Lecture Notes
Part III of MPRI 2 – 02
2021 - 2021

Michele Pagani
pagani@irif.fr

January 28, 2022

Contents

List of exercises

These notes are a continuation of the lecture notes by Thomas Ehrhard, `https://www.irif.fr/~ehrhard/pub/mpri-2020-2021.pdf`.

1 The Probabilistic Extension pPCF of PCF

1.1 The Syntax of pPCF

Figure 1 sketches the probabilistic extension of PCF, written pPCF. Let Γ be a typing context and A be a type, we denote by Λ_{Γ}^{A} the set of all terms M such that $\Gamma \vdash M : A$. In the case

$$A, B, \ldots := \iota \mid A \Rightarrow B$$

(a) The grammar of types, ι is the *ground type* of natural numbers.

$$\begin{aligned} M, N, \ldots := {} & \underline{n} \mid x \mid \mathsf{succ}(M) \mid \mathsf{if}(M, P, z \cdot R) \mid \lambda x^{A}\, M \mid (M)\, N \\ & \mid \mathsf{fix}(M) \mid \mathsf{coin} \end{aligned}$$

(b) The grammar of terms, with $n \in \mathbb{N}$, $p \in [0,1]$, and x, $y \ldots$ variables.

$$\frac{}{\Gamma \vdash \underline{n} : \iota} \qquad \frac{}{\Gamma, x : A \vdash x : A} \qquad \frac{}{\Gamma \vdash \mathsf{coin} : \iota} \qquad \frac{\Gamma \vdash M : \iota}{\Gamma \vdash \mathsf{succ}(M) : \iota}$$

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x^{A}\, M : A \Rightarrow B} \qquad \frac{\Gamma \vdash M : A \Rightarrow B \qquad \Gamma \vdash N : A}{\Gamma \vdash (M)\, N : B}$$

$$\frac{\Gamma \vdash M : A \Rightarrow A}{\Gamma \vdash \mathsf{fix}(M) : A} \qquad \frac{\Gamma \vdash M : \iota \qquad \Gamma \vdash P : A \qquad \Gamma, z : \iota \vdash R : A}{\Gamma \vdash \mathsf{if}(M, P, z \cdot R) : A}$$

(c) The typing rules, with $\Gamma = y_1 : A_1, \ldots, y_k : A_k$ a typing context, $k \in \mathbb{N}$ and $y_i \neq y_j$ whenever $i \neq j$.

$$\frac{}{\left(\lambda x^{A}\, M\right) N \xrightarrow{1} M\,[N/x]} \qquad \frac{}{\mathsf{fix}(M) \xrightarrow{1} (M)\, \mathsf{fix}(M)}$$

$$\frac{}{\mathsf{succ}(\underline{n}) \xrightarrow{1} \underline{n+1}} \qquad \frac{}{\mathsf{if}(\underline{0}, P, z \cdot R) \xrightarrow{1} P} \qquad \frac{}{\mathsf{if}(\underline{n+1}, P, z \cdot R) \xrightarrow{1} R\,[\underline{n}/z]}$$

$$\frac{}{\mathsf{coin} \xrightarrow{1/2} \underline{0}} \qquad \frac{}{\mathsf{coin} \xrightarrow{1/2} \underline{1}}$$

$$\frac{M \xrightarrow{p} M'}{(M)\, N \xrightarrow{p} (M')\, N} \qquad \frac{M \xrightarrow{p} M'}{\mathsf{succ}(M) \xrightarrow{p} \mathsf{succ}(M')}$$

$$\frac{M \xrightarrow{p} M'}{\mathsf{if}(M, P, z \cdot R) \xrightarrow{p} \mathsf{if}(M', P, z \cdot R)}$$

(d) The reduction relation $M \xrightarrow{p} M'$, with $p \in [0,1]$, M, M' pPCF terms.

Figure 1: Résumé of pPCF.

where Γ is empty, and so the elements of Λ^A_Γ are closed, we use Λ^A_0 to denote that set. A *program* will be a closed term of pPCF of ground type ι, i.e. an element of Λ^ι_0

By a simple inspection of the typing rules, the reader can check the following.

