\frac{}{\Gamma \vdash \star : \textsc{Unit}}\ \mathsf{S} \qquad \frac{}{\Gamma, x : \tau \vdash x : \tau}\ \mathsf{V} \qquad \frac{}{\Gamma \vdash r : \textsc{Num}}\ \mathsf{R}$$

$$\frac{\Gamma, x : \tau \vdash M : \rho}{\Gamma \vdash \lambda x.M : \tau \to \rho}\ \lambda \qquad \frac{\Gamma, x : \tau \to \rho, y : \tau \vdash M : \rho}{\Gamma \vdash \mathtt{fix}\ x.\lambda y.M : \tau \to \rho}\ \mathsf{X}$$

**Term Typing Rules**

$$\frac{\Gamma \vdash V : \tau \to \rho \quad \Gamma \vdash W : \tau}{\Gamma \vdash V\ W : \rho}\ @ \qquad \frac{\Gamma \vdash M : \tau \quad \Gamma, x : \tau \vdash N : \rho}{\Gamma \vdash \mathtt{let}\ M = x\ \mathtt{in}\ N : \rho}\ \mathsf{L}$$

$$\frac{\Gamma \vdash V : \textsc{Num} \quad \Gamma \vdash M : \tau \quad \Gamma \vdash N : \tau}{\Gamma \vdash \mathtt{if}\ V\ \mathtt{then}\ M\ \mathtt{else}\ N : \tau}\ \mathsf{I} \qquad \frac{\Gamma \vdash V_1 : \textsc{Num} \quad \cdots \quad \Gamma \vdash V_n : \textsc{Num}}{\Gamma \vdash f_n(V_1, \ldots, V_n) : \textsc{Num}}\ \mathsf{F}$$

- The closed terms of type $\tau$ forms a set $\mathbb{CT}_\tau$.
- Similarly for values and $\mathbb{CV}_\tau$.

**One-Step Reduction**

$$(\lambda x.M)V \to M[V/x]$$

$$(\texttt{fix}\ x.\lambda y.M)V \to M[\texttt{fix}\ x.\lambda y.M/x][V/y]$$

$$\texttt{let}\ V = x\ \texttt{in}\ M \to M[V/x]$$

$$\texttt{if}\ 0\ \texttt{then}\ M\ \texttt{else}\ N \to M$$

$$\texttt{if}\ r\ \texttt{then}\ M\ \texttt{else}\ N \to N \text{ if } r \neq 0$$

$$f(r_1, \cdots, r_n) \to f^*(r_1 \ldots, r_n)$$

$$\frac{M \to L}{\texttt{let}\ M = x\ \texttt{in}\ N \to \texttt{let}\ L = x\ \texttt{in}\ N}$$

**Step-Indexed Reduction**

$$\frac{}{V \Rightarrow_0 V} \qquad \frac{M \to N \quad N \Rightarrow_n V}{M \Rightarrow_{n+1} V}$$

**Step-Indexed Reduction**

$$\frac{}{V \Rightarrow_0 V} \qquad \frac{M \to N \quad N \Rightarrow_n V}{M \Rightarrow_{n+1} V}$$

- We write $M \Downarrow V$ (possibly omitting $V$) to mean that there is a natural number such that $M \Rightarrow_n V$. If no such $n, V$ exist, we write $M \Uparrow$.

**Step-Indexed Reduction**

$$\frac{}{V \Rightarrow_0 V} \qquad \frac{M \to N \quad N \Rightarrow_n V}{M \Rightarrow_{n+1} V}$$

- We write $M \Downarrow V$ (possibly omitting $V$) to mean that there is a natural number such that $M \Rightarrow_n V$. If no such $n, V$ exist, we write $M \Uparrow$.
- The calculus we hve introduced is expressive enough.

**Theorem**

PCF *is Turing-powerful.*

# PCF: Some Easy Meta-Theorems

**Proposition (Progress)**

*For every $M \in \mathbb{CT}_\tau$, either $M$ is a value or there is $N$ with $M \to N$*

# PCF: Some Easy Meta-Theorems

**Proposition (Progress)**

*For every $M \in \mathbb{CT}_\tau$, either $M$ is a value or there is $N$ with $M \to N$*

**Proposition (Subject Reduction)**

*For every $M \in \mathbb{CT}_\tau$ and for every $n \in \mathbb{N}$, if $M \to N$ and $M \Rightarrow_n V$, then $M \in \mathbb{CT}_\tau$ and $V \in \mathbb{CV}_\tau$.*

# PCF: Some Easy Meta-Theorems

**Proposition (Progress)**

*For every $M \in \mathbb{CT}_\tau$, either $M$ is a value or there is $N$ with $M \to N$*

