Fooling the Parallel Or Tester with Probability 8/27

Jean Goubault-Larrecq*
LSV, ENS Paris-Saclay, CNRS, Université Paris-Saclay, France
Email: goubault@lsv.fr
November 1, 2019

Abstract

It is well-known that the higher-order language PCF is not fully abstract: there is a program—the so-called parallel or tester, meant to test whether its input behaves as a parallel or—which never terminates on any input, operationally, but is denotationally non-trivial. We explore a probabilistic variant of PCF, and ask whether the parallel or tester exhibits a similar behavior there. The answer is no: operationally, one can feed the parallel or tester an input that will fool it into thinking it is a parallel or. We show that the largest probability of success of such would-be parallel ors is exactly $8/27$. The bound is reached by a very simple probabilistic program. The difficult part is to show that that bound cannot be exceeded.

1 Introduction

There is a recurring theme in security: to defeat a strong adversary, you need to rely on random choice. This paper will be a somewhat devious illustration of that principle, in the field of programming language semantics.

The higher-order, functional language PCF [Plo77] forms the core of actual programming languages such as Haskell [Bir98], Plotkin [Plo77], and independently Sazonov [Saz76], had shown that PCF, while being adequate (i.e., its operational and denotational semantics match, in a precise sense), is not fully abstract: there are programs that are contextually equivalent (a notion arising from the operational semantics), but have different denotational semantics. (One should note that, conversely, two programs with the same denotational semantics are always contextually equivalent.)

*This research was partially supported by Labex DigiCosme (project ANR-11-LABEX-0045-DIGICOSME) operated by ANR as part of the program “Investissement d’Avenir” Idex Paris-Saclay (ANR-11-IDEX-0003-02).
The argument is as follows. In the denotational model, there is a function of type \( \text{int} \to \text{int} \to \text{int} \) called \textit{parallel or}, which maps the pair \( 1, 1 \) to \( 1 \), and both \( 0, N \) and \( N, 0 \) to \( 0 \), for whatever program \( N \) (including non-terminating programs). One can show that parallel or is undefinable in PCF. More is true. One can define a PCF program, the \textit{parallel or tester}, which takes an argument \( f: \text{int} \to \text{int} \to \text{int} \), and tests whether \( f \) is a parallel or, by testing whether \( f11 = 1 \), \( f0\Omega = 0 \), and \( f\Omega0 = 0 \), where \( \Omega \) is a canonical non-terminating program. The parallel or tester is contextually equivalent to the always non-terminating program \( \lambda f.\Omega \), meaning that applying it to any PCF program (for \( f \)) will never terminate. However, the denotational semantics of the parallel or tester and of \( \lambda f.\Omega \) differ: applied to any given parallel or map (which exists in the denotational model), one returns and the other one does not.

We introduce a probabilistic variant of PCF which we call PCFP, and we define a suitable parallel or tester \textit{portest}. A PCFP program \( M \) \textit{fools} the parallel or tester if \textit{portest} applied to \( M \) terminates. In PCF, there is no way of fooling the parallel or tester. Our purpose is to show that one can fool the parallel or tester of PCFP with probability at most \( 8/27 \), and that this bound is attained. The optimal fooler is easy to define. The hard part is to show that one cannot do better.

A final word before we start. Even though we started by motivating it from matters related to full abstraction, which involves both operational and denotational semantics, the question we are addressing is purely \textit{operational} in nature: it is only concerned with the behavior of \textit{portest} under its operational semantics, under arbitrary PCFP contexts. Nonetheless, denotational semantics will be essential in our proof.

Outline. We define the syntax of \( \text{PCFP} \) in Section 2 its operational semantics in Section 3 and—once we have stated the required basic facts we need from domain theory in Section 4—its denotational semantics in Section 5. We state the adequacy theorem at the end of the latter section. This says that the operational and denotational probabilities that a term \( M \) of type \( \text{int} \) terminates on any given value \( n \in \mathbb{Z} \) are the same. We define the parallel tester, and show that it can be fooled with probability \( 8/27 \) at most, in Section 6. We conclude by citing some recent related work in Section 7.

2 The syntax of PCFP

PCFP is a typed language. The \textit{types} are given by the grammar:

\[
\sigma, \tau, \cdots ::= \text{int} \quad \text{basic types} \\
\mid D\tau \quad \text{type of (subprobability) distributions on } \tau \\
\mid \sigma \to \tau \quad \text{function types.}
\]

Mathematically, \( D\tau \) will be the type of subprobability valuations of elements of type \( \tau \). Operationally, an element of type \( D\tau \) is just a random value of type \( \tau \). There is only one basic type, \( \text{int} \), but one could envision a more expressive algebra of datatypes.
A computation type is a type of the form $D\tau$ or $\sigma \to \tau$ where $\tau$ is a computation type. The computation types are the types where one can do computation, in particular whose objects can be defined by recursion.