Remark: Let M be a term and Γ be a typing context. There is at most one type A such that $\Gamma \vdash M : A$.

Exercise 1. Give an example of expression M generated by the grammar of Figure 1b, such that M cannot be typed by the rules of Figure 1c. Can you find an M using only abstractions, applications and variables? and another M using only only variables, numerals, coin, branchings and succ(M)?

The *reduction relation* for evaluating pPCF terms is given in Figure 1d. In the β-rule (topmost leftmost rule of Figure 1d), the term $M\,[N/x]$ stands for M where the variable x is substituted with the term N, avoiding the capture of the free variables in N. If $M \xrightarrow{p} M'$ is the conclusion of one axiom rule (i.e. one of the rules in the first three lines of Figure 1d), then we call M the *redex* of the reduction, M' its *contractum* and p the *probability* to happen. This reduction is called *weak-head reduction* (or simply weak reduction) since it always reduces the leftmost outermost redex and never reduces redexes under abstractions. We say that M is *weak-normal*, or a *value*, if there is no reduction $M \xrightarrow{p} M'$.

Lemma 1 (Substitution) *Assume* $\Gamma, x : A \vdash M : B$ *and* $\Gamma \vdash N : A$, *then* $\Gamma \vdash M\,[N/x] : B$.

Exercise 2. Prove Lemma 1.

Proposition 2 (Subject reduction) *Assume* $M \xrightarrow{p} M'$. *If* $\Gamma \vdash M : A$, *then* $\Gamma \vdash M' : A$.

Exercise 3. Give a proof of Proposition 2.

Exercise 4. Give a counterexample to the inverse of subjection reduction, called subject expansion: give an example of reduction $M \xrightarrow{p} M'$ and of type A, environment Γ, such that $\Gamma \vdash M' : A$ but it is false that $\Gamma \vdash M : A$.

Exercise 5. Characterise the set of closed values of pPCF.

A *reduction sequence* from a term M to a term M' is a finite sequence $\varphi = (M_i)_{i=0}^k$ such that $M_0 = M$, $M_k = M'$ and for every $i < k$, $M_i \xrightarrow{p_i} M_{i+1}$ for some probability $p_i \in [0,1]$. By inspection of the rules in Figure 1d, the reader can check that the probability p_i in $M_i \xrightarrow{p_i} M_{i+1}$ is unique, given M_i and M_{i+1}. The *length of* φ is k and the *probability* $\mathsf{p}(\varphi)$ *of* φ is the product $\prod_{i=0}^{k-1} p_i$.

We say that a term M *deterministically reduces* to a value V, written $M \to_{\mathsf{d}}{}^* V$, if there is a reduction sequence φ from M to V of probability 1. Notice that such a reduction is unique, i.e. any other reduction sequence starting from M is a prefix of φ. The following exercise exploits the deterministic fragment of pPCF.

Exercise 6. Define terms representing the following functions:

1. the predecessor function, i.e. a term pred such that:

$$(\mathsf{pred})\,\underline{n} \to_{\mathsf{d}}{}^* \begin{cases} \underline{0} & \text{if } n = 0 \\ \underline{n-1} & \text{if } n > 0 \end{cases}$$

2. the addition function, i.e. a term add such that:

$$(\mathsf{add})\,\underline{n}\,\underline{m} \to_{\mathsf{d}}{}^* \underline{n+m}$$

3. the exponential function, i.e. a term $\mathsf{exp_2}$ such that:

$$(\mathsf{exp_2})\,\underline{n} \to_{\mathsf{d}}{}^* \underline{2^n}$$

4. the comparison function, i.e. a term cmp such that:

$$(\mathsf{cmp})\,\underline{n}\,\underline{m} \to_{\mathsf{d}}{}^* \begin{cases} \underline{0} & \text{if } n \leq m \\ \underline{1} & \text{if } n > m \end{cases}$$

The constructor coin is the stochastic primitive of pPCF, leading to different outcomes. Given a term M and a value V, we define the set of different reduction sequences from M to V as:

$$\mathsf{Path}^{\leq n}(M, V) = \{\varphi \,|\, \varphi \text{ reduction sequence of length } \textit{at most } n \text{ from } M \text{ to } V\} \tag{1}$$

$$\mathsf{Path}(M, V) = \bigcup_{n \in \mathbb{N}} \mathsf{Path}^{\leq n}(M, V) \tag{2}$$

The quantity $\sum_{\varphi \in \mathsf{Path}(M,V)} \mathsf{p}(\varphi)$ defines the probability that M reduces to V. We will formalise this idea in Section 1.3 by representing the reduction relation as a discrete time Markov chain whose states are terms, weak-normal terms being stationary. Before that, let us recall some notions we need in the sequel.