**Proposition (Subject Reduction)**

*For every $M \in \mathbb{CT}_\tau$ and for every $n \in \mathbb{N}$, if $M \to N$ and $M \Rightarrow_n V$, then $M \in \mathbb{CT}_\tau$ and $V \in \mathbb{CV}_\tau$.*

**Fact (Nontermination)**

*Termination does not hold: for every type $\tau$ there is a term $M \in \mathbb{CT}_\tau$ such that $M \Uparrow$.*

# PCF: Program Equivalence

- How could we even *define* a notion of program equivalence? We would like it to be:
  - **Adequate**: Two equivalent terms $M$ and $N$ should *behave the same*, at least as far as termination is concerned: either $M \Downarrow$ and $N \Downarrow$ or $M \Uparrow$ and $N \Uparrow$.
  - **Compatible**: Se could, in any term, substitute a subterm with an equivalent one, obtaining an overall equivalent term.

# PCF: Program Equivalence

- How could we even *define* a notion of program equivalence? We would like it to be:
  - **Adequate**: Two equivalent terms $M$ and $N$ should *behave the same*, at least as far as termination is concerned: either $M \Downarrow$ and $N \Downarrow$ or $M \Uparrow$ and $N \Uparrow$.
  - **Compatible**: Se could, in any term, substitute a subterm with an equivalent one, obtaining an overall equivalent term.
- There is a canonical way to construct such an equivalence, called **observational equivalence**, and spelled out as follows:

**Term Contexts** $C_{\mathbb{T}}, D_{\mathbb{T}} ::= C_{\mathbb{V}} \mid [\cdot] \mid C_{\mathbb{V}}\, V \mid V\, C_{\mathbb{V}} \mid \texttt{if}\ V\ \texttt{then}\ C_{\mathbb{T}}\ \texttt{else}\ D_{\mathbb{T}}$

$\mid \texttt{let}\ C_{\mathbb{T}} = x\ \texttt{in}\ N \mid \texttt{let}\ M = x\ \texttt{in}\ C_{\mathbb{T}}$

**Value Contexts** $C_{\mathbb{V}}, D_{\mathbb{V}} ::= \lambda x.C_{\mathbb{T}} \mid \texttt{fix}\ x.\lambda y.C_{\mathbb{T}}$

# PCF: Program Equivalence

- How could we even *define* a notion of program equivalence? We would like it to be:
  - **Adequate**: Two equivalent terms $M$ and $N$ should *behave the same*, at least as far as termination is concerned: either $M \Downarrow$ and $N \Downarrow$ or $M \Uparrow$ and $N \Uparrow$.
  - **Compatible**: Se could, in any term, substitute a subterm with an equivalent one, obtaining an overall equivalent term.
- There is a canonical way to construct such an equivalence, called **observational equivalence**, and spelled out as follows:

**Term Contexts** $C_{\mathbb{T}}, D_{\mathbb{T}} ::= C_{\mathbb{V}} \mid [\cdot] \mid C_{\mathbb{V}}\, V \mid V\, C_{\mathbb{V}} \mid \texttt{if}\ V\ \texttt{then}\ C_{\mathbb{T}}\ \texttt{else}\ D_{\mathbb{T}}$

$\mid \texttt{let}\ C_{\mathbb{T}} = x\ \texttt{in}\ N \mid \texttt{let}\ M = x\ \texttt{in}\ C_{\mathbb{T}}$

**Value Contexts** $C_{\mathbb{V}}, D_{\mathbb{V}} ::= \lambda x.C_{\mathbb{T}} \mid \texttt{fix}\ x.\lambda y.C_{\mathbb{T}}$

- Given two terms $M, N$ such that $\Gamma \vdash M : \tau$ and $\Gamma \vdash N : \tau$, we say that $M$ and $N$ are *observationally equivalent*, and we write $M \equiv^{\tau}_{\Gamma} N$ iff whenever $\emptyset \vdash C[\Gamma \vdash \cdot : \tau] : \textsc{Unit}$, it holds that $C[M] \Downarrow$ iff $C[N] \Downarrow$.
- A preorder $\leq^{\tau}_{\Gamma}$ can be defined similarly.

# $\mathsf{PCF}_{\oplus}$, Informally

- How could turn PCF into a calculus for randomized computation?

# $\mathsf{PCF}_{\oplus}$, Informally

- How could turn PCF into a calculus for randomized computation?
- The way we follow consists in endowing PCF with **an operator for binary, fair, probabilistic choice** (similarly to what one does in, say, imperative programs).