Our language will have functions, and a function mapping inputs of type $\sigma$ to outputs of type $\tau$ will have type $\sigma \to \tau$. We write $\sigma_1 \to \sigma_2 \to \cdots \to \sigma_n \to \tau$ for $\sigma_1 \to (\sigma_2 \to (\cdots \to (\sigma_n \to \tau) \cdots))$, and this is a type of functions taking $n$ inputs, of respective types $\sigma_1, \sigma_2, \ldots, \sigma_n$ and returning outputs of type $\tau$.

We fix a countably infinite set of variables $x_1, y_1, z_1, \ldots$, for each type $\tau$. Each variable has a unique type, which we read off from its subscript. We will occasionally omit the type subscript when it is clear from context, or irrelevant.

The terms $M, N, \ldots$ of our language are defined inductively, together with their types, in Figure 1. We agree to write $M : \tau$ to mean “$M$ is a term, of type $\tau$”. We shall write $MN_1N_2 \cdots N_n$ for $(\cdots((MN_1)N_2)\cdots)N_n$, and $\lambda x_1, \lambda x_2, \cdots, \lambda x_n.M$ for $\lambda x_1, \lambda x_2, \cdots, \lambda x_n.M$. We shall also use the abbreviations $\text{let } x_1 = M \text{ in } N$ for $(\lambda x_1.M)N$ and $\text{letrec } f_r = M \text{ in } N$, where $M : \tau$, for let $f_r = \text{rec } (\lambda f_r.M)$ in $N$. Finally, we shall write $\text{do } x_1 := M; N$ for $\text{bind}_{\tau, \sigma} M(x_1, M, N)$, of type $D\tau$ (draw $x_1$ at random along distribution $M$, then run $N$). $M \oplus N$ is meant to execute either $M$ or $N$ with probability $1/2$.

The free variables and the bound variables of a term $M$ are defined as usual. A term with no free variable is ground. For a substitution $\theta \overset{\text{def}}{=} [x_1 := N_1, \cdots, x_k := N_k]$ (where each $N_i$ has the same type as $x_i$, and the variables $x_i$ are pairwise distinct), we write $M\theta$ for the parallel substitution of each $N_i$ for each $x_i$, and $\text{dom } \theta$ for $\{x_1, \cdots, x_k\}$. We say that $\theta$ is ground if $N_1, \ldots, N_k$ are all ground.

**Example 2.1** The term $\text{rand } \text{int} \overset{\text{def}}{=} \text{rec } \text{int} \to D\text{int} (\lambda r. \lambda n. \text{int}. r(s(m) \oplus \text{ret } n) m))$ is of type $D\text{int}$. As we will see, this draws a natural number $n$ at random, with probability $1/2^{n+1}$.
Example 2.2 Rejection sampling is a process by which one draws an element of a subset \( A \) of a space \( X \), as follows: we draw an element of \( X \) at random, and we return it if it lies in \( A \), otherwise we start all over again. Here is a simple example of rejection sampling, meant to draw a number uniformly among \( \{0, 1, 2\} \). The idea is to draw two independent bits at random, representing a number in \( X \) defined by removing the square brackets in \( \{0, 1, 2, 3\} \), and to use rejection sampling on \( A \) defined by \( \{0, 1, 2\} \). Formally, we define the PCF term \( \text{rand}_3 \defeq \text{rec}_{\text{Dint}}(\lambda p_{\text{Dint}}.((\text{ret}_{\text{int}} \ 0 \oplus \text{ret}_{\text{int}} 1) \oplus (\text{ret}_{\text{int}} 2 \oplus p_{\text{Dint}}))) \). Note that this uses recursion to define a distribution, not a function.

3 Operational semantics

The elementary contexts \( E \), with their types \( \sigma \vdash \tau \), are defined as:

- \([\_ \ N] \) of type \( (\sigma \rightarrow \tau) \vdash \tau \), for every \( N: \sigma \), and for every type \( \tau \);
- \([s\_] \) and \([p\_] \), of type \( \text{int} \vdash \text{int} \);
- \([\text{if}_{} = 0 \text{ then } N \text{ else } P] \), of type \( \text{int} \vdash \tau \), for all \( N, P: \tau \);
- \([\text{bind}_{\sigma, \tau} \ _\ N] \), of type \( D\sigma \vdash D\tau \), for every \( N: \sigma \rightarrow D\tau \).

The initial contexts are \([\_] \) (of type \( \sigma \vdash \sigma \) for any \( \sigma \)) and \([\text{ret}_{\text{int}} \_] \) (of type \( \text{int} \vdash D\text{int} \)). The (evaluation) contexts \( C \) are the finite sequences \( E_0 E_1 \cdots E_n \), \( n \in \mathbb{N} \), where \( E_0 \) is an initial context of type \( \sigma_1 \vdash \sigma_0 \), each \( E_i \) \( (1 \leq i \leq n) \) is an elementary context of type \( \sigma_{i+1} \vdash \sigma_i \). Then we say that \( C \) has type \( \sigma_n \vdash \sigma_0 \).