1.2 Compendium of Markov Chains

Let S be a countable set and let $R \in [0, 1]^{S \times S}$ be a matrix with S-indexed rows and columns. One says that R is *sub-stochastic* if $\forall i \in S, \sum_{j \in S} R_{i,j} \leq 1$, we call R *stochastic* whenever the previous sum is equal to 1 for all i. Given two such matrices R and T, their *product* RT is given by

$$\forall (i, j) \in S^2, (RT)_{i,j} = \sum_{k \in I} R_{i,k} T_{k,j}$$

which is also a (sub-)stochastic matrix. Given $n \in \mathbb{N}$, we denote by R^n the n-fold product of R, which is the diagonal matrix if $n = 0$.

A stochastic matrix represents a one-step evolution of a discrete-time Markov process. A typical example is a random-walk, as the following one.

Example. Let $S = \mathbb{N}$ and consider the following matrix over $[0, 1]^{S \times S}$:

$$W_{i,j} = \begin{cases} 1 & \text{if } i = j = 0, \\ \frac{1}{2} & \text{if } i > 0 \text{ and } (j = i - 1 \text{ or } j = i + 1), \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Notice that W is stochastic. In fact, W defines a Markov process describing a particle travelling over $\mathbb{N}$: once the particle reaches 0, it will stay there, otherwise it will move $+1$ or -1 with equal probability $\frac{1}{2}$. The matrix W^n will then describe the state of the particle after n iterations.

Given a stochastic matrix R over S, the set of *stationary states* of R is defined by:

$$S_1^R = \{i \in S \mid R_{i,i} = 1\} \tag{4}$$

so that if $i \in S_1^R$ and $R_{i,j} \neq 0$ then $i = j$.

Let $(i, j) \in S \times S_1^R$. Then the n-indexed sequence $(R^n)_{i,j} \in [0, 1]$ is monotone. Indeed, for all n we have

$$(R^{n+1})_{i,j} = \sum_{k \in S} (R^n)_{i,k} R_{k,j} \geq (R^n)_{i,j} R_{j,j} = (R^n)_{i,j}$$

So we can define a matrix $R^\infty \in [0, 1]^{S \times S}$ as follows

$$(R^\infty)_{i,j} = \begin{cases} \sup_{n \in \mathbb{N}} (R^n)_{i,j} & \text{if } (i, j) \in S \times S_1^R \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

The matrix S^∞ is a sub-stochastic matrix because, given $i \in I$

$$\begin{aligned} \sum_{j \in S} (R^\infty)_{i,j} &= \sum_{j \in S_1^R} \sup_{n \in \mathbb{N}} (R^n)_{i,j} \\ &= \sup_{n \in \mathbb{N}} \sum_{j \in S_1^R} (R^n)_{i,j} \quad \text{by the monotone convergence theorem} \\ &\leq \sup_{n \in \mathbb{N}} \sum_{j \in S} (R^n)_{i,j} = 1 \end{aligned}$$

1.3 The Markov Chain of pPCF

Given a context Γ and a type A, we consider Λ_Γ^A as a set of states, and we define the reduction relation as a stochastic matrix Red given by

$$\mathsf{Red}(\Gamma, A)_{M,M'} = \begin{cases} p & \text{if } M \xrightarrow{p} M' \\ 1 & \text{if } M \text{ is a value and } M' = M \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

We also use the notation $\mathsf{Red}(A)$ for the matrix $\mathsf{Red}(\Gamma, A)$ when the typing context is empty. Also, we will simply write Red if the typing annotation is irrelevant or clear from the context. The number $\mathsf{Red}(\Gamma, A)_{M,M'}$ is the probability of M to reduce to M' in one step. Notice that all weak-normal terms are stationary states of $\mathsf{Red}(\Gamma, A)$, but not all stationary states are weak-normal terms. Therefore, if V is a weak-normal form, then the n-fold product $\mathsf{Red}(\Gamma, A)^n_{M,V}$ gives the probability that M reduces to V in at most n steps. This is precised by the following proposition (recall notation (1)).