# $\mathsf{PCF}_{\oplus}$: Types, Terms, Values

$$\textbf{Types} \quad \tau, \rho \ ::= \ \textsc{Unit} \mid \textsc{Num} \mid \tau \to \rho$$

$$\textbf{Terms} \quad M, N \ ::= \ V \mid V\ W \mid \texttt{let}\ M = x\ \texttt{in}\ N \mid M \oplus N \mid \texttt{if}\ V\ \texttt{then}\ M\ \texttt{else}\ N \mid f_n(V_1, \ldots, V_n)$$

$$\textbf{Values} \quad V, W \ ::= \ \star \mid x \mid r \mid \lambda x.M \mid \texttt{fix}\ x.\lambda y.M$$

# $\mathsf{PCF}_{\oplus}$: Type Assignment Rules

## Value Typing Rules

$$\frac{}{\Gamma \vdash \star : \textsc{Unit}}\ \mathsf{S} \qquad \frac{}{\Gamma, x : \tau \vdash x : \tau}\ \mathsf{V} \qquad \frac{}{\Gamma \vdash r : \textsc{Num}}\ \mathsf{R}$$

$$\frac{\Gamma, x : \tau \vdash M : \rho}{\Gamma \vdash \lambda x.M : \tau \to \rho}\ \lambda \qquad \frac{\Gamma, x : \tau \to \rho, y : \tau \vdash M : \rho}{\Gamma \vdash \mathtt{fix}\ x.\lambda y.M : \tau \to \rho}\ \mathsf{X}$$

## Term Typing Rules

$$\frac{\Gamma \vdash V : \tau \to \rho \quad \Gamma \vdash W : \tau}{\Gamma \vdash V\ W : \rho}\ @ \qquad \frac{\Gamma \vdash M : \tau \quad \Gamma, x : \tau \vdash N : \rho}{\Gamma \vdash \mathtt{let}\ M = x\ \mathtt{in}\ N : \rho}\ \mathsf{L} \qquad \frac{\Gamma \vdash M : \tau \quad \Gamma \vdash N : \tau}{\Gamma \vdash M \oplus N : \tau}\ \oplus$$

$$\frac{\Gamma \vdash V : \textsc{Num} \quad \Gamma \vdash M : \tau \quad \Gamma \vdash N : \tau}{\Gamma \vdash \mathtt{if}\ V\ \mathtt{then}\ M\ \mathtt{else}\ N : \tau}\ \mathsf{I} \qquad \frac{\Gamma \vdash V_1 : \textsc{Num} \quad \cdots \quad \Gamma \vdash V_n : \textsc{Num}}{\Gamma \vdash f_n(V_1, \ldots, V_n) : \textsc{Num}}\ \mathsf{F}$$

# $\mathsf{PCF}_{\oplus}$: Type Assignment Rules

## Value Typing Rules

$$\frac{}{\Gamma \vdash \star : \textsc{Unit}}\ \mathsf{S} \qquad \frac{}{\Gamma, x : \tau \vdash x : \tau}\ \mathsf{V} \qquad \frac{}{\Gamma \vdash r : \textsc{Num}}\ \mathsf{R}$$

$$\frac{\Gamma, x : \tau \vdash M : \rho}{\Gamma \vdash \lambda x.M : \tau \to \rho}\ \lambda \qquad \frac{\Gamma, x : \tau \to \rho, y : \tau \vdash M : \rho}{\Gamma \vdash \mathtt{fix}\ x.\lambda y.M : \tau \to \rho}\ \mathsf{X}$$

## Term Typing Rules

$$\frac{\Gamma \vdash V : \tau \to \rho \quad \Gamma \vdash W : \tau}{\Gamma \vdash V\ W : \rho}\ @ \qquad \frac{\Gamma \vdash M : \tau \quad \Gamma, x : \tau \vdash N : \rho}{\Gamma \vdash \mathtt{let}\ M = x\ \mathtt{in}\ N : \rho}\ \mathsf{L} \qquad \frac{\Gamma \vdash M : \tau \quad \Gamma \vdash N : \tau}{\Gamma \vdash M \oplus N : \tau}\ \oplus$$

$$\frac{\Gamma \vdash V : \textsc{Num} \quad \Gamma \vdash M : \tau \quad \Gamma \vdash N : \tau}{\Gamma \vdash \mathtt{if}\ V\ \mathtt{then}\ M\ \mathtt{else}\ N : \tau}\ \mathsf{I} \qquad \frac{\Gamma \vdash V_1 : \textsc{Num} \quad \cdots \quad \Gamma \vdash V_n : \textsc{Num}}{\Gamma \vdash f_n(V_1, \ldots, V_n) : \textsc{Num}}\ \mathsf{F}$$

- The sets $\mathbb{CT}_\tau$ and $\mathbb{CV}_\tau$ are defined as in PCF.

# (Sub)Distributions

- Given any set $X$, a *distribution* on $X$ is a function $\mathcal{D} : X \to \mathbb{R}_{[0,1]}$ such that $\mathcal{D}(x) > 0$ only for denumerably many elements of $X$ and that $\sum_{x \in X} \mathcal{D}(x) \leq 1$.