The notation \( C[M] \) makes sense for every context \( C \defeq E_0 E_1 \cdots E_n \) of type \( \sigma \vdash \tau \) and every \( M: \sigma \), and is defined as \( E_0[E_1[\cdots[E_n[M]]]] \), where \( E[M] \) is defined by removing the square brackets in \( E \) and replacing the hole \( _\_ \) by \( M \).

E.g., if \( C = [\text{ret}_{\text{int}} \_][p\_] \), then \( C[M] = \text{ret}_{\text{int}}(pM) \).

A configuration (of type \( \tau \)) is a pair \( C \cdot M \), where \( C \) is a context of type \( \sigma \vdash \tau \) and \( M: \sigma \).
The operational semantics of PCF$_p$—an abstract interpreter that runs PCF$_p$ programs—is a probabilistic transition system on configurations, defined by the rules of Figure 2. We write $s \overset{\alpha}{\to} s'$ to say that one can go from configuration $s$ to configuration $s'$ in one step, with probability $\alpha$.

A trace is a sequence $s_0 \overset{\alpha_0}{\to} s_1 \overset{\alpha_1}{\to} \cdots \overset{\alpha_m}{\to} s_m$, where $m \in \mathbb{N}$, and where each $s_{i-1} \overset{\alpha_i}{\to} s_i$ is an instance of a rule of Figure 2. The trace starts at $s_0$, ends at $s_m$, its length is $m$ and its weight is the product $\alpha \overset{\text{def}}{=} \alpha_1 \cdots \alpha_2 \cdots \alpha_m$. In that case, we also write $s_0 \overset{\alpha}{\to^*} s_m$.

The run starting at $s_0$ is the tree of all traces starting at $s_0$. Its root is $s_0$ itself, and for each vertex $s$ in the tree, for each instance of a rule of the form $s \overset{\alpha}{\to} t$, $t$ is a successor of $s$, and the edge from $s$ to $t$ is labeled $\alpha$.

For every configuration $s$ of type $\text{Dint}$, and every $n \in \mathbb{Z}$, we define $\Pr[s \downarrow n]$ as the sum of the weights of all traces that start at $s$ and end at $[\text{ret}_{\text{int}} \_] \cdot n$. This is the subprobability that $s$ eventually computes $n$. We also write $\Pr[M \downarrow n]$ for $\Pr[[\_] \cdot M \downarrow n]$, where $M : \text{Dint}$.

Figure 3: An example run in PCF$_p$
Example 3.1 The run starting at \texttt{rand\_int} (see Example 2.1) is shown in Figure 3. We have abbreviated some sequences of \( \rightarrow \) steps as \( \rightarrow^* \). One sees that \( \Pr[\texttt{rand\_int} \downarrow n] = 1/2^{n+1} \) for every \( n \in \mathbb{N} \), and is zero for every \( n < 0 \). Notice the infinite branch on the left, whose weight is 0.

Example 3.2 We let the reader draw the run starting at \texttt{rand3} (see Example 2.2), and check that \( \Pr[\texttt{rand3} \downarrow n] \) is equal to 1/3 if \( n \in \{0, 1, 2\} \), 0 otherwise. Explicitly, if \( n \in \{0, 1, 2\} \), show that the traces that start at \texttt{rand3} and end at \([\texttt{ret\_int \_}] \cdot n \) have respective weights 1/4, 1/4 ⋅ 1/4, \ldots, \((1/4)^n \cdot 1/4\), \ldots, and that the sum of those weights is 1/3.

The following is immediate.

Lemma 3.3 The following hold:

1. For every rule \( s \xrightarrow{\alpha} t \), \( t \) and \( s \) have the same type.

2. For every rule of the form \( s \xrightarrow{1} t \) of type \texttt{DInt}, for every \( n \in \mathbb{Z} \), \( \Pr[t \downarrow n] = \Pr[s \downarrow n] \).

3. \( \Pr[C \cdot M \oplus N \downarrow n] = \frac{1}{2} \Pr[C \cdot M \downarrow n] + \frac{1}{2} \Pr[C \cdot N \downarrow n] \). \qed

4 A refresher on domain theory

We will require some elementary domain theory, for which we refer the reader to \cite{GHK03, AJ94, Gon13}. A poset \( X \) is a set with a partial ordering, which we will always write as \( \leq \). A directed family \( D \subseteq X \) is a non-empty family such that every pair of points of \( D \) has an upper bound in \( D \). A \texttt{dcpo} is a poset in which every directed family \( D \) has a supremum \( \sup D \). If \( D = (x_i)_{i \in I} \), we also write \( \sup_{i \in I} x_i \) for \( \sup D \).