Proposition 3 *Let M be term and V be a value in Λ_Γ^A. One has*

$$\mathsf{Red}(\Gamma, A)^n_{M,V} = \sum_{\varphi \in \mathsf{Path}^{\leq n}(M,V)} \mathsf{p}(\varphi)\,.$$

Hence, $\mathsf{Red}(\Gamma, A)^\infty_{M,V} = \sum_{\varphi \in \mathsf{Path}(M,V)} \mathsf{p}(\varphi)$.

Exercise 7. Prove Proposition 3.

Exercise 8. Does Red have stationary states that are not weak-head normal terms? and what about Red^2?

Exercise 9. A stochastic program can have different notions of termination. Given a program M, we say that :

- *M strongly terminates* (ST), whenever the set $\bigcup_n \mathsf{Path}(M, \underline{n})$ is finite;
- *M positively almost surely terminates* (PAST), whenever the expected runtime
$$\sum_{n\in\mathbb{N}} \sum_{\varphi\in\mathsf{Path}(M,\underline{n})} \mathsf{p}(\varphi)\mathrm{length}(\varphi)$$
is finite;
- *M almost surely terminates* (AST), whenever $\sum_n \mathsf{Red}^{\infty}_{M,\underline{n}} = 1$.

Prove that ST → PAST → AST and that no implication can be inverted. (*This exercise is not trivial. You can have a look at [?] to have some inspiration. . .*).

1.4 Basic Examples

We illustrate the expressive power of pPCF by encoding in this language simple probabilistic algorithms. We explain intuitively the behaviour of these programs, but a formal proof of their soundness would require more sophisticated tools, like a denotational semantics. In fact, the next section will provide one of such semantics, based on probabilistic coherence spaces.

"Let" construction. This version of pPCF, which is globally call-by-name, offers however the possibility of handling integers in a call-by-value way. For instance, we can define the typical call-by-value "let" construction as follows

$$\mathsf{let}\ x\ \mathsf{be}\ M\ \mathsf{in}\ N = \mathsf{if}(M, N\left[\underline{0}/x\right], z\cdot N\left[\mathsf{succ}(z)/x\right]) \tag{7}$$

and this construction is restricted to the type of natural numbers; it can be typed as:

$$\frac{\Gamma\vdash M:\iota \qquad \Gamma, x:\iota\vdash N:A}{\Gamma\vdash \mathsf{let}\ x\ \mathsf{be}\ M\ \mathsf{in}\ N : A}$$

The effect of this construction is that, before replacing x with M in N, M must be evaluated to a value $\underline{n}$. This is particularly important in the case where M is a probabilistic integer since this construction allows to "roll the dice" only once and then provide N with as many copies of the result as needed.

In accordance with this intuition, one can also check that the following reduction inference is derivable from the rules of Figure 1d

$$\frac{M \xrightarrow{p} M'}{\mathsf{let}\ x\ \mathsf{be}\ M\ \mathsf{in}\ N \xrightarrow{p} \mathsf{let}\ x\ \mathsf{be}\ M'\ \mathsf{in}\ N} \tag{8}$$

whereas *it is not true* that

$$\frac{M \xrightarrow{p} M'}{N\left[M/x\right] \xrightarrow{p} N\left[M'/x\right]} \tag{9}$$

Exercise 10. Prove (8) and give a counterexample to (9).

We have of course

$$\frac{}{\mathsf{let}\ x\ \mathsf{be}\ \underline{n}\ \mathsf{in}\ N \xrightarrow{1} N\,[\theta(n)/x]}$$

where $\theta(0) = \underline{0}$ and $\theta(n+1) = \mathsf{succ}(\underline{n})$ (which reduces to $\underline{n+1}$ in one deterministic step) by definition of this construction.