# (Sub)Distributions

- Given any set $X$, a *distribution* on $X$ is a function $\mathcal{D} : X \to \mathbb{R}_{[0,1]}$ such that $\mathcal{D}(x) > 0$ only for denumerably many elements of $X$ and that $\sum_{x \in X} \mathcal{D}(x) \leq 1$.
- The *support* of a distribution $\mathcal{D}$ on $X$ is the subset $\mathsf{SUPP}(\mathcal{D})$ of $X$ defined as

$$\mathsf{SUPP}(\mathcal{D}) := \{x \in X \mid \mathcal{D}(x) > 0\}$$

# (Sub)Distributions

- Given any set $X$, a *distribution* on $X$ is a function $\mathcal{D} : X \to \mathbb{R}_{[0,1]}$ such that $\mathcal{D}(x) > 0$ only for denumerably many elements of $X$ and that $\sum_{x \in X} \mathcal{D}(x) \leq 1$.
- The *support* of a distribution $\mathcal{D}$ on $X$ is the subset $\mathsf{SUPP}(\mathcal{D})$ of $X$ defined as

$$\mathsf{SUPP}(\mathcal{D}) := \{x \in X \mid \mathcal{D}(x) > 0\}$$

- The set of all distributions over $X$ is indicated as $\mathbf{D}(X)$.

# (Sub)Distributions

- Given any set $X$, a *distribution* on $X$ is a function $\mathcal{D} : X \to \mathbb{R}_{[0,1]}$ such that $\mathcal{D}(x) > 0$ only for denumerably many elements of $X$ and that $\sum_{x \in X} \mathcal{D}(x) \leq 1$.
- The *support* of a distribution $\mathcal{D}$ on $X$ is the subset $\mathsf{SUPP}(\mathcal{D})$ of $X$ defined as

$$\mathsf{SUPP}(\mathcal{D}) := \{x \in X \mid \mathcal{D}(x) > 0\}$$

- The set of all distributions over $X$ is indicated as $\mathbf{D}(X)$.
- We indicate the distribution assigning probability 1 to the element $x \in X$ and 0 to any other element of $X$ as $\delta(x)$.

# (Sub)Distributions

- Given any set $X$, a *distribution* on $X$ is a function $\mathcal{D} : X \to \mathbb{R}_{[0,1]}$ such that $\mathcal{D}(x) > 0$ only for denumerably many elements of $X$ and that $\sum_{x \in X} \mathcal{D}(x) \leq 1$.
- The *support* of a distribution $\mathcal{D}$ on $X$ is the subset $\mathsf{SUPP}(\mathcal{D})$ of $X$ defined as

$$\mathsf{SUPP}(\mathcal{D}) := \{x \in X \mid \mathcal{D}(x) > 0\}$$

- The set of all distributions over $X$ is indicated as $\mathbf{D}(X)$.
- We indicate the distribution assigning probability 1 to the element $x \in X$ and 0 to any other element of $X$ as $\delta(x)$.
- Whenever $I$ is an index set, the expression $\{x_i : p_i\}_{i \in I}$ stands for the distribution assigning probability $p_i$ to $x_i$, for every $i \in I$.

# (Sub)Distributions

- Given any set $X$, a *distribution* on $X$ is a function $\mathcal{D} : X \to \mathbb{R}_{[0,1]}$ such that $\mathcal{D}(x) > 0$ only for denumerably many elements of $X$ and that $\sum_{x \in X} \mathcal{D}(x) \leq 1$.
- The *support* of a distribution $\mathcal{D}$ on $X$ is the subset $\mathsf{SUPP}(\mathcal{D})$ of $X$ defined as

$$\mathsf{SUPP}(\mathcal{D}) := \{x \in X \mid \mathcal{D}(x) > 0\}$$

- The set of all distributions over $X$ is indicated as $\mathbf{D}(X)$.
- We indicate the distribution assigning probability 1 to the element $x \in X$ and 0 to any other element of $X$ as $\delta(x)$.
- Whenever $I$ is an index set, the expression $\{x_i : p_i\}_{i \in I}$ stands for the distribution assigning probability $p_i$ to $x_i$, for every $i \in I$.
- Given a distribution $\mathcal{D}$ on $X$, its *sum* $\sum \mathcal{D}$ is simply $\sum_{x \in X} \mathcal{D}(x)$.