The product \( X \times Y \) of two dcpo’s is the set of pairs \((x, y), x \in X, y \in Y \), ordered by \((x, y) \leq (x', y') \) if and only if \( x \leq x' \) and \( y \leq y' \).

For any two dcpo’s \( X \) and \( Y \), a map \( f: X \to Y \) is \textit{Scott-continuous} if and only if it is monotonic \( (x \leq x') \implies f(x) \leq f(x') \) and preserves directed suprema (for every directed family \((x_i)_{i \in I} \) in \( X \), \( \sup_{i \in I} f(x_i) = f(\sup_{i \in I} x_i) \)). There is a category \texttt{Dcpo} of dcpo’s and Scott-continuous maps.

We order maps from \( X \) to \( Y \) by \( f \leq g \) if and only if \( f(x) \leq g(x) \) for every \( x \in X \). The poset \([X \to Y]\) of all Scott-continuous maps from \( X \) to \( Y \) is then again a dcpo, and directed suprema are computed pointwise: \((\sup_{i \in I} f_i)(x) = \sup_{i \in I} (f_i(x)) \). \texttt{Dcpo} is a Cartesian-closed category—a model of simply-typed \( \lambda \)-calculus—and that can be said more concretely as follows:

- for all dcpo’s \( X, Y \), there is a Scott-continuous map \texttt{App}: \([X \to Y] \times X \to Y \) defined by \texttt{App}(\( f \), \( x \)) \texttt{def} \( f(x) \);

- for all dcpo’s \( X, Y, Z \), for every Scott-continuous map \( f: Z \times X \to Y \), the map \( \Lambda_\ast X(f): Z \to \[X \to Y\] \) defined by \( \Lambda_\ast X(f)(z) \texttt{def} f(z, x) \) is Scott-continuous;
• those satisfy certain equations which we will not require.

If the dcpo \( X \) is pointed, namely if it has a least element \( \bot \), then every Scott-continuous map \( f : X \to X \) has a least fixed point \( \lfp_X(f) \defeq \sup_{n \in \mathbb{N}} f^n(\bot) \).
This is used to interpret recursion. Additionally, the map \( \lfp_X : [X \to X] \to X \) is itself Scott-continuous.

The set \( \mathbb{R}_+ \defeq \mathbb{R}_+ \cup \{\infty\} \) of extended non-negative real numbers is a dcpo under the usual ordering. We write \( \mathcal{L}X \) for \( [X \to \mathbb{R}_+] \). Its elements are called the lower semicontinuous functions in analysis.

A \emph{Scott-open} subset \( U \) of a dcpo \( X \) is an upwards-closed subset \( (x \in U \text{ and } x \leq y \text{ imply } y \in U) \) that is inaccessible from below (every directed family \( D \) such that \( \sup D \in U \) intersects \( U \)). The lattice of Scott-open subsets is written \( \mathcal{O}X \), and forms a topology, the \emph{Scott topology} on \( X \). Note that \( \mathcal{O}X \) is itself a dcpo under inclusion, and directed suprema are computed as unions.

The \emph{Scott-closed} sets are the complements of Scott-open sets, i.e., the downwards-closed subsets \( C \) such that for every directed family \( D \subseteq C \), \( \sup D \in C \).

In order to give a denotational semantics to probabilistic choice, we will follow Jones [JP89, Jon90]. A \emph{continuous valuation} on \( X \) is a map \( \nu : \mathcal{O}X \to \mathbb{R}_+ \) that is \emph{strict} \( (\nu(\emptyset) = 0) \), \emph{monotone} \( (U \subseteq V \text{ implies } \nu(U) \leq \nu(V)) \), \emph{modular} \( (\nu(U) + \nu(V) = \nu(U \cup V) + \nu(U \cap V)) \), and \emph{Scott-continuous} \( (\nu(\bigcup_{i \in I} U_i) = \sup_{i \in I} \nu(U_i)) \). A \emph{subprobability valuation} additionally satisfies \( \nu(X) \leq 1 \). Continuous valuations and measures are very close concepts: see [KL05] for details.

Among subprobability valuations, one finds the \emph{Dirac valuation} \( \delta_x \), for each \( x \in X \), defined by \( \delta_x(U) \defeq 1 \text{ if } x \in U, 0 \text{ otherwise.} \) One can integrate any Scott-continuous map \( f : X \to \mathbb{R}_+ \), and the integral \( \int_{x \in X} f(x)d\nu \) is Scott-continuous and linear (i.e., commutes with sums and scalar products by elements of \( \mathbb{R}_+ \)) both in \( f \) and in \( \nu \).