Random Generators. Using the functions defined in Exercice 6, we can define a closed term unif_2 of type $\iota \Rightarrow \iota$ which, given an integer n, yields a uniform probability distribution on the integers $0, \dots, 2^n - 1$:

$$\mathsf{unif}_2 = \mathsf{fix}(\lambda f^{\iota \Rightarrow \iota}\, \lambda x^{\iota}\, \mathsf{if}(x, \underline{0}, z \cdot \mathsf{if}(\mathsf{coin}, (f)\, z, z' \cdot (\mathsf{add})\, (\mathsf{exp}_2)\, z\, (f)\, z))) \tag{10}$$

Observe that, when evaluating $(\mathsf{unif}_2)\, M$ (where $\vdash M : \iota$), the term M is evaluated only once thanks to the CBV feature of the conditional construct. Indeed, we do not want the upper bound of the interval on which we produce a probability distribution to change during the computation (the result would be unpredictable!).

Exercise 11. Using the unif_2 and let constructions, define a term unif which, given an integer n, yields a *uniform probability distribution* on the integers $0, \dots, n$.

Exercise 12. Define a closed term binom of type $\iota \Rightarrow \iota$ which, given an integer n, yields a (fair) *binomial distribution* out of n trials, i.e. $(\mathsf{binom})\, \underline{n}$ evaluates to $\underline{k}$ with the probability of getting k-times $\underline{1}$ in a sequence of n independent evaluations of coin.

Las Vegas algorithms. A Las Vegas algorithm is a randomized algorithm that always gives the correct result but its running time depends on the draws from the random variables in the algorithm.

Exercise 13. One of the simplest example of a Las Vegas algorithm can be used to find zeros in a finite array: given a function $f : \mathbb{N} \to \mathbb{N}$ and $n \in \mathbb{N}$, find a $k \in \{0, \dots, n\}$ such that $f(k) = 0$. This can be done by iterating random choices of k until we get a value such that $f(k) = 0$. Define a closed term M of type $(\iota \Rightarrow \iota) \Rightarrow \iota \Rightarrow \iota$ that implements this algorithm.

One can notice that our CBV version of the conditional is fundamental in solving Exercise 13. In fact, we strongly believe that this algorithm cannot be written with the usual version of the conditional (as in standard PCF) but we didn't really try to prove this. Do you have some hints in proving (or disproving) this conjecture?

Random-walks. We can define a random-walk over $\mathbb{N}$ as a closed term W of type $\iota \Rightarrow \iota$, meaning that a particle at position $i \in \mathbb{N}$ will evolve in one step to position $j \in \mathbb{N}$ with the probability of $(W)\, \underline{i}$ to evaluate to $\underline{j}$.

Exercise 14. Define a closed term W of type $\iota \Rightarrow \iota$ representing the random-walk of Equation (3).

The following exercise give you an exemple of how natural is the use of higher-order combinators for probabilistic programming. One can in fact defines an iterator of random processes independently from the specific process to iterate.

Exercise 15. Define a closed term iter of type $(\iota \Rightarrow \iota) \Rightarrow \iota \Rightarrow \iota \ \Rightarrow \ \iota$ that takes a term W representing a random-walk, a numeral $\underline{n}$ and returns a term of type $\iota \Rightarrow \iota$ simulating n-iterations of W.

In the above exercices, we just argue intuitively that the solutions actually satisfy the required specification. In fact, proving the soundness formally can be quite burdensome: for example, try to prove that the term $(\mathsf{iter})\, W\, \underline{n}$, with W and iter defined in resp. Exercice 14 and 15, expresses in pPCF the matrix W^n, for W given in (3). The major difficulty is that the operational semantics of pPCF, i.e. the definition of the matrix Red^∞ is not defined compositionally but with respect to a Makov chain (section 1.3). The next section will present the probabilistic coherence spaces as a denotational model of pPCF. One major feature of a denotational semantics is to be defined compositionally on the structure of a term. The adequacy theorem (Section **??**) will then prove the equivalence between the denotational model and the definition of Red^∞ on ground types, so allowing for compositional proofs of soundness.