# $\mathsf{PCF}_{\oplus}$: Reduction

**One-Step Reduction**

$$
\begin{aligned}
(\lambda x.M)V &\to \delta(M[V/x]) \\
(\texttt{fix}\ x.\lambda y.M)W &\to \delta(M[\texttt{fix}\ x.\lambda y.M/x][W/y]) \\
\texttt{let}\ V = x\ \texttt{in}\ M &\to \delta(M[V/x]) \\
\texttt{if}\ 0\ \texttt{then}\ M\ \texttt{else}\ N &\to \delta(M) \\
\texttt{if}\ r\ \texttt{then}\ M\ \texttt{else}\ N &\to \delta(N)\ \text{if}\ r \neq 0 \\
M \oplus N &\to \left\{ M : \frac{1}{2}, N : \frac{1}{2} \right\} \\
f(r_1, \cdots, r_n) &\to \delta(f^*(r_1 \ldots, r_n))
\end{aligned}
$$

$$
\frac{M \to \{L_i : p_i\}_{i \in I}}{\texttt{let}\ M = x\ \texttt{in}\ N \to \{\texttt{let}\ L_i = x\ \texttt{in}\ N : p_i\}_{i \in I}}
$$

# $\mathsf{PCF}_{\oplus}$: Reduction

**Step-Indexed Reduction**

$$\frac{M \to \mathcal{D}}{M \Rightarrow_0 \emptyset} \qquad \frac{}{V \Rightarrow_0 \delta(V)} \qquad \frac{}{V \Rightarrow_{n+1} \emptyset} \qquad \frac{M \to \mathcal{D} \quad \forall N \in \mathsf{SUPP}(\mathcal{D}).N \Rightarrow_n \mathcal{E}_N}{M \Rightarrow_{n+1} \sum_{N \in \mathsf{SUPP}(\mathcal{D})} \mathcal{D}(N) \cdot \mathcal{E}_N}$$

# $\mathsf{PCF}_{\oplus}$: Some Easy Meta-Theorems

**Lemma**

*If* $M \Rightarrow_n \mathcal{D}$, *then* $\mathsf{SUPP}(\mathcal{D})$ *is a finite set.*

# $\mathsf{PCF}_{\oplus}$: Some Easy Meta-Theorems

**Lemma**

*If $M \Rightarrow_n \mathcal{D}$, then $\mathsf{SUPP}(\mathcal{D})$ is a finite set.*

**Proposition (Progress)**

*For every $M \in \mathbb{CT}_\tau$, either $M$ is a value or there is $\mathcal{D}$ with $M \to \mathcal{D}$*

# $\mathsf{PCF}_{\oplus}$: Some Easy Meta-Theorems

**Lemma**

*If $M \Rightarrow_n \mathcal{D}$, then $\mathsf{SUPP}(\mathcal{D})$ is a finite set.*

**Proposition (Progress)**

*For every $M \in \mathbb{CT}_\tau$, either $M$ is a value or there is $\mathcal{D}$ with $M \to \mathcal{D}$*

**Proposition (Subject Reduction)**

*For every $M \in \mathbb{CT}_\tau$ and for every $n \in \mathbb{N}$, if $M \to \mathcal{D}$ and $M \Rightarrow_n \mathcal{E}$, then $\mathcal{D} \in \mathbf{D}(\mathbb{CT}_\tau)$ and $\mathcal{E} \in \mathbf{D}(\mathbb{CV}_\tau)$.*

# $\mathsf{PCF}_{\oplus}$: Some Easy Meta-Theorems

## Lemma

*If $M \Rightarrow_n \mathcal{D}$, then $\mathsf{SUPP}(\mathcal{D})$ is a finite set.*

## Proposition (Progress)

*For every $M \in \mathbb{CT}_\tau$, either $M$ is a value or there is $\mathcal{D}$ with $M \to \mathcal{D}$*

## Proposition (Subject Reduction)

*For every $M \in \mathbb{CT}_\tau$ and for every $n \in \mathbb{N}$, if $M \to \mathcal{D}$ and $M \Rightarrow_n \mathcal{E}$, then $\mathcal{D} \in \mathbf{D}(\mathbb{CT}_\tau)$ and $\mathcal{E} \in \mathbf{D}(\mathbb{CV}_\tau)$.*

## Corollary

*For every $M \in \mathbb{CT}_\tau$ and for every $n \in \mathbb{N}$, there is exactly one distribution $\mathcal{D}_n$ such that $M \Rightarrow_n \mathcal{D}_n$. We will write $\langle M \rangle_n$ for such a distribution.*

# $\mathsf{PCF}_{\oplus}$: The Operational Semantics of a Term

- Given two distributions $\mathcal{D}, \mathcal{E} \in \mathbf{D}(X)$, we write $\mathcal{D} \leq \mathcal{E}$ iff $\mathcal{D}(x) \leq \mathcal{E}(x)$ for every $x \in X$. This relation endows $\mathbf{D}(X)$ with the structure of a partial order, which is actually an $\omega\mathbf{CPO}$:
- Given a closed term $M \in \mathbb{CT}_\tau$, the **operational semantics** of $M$ is defined to be the distribution $\langle M \rangle \in \mathbb{CV}_\tau$ defined as $\sum_{n \in \mathbb{N}} \langle M \rangle_n$.