We write \( V_{\leq 1}X \) for the poset of subprobability valuations on \( X \). This is a dcpo under the pointwise ordering \( (\mu \leq \nu \text{ if and only if } \mu(U) \leq \nu(U) \text{ for every } U \in \mathcal{O}X) \), and directed suprema are computed pointwise \( (\sup_{i \in I} \nu_i(U) = \sup_{i \in I} (\nu_i(U))) \). Additionally, \( V_{\leq 1} \) defines a \emph{monad} on \( \textbf{Dcpo} \). Concretely:

• there is a \emph{unit} \( \eta : X \to V_{\leq 1}X \), which is the continuous map \( x \mapsto \delta_x \);
• every Scott-continuous map \( f : X \to V_{\leq 1}Y \) has an \emph{extension} \( f^! : V_{\leq 1}X \to V_{\leq 1}Y \), defined by \( f^!(\nu)(V) \defeq \int_{x \in X} f(x)(V)d\nu \);
• those satisfy a certain number of equations, of which we will need the following:

\[
f^!(\eta(x)) = f(x) \quad (1)
\]

\[
\int_{y \in Y} h(y)df^!(\nu) = \int_{x \in X} \left( \int_{y \in Y} h(y)df(x) \right) d\nu, \quad (2)
\]

for all Scott-continuous maps \( f : X \to Y, h : Y \to \mathbb{R}_+ \), and every \( \nu \in V_{\leq 1}X \).

Note that the map \( f \mapsto f^! \) is itself Scott-continuous.
Taking suprema over \( n \) establishes soundness, namely
\[
\text{The proof is relatively standard, and given in the appendices. Appendix A}
\]
\[\text{Example 5.2}
\]
Theorem 5.1 (Adequacy)
\[\text{The operational semantics and the denotational semantics match, namely:}
\]
\[\text{The types } \tau \text{ are interpreted as dcpos } \llbracket \tau \rrbracket, \text{ as follows: } \llbracket \text{int} \rrbracket = \mathbb{Z}, \text{ with equality as ordering; } \llbracket D\tau \rrbracket = \mathbb{V}_{\leq 1}[\llbracket \tau \rrbracket]; \text{ and } \llbracket \sigma \rightarrow \tau \rrbracket = \llbracket \sigma \rrbracket \rightarrow \llbracket \tau \rrbracket]. \text{ Note that } \llbracket \tau \rrbracket \text{ is pointed for every computation type } \tau, \text{ so if}_{[\llbracket \tau \rrbracket]} \text{ makes sense in those cases.}
\]
An environment is a map \( \rho \) sending each variable \( x_r \) to an element \( \rho(x_r) \) of \( \llbracket \tau \rrbracket \). The dcpo \( \text{Enum} \) of environments is the product \( \prod_x \text{variable}_{\llbracket \tau \rrbracket} \), with the usual componentwise ordering. When \( V \in [\sigma] \), we write \( \rho[x_\sigma := V] \) for the environment that maps \( x_\sigma \) to \( V \), and all other variables \( y \) to \( \rho(y) \).

Let us write \( V \in X \mapsto f(V) \) for the function that maps every \( V \in X \) to the value \( f(V) \). We can now define the value \( \llbracket M \rrbracket \) of terms \( M : \tau \), as Scott-continuous maps \( \rho \in \text{Env} \mapsto \llbracket M \rrbracket \rho \), by induction on \( M \), see Figure 4.

The operational semantics and the denotational semantics match, namely:

**Theorem 5.1 (Adequacy)** For every ground term \( M : \text{Dint} \), for every \( n \in \mathbb{Z} \), \( \llbracket M \rrbracket ([\{n\}]) = \text{Pr}([M \downarrow n]) \).

The proof is relatively standard, and given in the appendices. Appendix A establishes soundness, namely \( \llbracket M \rrbracket ([\{n\}]) \geq \text{Pr}([M \downarrow n]) \), and Appendix B shows the converse inequality, using appropriate logical relations.

**Example 5.2** We retrieve the result of Example 3.1 using adequacy as follows.
\[
\llbracket \lambda r \rightarrow \text{Dint}. \lambda m_{\text{int}}. r(s m) \oplus \text{ret}_{\text{int}} m \rrbracket \text{ is the function } F \text{ that maps every } r \in \llbracket \text{int} \rightarrow \text{Dint} \rrbracket \text{ (the value of } r \text{) and every } m \in \llbracket \text{int} \rrbracket = \mathbb{Z} \text{ to } 1/2\varphi(m+1)+1/2\delta_m. \text{ Let } \varphi_k \text{ def } F^k(\bot), \text{ for every } k \in \mathbb{N}. \text{ Then } \varphi_0 = \bot \text{ maps every } m \in \mathbb{N} \text{ to the zero valuation, } \varphi_1(m) = 1/2\delta_m \text{ for every } m \in \mathbb{N}, \varphi_2(m) = 1/4\delta_{m+1} + 1/2\delta_m \text{ for every } m \in \mathbb{N}, \text{ etc. By induction on } k, \varphi_k(m) = \sum_{i=0}^{k-1} 1/2^{i+1}\delta_{m+i}. \text{ Taking suprema over } k, \text{ we obtain that if}_{\llbracket \text{int} \rightarrow \text{Dint} \rrbracket} (F) \text{ maps every } m \in \mathbb{N} \text{ to } \sum_{i=0}^{\infty} 1/2^{i+1}\delta_{m+i}. \text{ Then } \llbracket \text{randnat} \rrbracket = \text{if}_{\llbracket \text{int} \rightarrow \text{Dint} \rrbracket} (F)(0) = \sum_{n\in\mathbb{N}} 1/2^{n+1}\delta_n.
\]

**Example 5.3** We retrieve the result of Example 2.2 using adequacy, as follows.

The semantics of \( \lambda p_{\text{Dint}}. ((\text{ret}_{\text{int}} 0 \oplus \text{ret}_{\text{int}} 1) \oplus (\text{ret}_{\text{int}} 2 \oplus p_{\text{Dint}})) \) is the function \( f \) that maps every \( \nu \in \llbracket \text{Dint} \rrbracket \) to \( 1/2\delta_0 + 1/3\delta_1 + 1/4\delta_2 + 1/5\nu \) and, for every \( n \in \mathbb{N} \), \( f^n(0) = a_n\delta_0 + a_n\delta_1 + a_n\delta_2 \) where \( a_n = 1/4 + (1/4)^2 + \cdots + (1/4)^n = 1/4(1-
(1/4)^n)/(1-1/4). Since \([\text{int}]\) has equality as ordering, the ordering on \([\text{Dint}]\) is given by comparing the coefficients of each \(\delta_N\), \(N \in [\text{int}]\). In particular, the least fixed point of \(f\) is obtained as \(a\delta_0 + a\delta_1 + a\delta_2\), where \(a = \sup\{a_n : n \in \mathbb{N}\}\).

**Example 5.4** Here is a lengthier example, which we will leave to the reader. While lengthy, working denotationally is doable. Proving the same argument operational would be next to impossible, even in the special case \(\tau = \text{int}\).

We define a more general form of rejection sampling, as follows. Let \(\tau\) be any type. We consider the PCF term:

\[
\text{sample} \overset{\text{def}}{=} \lambda p_D. \lambda \text{sel}_\tau \rightarrow \text{Dint}. \\
\text{rec}_D, (\lambda r_D. \text{do } x_\tau \leftarrow p_D; \\
\text{do } b_{\text{int}} \leftarrow \text{sel}_\tau \rightarrow \text{Dint} x_\tau; \\
\text{if } b_{\text{bool}} = 0 \text{ then ret, } x_\tau \text{ else } r_D).
\]

The idea is that we draw \(x\) according to distribution \(p\), then we call \(\text{sel}\) as a predicate on \(x\). If the result, \(b\), is true (zero) then we return \(x\), otherwise we start all over. Note that \(\text{sel}\) can itself return a random \(b\).

For every \(g \in \mathcal{L}[[\tau]]\), and every \(\nu [[\text{Dint}]]\), we let \(g \cdot \nu\) (sometimes written \(g \, d\nu\)) be the continuous valuation defined from \(\nu\) by using \(g\) as a density, namely \((g \cdot \nu)(U) \overset{\text{def}}{=} \int_{x \in [\tau]} \chi_U(x) g(x) d\nu\) for every open subset \(U\) of \([\tau]\), where \(\chi_U\) is the characteristic map of \(U\). One can check that \(g \cdot \nu = (x \mapsto g(x)\delta_U)\nu\), using the equality \(\chi_U(x) = \delta_U(x)\), and, using \([\mathbb{2}]\), that for every \(h \in \mathcal{L}[[\tau]]\), \(\int_{x \in X} h(x) d(g \cdot \nu) = \int_{x \in X} h(x) g(x) d\nu\).

For every \(s \in [[\tau \rightarrow \text{Dint}]]\), for every \(x \in [[\tau]]\), let \(s_0(x) \overset{\text{def}}{=} s(x)\{0\}\), \(s_1(x) \overset{\text{def}}{=} s(x)\mathbb{Z} \setminus \{0\}\). We let the reader check that, for every environment \(\[\text{sample}\]\) \(\rho\) maps every subprobability valuation \(\nu\) on \([\tau]\) and every \(s \in [[\tau \rightarrow \text{Dint}]]\) to the subprobability valuation \(\frac{1}{1 - \nu([\tau])}(s_0 \cdot \nu)\) if \((s_1 \cdot \nu)([\tau]) \neq 0\), to the zero valuation otherwise.