# $\mathsf{PCF}_{\oplus}$: The Operational Semantics of a Term

- Given two distributions $\mathcal{D}, \mathcal{E} \in \mathbf{D}(X)$, we write $\mathcal{D} \leq \mathcal{E}$ iff $\mathcal{D}(x) \leq \mathcal{E}(x)$ for every $x \in X$. This relation endows $\mathbf{D}(X)$ with the structure of a partial order, which is actually an $\omega$**CPO**:
- Given a closed term $M \in \mathbb{CT}_{\tau}$, the **operational semantics** of $M$ is defined to be the distribution $\langle M \rangle \in \mathbb{CV}_{\tau}$ defined as $\sum_{n \in \mathbb{N}} \langle M \rangle_n$.

**Term Contexts**

$$C_{\mathbb{T}}, D_{\mathbb{T}} ::= C_{\mathbb{V}} \mid [\cdot] \mid C_{\mathbb{V}}\, V \mid V\, C_{\mathbb{V}} \mid \mathtt{let}\ C_{\mathbb{T}} = x\ \mathtt{in}\ N \mid \mathtt{let}\ M = x\ \mathtt{in}\ C_{\mathbb{T}} \mid C_{\mathbb{T}} \oplus D_{\mathbb{T}} \mid \mathtt{if}\ V\ \mathtt{then}\ C_{\mathbb{T}}\ \mathtt{else}\ D_{\mathbb{T}}$$

**Value Contexts**

$$C_{\mathbb{V}}, D_{\mathbb{V}} ::= \lambda x.C_{\mathbb{T}} \mid \mathtt{fix}\ x.\lambda y.C_{\mathbb{T}}$$

# $\mathsf{PCF}_{\oplus}$: The Operational Semantics of a Term

- Given two distributions $\mathcal{D}, \mathcal{E} \in \mathbf{D}(X)$, we write $\mathcal{D} \leq \mathcal{E}$ iff $\mathcal{D}(x) \leq \mathcal{E}(x)$ for every $x \in X$. This relation endows $\mathbf{D}(X)$ with the structure of a partial order, which is actually an $\omega\mathbf{CPO}$:
- Given a closed term $M \in \mathbb{CT}_\tau$, the **operational semantics** of $M$ is defined to be the distribution $\langle M \rangle \in \mathbb{CV}_\tau$ defined as $\sum_{n \in \mathbb{N}} \langle M \rangle_n$.

**Term Contexts**

$$C_{\mathbb{T}}, D_{\mathbb{T}} ::= C_{\mathbb{V}} \mid [\cdot] \mid C_{\mathbb{V}}\, V \mid V\, C_{\mathbb{V}} \mid \mathtt{let}\ C_{\mathbb{T}} = x\ \mathtt{in}\ N \mid \mathtt{let}\ M = x\ \mathtt{in}\ C_{\mathbb{T}} \mid C_{\mathbb{T}} \oplus D_{\mathbb{T}} \mid \mathtt{if}\ V\ \mathtt{then}\ C_{\mathbb{T}}\ \mathtt{else}\ D_{\mathbb{T}}$$

**Value Contexts**

$$C_{\mathbb{V}}, D_{\mathbb{V}} ::= \lambda x.C_{\mathbb{T}} \mid \mathtt{fix}\ x.\lambda y.C_{\mathbb{T}}$$

- Given two terms $M, N$ such that $\Gamma \vdash M : \tau$ and $\Gamma \vdash N : \tau$, we say that $M$ and $N$ are $(\Gamma, \tau)$-*equivalent*, and we write $M \equiv^{\tau}_{\Gamma} N$ iff whenever $\emptyset \vdash C[\Gamma \vdash \cdot : \tau] : \textsc{Unit}$, it holds that $\sum \langle C[M] \rangle = \sum \langle C[N] \rangle$.

# $\mathsf{PCF}_{\oplus}$: from Equivalences to Metrics?

- The definition of contexual equivalence asks that $\sum\langle C[M]\rangle = \sum\langle C[N]\rangle$ for every context $C$.
- But what if $\sum\langle C[M]\rangle$ and $\sum\langle C[N]\rangle$ are very close, without being really equal to each other?
- It makes sense to generalize contextual equivalence to a notion of **distance**:

$$\delta^{\Gamma,\tau}(M,N) = \sup_{\emptyset\vdash C[\Gamma\vdash\cdot:\tau]:\textsc{Unit}} \left|\sum\langle C[M]\rangle - \sum\langle C[N]\rangle\right|.$$

- For every $\Gamma, \tau$, $\delta^{\Gamma,\tau}$ is indeed a *pseudo-metric*:

$$\begin{aligned}\delta^{\Gamma,\tau}(M,M) &= 0\\ \delta^{\Gamma,\tau}(M,N) &= \delta^{\Gamma,\tau}(N,M)\\ \delta^{\Gamma,\tau}(M,L) &\leq \delta^{\Gamma,\tau}(M,N) + \delta^{\Gamma,\tau}(N,L)\end{aligned}$$

# Termination in a Probabilistic Setting

- Let $M$ be any closed term. We say that $M$ is **almost surely terminating** if $\sum\langle M\rangle = 1$, namely if its probability of convergence is 1.

# Termination in a Probabilistic Setting

- Let $M$ be any closed term. We say that $M$ is **almost surely terminating** if $\sum\langle M\rangle = 1$, namely if its probability of convergence is 1.
- **Example**:

$$GEO := (\texttt{fix}\ f.\lambda x.x \oplus (\texttt{let}\ succ_1(x) = y\ \texttt{in}\ f\ y))0$$

# Termination in a Probabilistic Setting

- Let $M$ be any closed term. We say that $M$ is **almost surely terminating** if $\sum\langle M\rangle = 1$, namely if its probability of convergence is 1.
- **Example**:

$$GEO := (\texttt{fix}\ f.\lambda x.x \oplus (\texttt{let}\ succ_1(x) = y\ \texttt{in}\ f\ y))0$$

- The **expected evaluation length** of any closed term as follows:

$$ExLen(M) := \sum_{m=0}^{\infty}\left(1 - \sum_{n=0}^{m}\sum\langle M\rangle_n\right)$$

Let $M$ be any closed term. We say that $M$ is **positively almost surely terminating** if $ExLen(M) < +\infty$.

# Termination in a Probabilistic Setting

- Let $M$ be any closed term. We say that $M$ is **almost surely terminating** if $\sum\langle M\rangle = 1$, namely if its probability of convergence is 1.
- **Example**:

$$GEO := (\texttt{fix}\ f.\lambda x.x \oplus (\texttt{let}\ succ_1(x) = y\ \texttt{in}\ f\ y))0$$

- The **expected evaluation length** of any closed term as follows:

$$ExLen(M) := \sum_{m=0}^{\infty}\left(1 - \sum_{n=0}^{m}\sum\langle M\rangle_n\right)$$

Let $M$ be any closed term. We say that $M$ is **positively almost surely terminating** if $ExLen(M) < +\infty$.

**Lemma**

*Every positively almost-surely terminating term is almost-surely terminating.*

# Variations on PCF and $\mathsf{PCF}_\oplus$: Going Untyped

- We can turn $\mathsf{PCF}_\oplus$ into a **pure and untyped**, rather than typed, calculus:
  - **Terms**: $M ::= x \mid \lambda M. \mid MM \mid M \oplus M$;
  - **Values**: $V ::= \lambda M.$;
  - **One-Step Reduction**:

$$(\lambda x.M)V \to \delta(M[V/x]) \qquad M \oplus N \to \left\{ M : \frac{1}{2}, N : \frac{1}{2} \right\}$$

$$\frac{M \to \{L_i : p_i\}_{i \in I}}{MN \to \{L_i N : p_i\}_{i \in I}} \qquad \frac{M \to \{L_i : p_i\}_{i \in I}}{VM \to \{V L_i : p_i\}_{i \in I}}$$

  - The obtained calculus will be referred to as $\Lambda_\oplus$.

# Variations on PCF and $\mathsf{PCF}_\oplus$: Going Untyped

- We can turn $\mathsf{PCF}_\oplus$ into a **pure and untyped**, rather than typed, calculus:
  - **Terms**: $M ::= x \mid \lambda M. \mid MM \mid M \oplus M$;
  - **Values**: $V ::= \lambda M.$;
  - **One-Step Reduction**:

$$(\lambda x.M)V \to \delta(M[V/x]) \qquad M \oplus N \to \left\{ M : \frac{1}{2}, N : \frac{1}{2} \right\}$$

$$\frac{M \to \{L_i : p_i\}_{i \in I}}{MN \to \{L_i N : p_i\}_{i \in I}} \qquad \frac{M \to \{L_i : p_i\}_{i \in I}}{VM \to \{V L_i : p_i\}_{i \in I}}$$

  - The obtained calculus will be referred to as $\Lambda_\oplus$.
- Similarly, one can form $\Lambda$, starting from PCF.