In particular, if \(s\) is a predicate, implemented as a function that maps every \(x \in U \subseteq [[\tau]]\) to \(\delta_U\) and every \(x \in V \subseteq [[\tau]]\) (for some disjoint open sets \(U\) and \(V\)) to \(\delta_V\), so that \(s_0 = \chi_U\) and \(s_1 = \chi_V\), then \([\text{sample}]\) \(\rho(\nu)(s)\) is the subprobability valuation \(\frac{1}{1 - \nu([\tau])}\nu_U\) if \(\nu(V) \neq 0\), the zero valuation otherwise. (\(\nu_U\) denotes the restriction of \(\nu\) to \(U\), defined by \(\nu_U(V) \overset{\text{def}}{=} \nu(U \cap V)\).)

In the special case where \(V\) is the complement of \(U\), it follows that \(\text{sample}\) implements conditional probabilities: \([\text{sample}]\) \(\rho(\nu)(s)(W)\) is the probability that a \(\nu\)-random element lies in \(W\), conditioned on the fact that it is in \(U\).

## 6 The parallel or tester

In PCF, computation happens at type \(\text{Dint}\), not \(\text{int}\), hence let us call \(\text{parallel}\) or function any \(f \in [[\text{Dint} \rightarrow \text{Dint} \rightarrow \text{Dint}]]\) such that \(f(\delta_0)(\delta_1) = \delta_1\) and \(f(\delta_0)(\nu) = f(\nu)(\delta_0) = \delta_0\) for every \(\nu \in [[\text{Dint}]]\). Realizing that every element of
\[ \langle \text{Dint} \rangle \text{ is of the form } a\delta_0 + b\delta_1, \text{ with } a, b \in \mathbb{R}_+ \text{ such that } a + b \leq 1, \text{ the function } \text{por} \text{ defined by } \text{por}(a\delta_0 + b\delta_1)(a'\delta_0 + b'\delta_1) \overset{\text{}\text{}\text{def}}{=} (a + a' - ab\delta_0 + bb'\delta_1) \delta_0 + bb'\delta_1 \text{ is such a parallel or function.} \]

Note how parallel ors differ from the usual left-to-right sequential or used in most programming languages:

\[ \text{lror} \overset{\text{}\text{}\text{def}}{=} \lambda p \text{Dint}. \lambda q \text{Dint}. \text{do } x_{\text{int}} \leftarrow p \text{Dint}; \text{if } x_{\text{int}} = 0 \text{ then } \text{ret}_{\text{int}} 0 \text{ else } q \text{Dint} \]

whose semantics is given by \[ \llbracket \text{lror} \rrbracket (a\delta_0 + b\delta_1)(a'\delta_0 + b'\delta_1) = (a + ba')\delta_0 + bb'\delta_1 \]—so \[ \llbracket \text{lror} \rrbracket \] maps \( \delta_1, \delta_1 \) to \( \delta_1 \), and \( \delta_0, \nu \) to \( \delta_0 \), but maps \( a\delta_0 + b\delta_1, \delta_0 \) to \( (a + b)\delta_0, \) not \( \delta_0 \).

Symmetrically, there is a right-to-left sequential or:

\[ \text{rlor} \overset{\text{}\text{}\text{def}}{=} \lambda p \text{Dint}. \lambda q \text{Dint}. \text{do } x_{\text{int}} \leftarrow q \text{Dint}; \text{if } x_{\text{int}} \text{ then } \text{ret}_{\text{int}} 0 \text{ else } p \text{Dint}. \]

We define a parallel or tester as follows:

\[ \text{portest} \overset{\text{}\text{}\text{def}}{=} \lambda f \text{Dint} \rightarrow \text{Dint} \rightarrow \text{Dint}. \text{do } x_{\text{int}} \leftarrow f(\text{ret}_{\text{int}} 1)(\text{ret}_{\text{int}} 1); \]
\[ \text{if } x_{\text{int}} = 0 \text{ then } \Omega \]
\[ \text{else } (\text{do } y_{\text{int}} \leftarrow f(\text{ret}_{\text{int}} 0)(\Omega); \]
\[ \text{if } y_{\text{int}} = 0 \text{ then } (\text{do } z_{\text{int}} \leftarrow f(\Omega)(\text{ret}_{\text{int}} 0); \]
\[ \text{if } z_{\text{int}} = 0 \text{ then } \text{ret}_{\text{unit}} 0 \text{ else } \Omega) \]
\[ \text{else } \Omega), \]

where \( \Omega \overset{\text{}\text{}\text{def}}{=} \text{rec } (\lambda a \text{Dint}. a \text{Dint}). \) One can check that \[ \llbracket \text{portest} \rrbracket (\text{por}) = \delta_0 \], and that would hold for any other parallel or function instead of \( \text{por} \). If things worked in PCF\textsubscript{P} as in PCF, we would be able to show that \text{portest} is contextually equivalent to the constant map that loops on every input \( f \text{Dint} \rightarrow \text{Dint} \rightarrow \text{Dint}. \)

However, that is not the case. As we will now see, there is a PCF\textsubscript{P} term, the poor man’s parallel or \( \text{pmpor} \), such that \text{portest pmpor} terminates with non-zero probability. That term takes its two arguments of type \( \text{Dint} \), then decides to do one of the following three actions with equal probability 1/3: (1) call \text{lror} on the two arguments; (2) call \text{rlor} on the two arguments; or (3) return true (0), regardless of its arguments.