# Variations on PCF and $\mathsf{PCF}_{\oplus}$: Acting on the Reduction Strategy

- All the calculi introduced so far evaluate terms in the so-called *call-by-value* regime: when performing a substitution, the substituted term is a *value*, e.g.

$$(\lambda x.M)V \to M[V/x]$$

# Variations on PCF and $\mathsf{PCF}_{\oplus}$: Acting on the Reduction Strategy

- All the calculi introduced so far evaluate terms in the so-called *call-by-value* regime: when performing a substitution, the substituted term is a *value*, e.g.

$$(\lambda x.M)V \to M[V/x]$$

- Nothing prevents us to go **call-by-name**, instead.

# Variations on PCF and $\mathsf{PCF}_{\oplus}$: Acting on the Reduction Strategy

- All the calculi introduced so far evaluate terms in the so-called *call-by-value* regime: when performing a substitution, the substituted term is a *value*, e.g.

$$(\lambda x.M)V \to M[V/x]$$

- Nothing prevents us to go **call-by-name**, instead.
- One could, e.g., define $\Lambda_{\oplus}^{\mathsf{name}}$ by leaving the terms as they are, and modifying one-step reduction as follows:

$$(\lambda x.M)N \to \delta(M[N/x]) \qquad M \oplus N \to \left\{M : \frac{1}{2}, N : \frac{1}{2}\right\}$$

$$\frac{M \to \{L_i : p_i\}_{i \in I}}{MN \to \{L_i N : p_i\}_{i \in I}}$$

# Variations on PCF and $\mathsf{PCF}_\oplus$: Acting on the Reduction Strategy

- All the calculi introduced so far evaluate terms in the so-called *call-by-value* regime: when performing a substitution, the substituted term is a *value*, e.g.

$$(\lambda x.M)V \to M[V/x]$$

- Nothing prevents us to go **call-by-name**, instead.
- One could, e.g., define $\Lambda_\oplus^{\mathsf{name}}$ by leaving the terms as they are, and modifying one-step reduction as follows:

$$(\lambda x.M)N \to \delta(M[N/x]) \qquad M \oplus N \to \left\{M : \frac{1}{2}, N : \frac{1}{2}\right\}$$

$$\frac{M \to \{L_i : p_i\}_{i \in I}}{MN \to \{L_i N : p_i\}_{i \in I}}$$

- But **beware**: while a form of confluence can be proved in PCF and $\Lambda$, the same cannot be said for $\mathsf{PCF}_\oplus$ and $\Lambda_\oplus$.

# Variations on PCF and $\mathsf{PCF}_{\oplus}$: Dropping Recursion

- If we drop recursion from PCF, one obtains a calculus called ST, often referred to as the **simply-typed $\lambda$-calculus**.
  - We just have to remove $\texttt{fix}\, x.\lambda y.M$ from the grammar of values.

# Variations on PCF and $\mathsf{PCF}_{\oplus}$: Dropping Recursion

- If we drop recursion from PCF, one obtains a calculus called ST, often referred to as the **simply-typed $\lambda$-calculus**.
  - We just have to remove $\mathtt{fix}\, x.\lambda y.M$ from the grammar of values.
- Nothing in the metatheory changes. Actually we can prove something that we could not prove in PCF:

**Proposition**

ST *is terminating, i.e. for every* $M \in \mathbb{CT}_\tau$ *there are a natural number* $n$ *and a value* $V$ *such that* $M \Rightarrow_n V$.

# Variations on PCF and $\mathsf{PCF}_{\oplus}$: Dropping Recursion

- If we drop recursion from PCF, one obtains a calculus called ST, often referred to as the **simply-typed $\lambda$-calculus**.
  - We just have to remove $\texttt{fix}\, x.\lambda y.M$ from the grammar of values.
- Nothing in the metatheory changes. Actually we can prove something that we could not prove in PCF:

**Proposition**

ST *is terminating, i.e. for every* $M \in \mathbb{CT}_\tau$ *there are a natural number* $n$ *and a value* $V$ *such that* $M \Rightarrow_n V$.

- On the other hand, the expressive power of the calculus is **very poor**: there is no recursion-theoretic or complexity class that ST can capture.

# Variations on PCF and $\mathsf{PCF}_\oplus$: Dropping Recursion

- If we drop recursion from PCF, one obtains a calculus called ST, often referred to as the **simply-typed $\lambda$-calculus**.
  - We just have to remove $\mathtt{fix}\, x.\lambda y.M$ from the grammar of values.
- Nothing in the metatheory changes. Actually we can prove something that we could not prove in PCF:

**Proposition**

ST *is terminating, i.e. for every* $M \in \mathbb{CT}_\tau$ *there are a natural number* $n$ *and a value* $V$ *such that* $M \Rightarrow_n V$.

- On the other hand, the expressive power of the calculus is **very poor**: there is no recursion-theoretic or complexity class that ST can capture.
- We can play the same game with $\mathsf{PCF}_\oplus$, obtaining ST, with very similar outcomes.

# Thank You!

Questions?