In order to define \( \text{pmpor} \), we need to draw an element out of three with equal probability. We do that by rejection sampling, imitating rand\textsubscript{3} (Examples 2.2, 3.2 and 5.3): we draw one element among four with equal probability, and we repeat until it falls in a specified subset of three. Hence we define:

\[ \text{pmpor} \overset{\text{}\text{}\text{def}}{=} \lambda p \text{Dint}. \lambda q \text{Dint}. \text{recDint}(\lambda r. \]
\[ (\text{lror } p q) \oplus (\text{rlor } p q) \oplus (\text{ret}_{\text{int}} 0 \oplus r)) \]
One can show that $\lror$ maps every pair of subprobability distributions $\mu, \nu$ on $[\int \mathbb{N}]$ to $\frac{1}{2} \lror (\mu(\nu)) + \frac{1}{2} \lror (\mu(\nu)) + \frac{1}{2} \delta_0$. Intuitively, $\lror$ will terminate with probability $(2/3)^3 = 8/27 \approx 0.296296 \ldots$; with $f = \lror$, the first test $f(\delta_1)(\delta_1) = \delta_1$ will succeed whether $f$ acts as $\lror$ or as $\ror$ (but not as the constant map returning $\delta_0$), which happens with probability $2/3$; the second test $f(\delta_0)(0) = \delta_0$ will succeed whether $f$ acts as $\lror$ or as the constant map returning $\delta_0$ (but not as $\lor$), again with probability $2/3$; and the final test $f(0)(\delta_0) = \delta_0$ will symmetrically succeed with probability $2/3$.

We now show that the probability $8/27$ is optimal. To this end, we need to use a logical relation $(\lror)_\tau$, namely a family of relations $\lror$, one for each type $\tau$, and related by certain constraints to be described below. Each $\lror$ will be an $I$-ary relation on values in $[\int \mathbb{N}]$, for some non-empty set $I$, namely $\lror \subseteq ([\int \mathbb{N}])^I$. In practice, we will take $I \mydef \{1, 2, 3\}$, but the proofs are easier if we keep $I$ arbitrary for now.

Our construction will be parameterized by an $I$-ary relation $\lror \subseteq [\int \mathbb{N}]$. We will also define an auxiliary family of relations $\lror$, as certain subsets of $([\int \mathbb{N}])^I$. We require $\lror$ to contains the all zero tuple $\mathbf{0} \mydef (0)_{i \in I}$, to be closed under directed suprema, and to be convex. (By convex, we mean that for all $\vec{x}, \vec{y} \in \lror$ and $a \in [0, 1]$, $a\vec{x} + (1 - a)\vec{y}$ is in $\lror$ as well.)

We define:

- $(n_i)_{i \in I} \in \lror_{\text{int}}$ if and only if all $n_i$ are equal;
- $(f_i)_{i \in I} \in \lror_{\text{dir}}$ if and only if for all $(V_i)_{i \in I} \in \lror_{\sigma}$, $(f_i(V_i))_{i \in I} \in \lror_{\tau}$;
- $(\nu_i)_{i \in I} \in \lror_{\text{pt}}$ if and only if for all $(h_i)_{i \in I} \in \lror_{\tau}$, $(\int_{V \in [\int \mathbb{N}]} h_i(V)d\nu_i)_{i \in I} \in \lror$;
- $(h_i)_{i \in I} \in \lror_{\text{dir}}$ if and only if for all $(V_i)_{i \in I} \in \lror_{\tau}$, $(h_i(V))_{i \in I} \in \lror$.

We also define $\lror \subseteq \Env^{I}$ by $(\rho_i)_{i \in I} \in \lror$ if and only if for every variable $x_\tau$, $(\rho_\tau(x_\tau))_{i \in I} \in \lror_{\tau}$. We prove the following basic lemma of logical relations:

**Proposition 6.1** For all $(\rho_i)_{i \in I} \in \lror$, for every $M : \tau$, $([\int \mathbb{N}]\rho_i)_{i \in I}$ is in $\lror$.

**Proof.** Step 1. We claim that for every type $\tau$, $\lror$ is closed under directed suprema taken in $([\int \mathbb{N}])^I$, and contains the least element $(-\tau)_{i \in I}$ if $\tau$ is a computation type. This is by induction on $\tau$. The claim is trivial for $\int$, since $[\int \mathbb{N}]^I$ is ordered by equality. For every directed family $(\vec{f}_j)_{j \in J}$ in $\lror_{\tau} \rightarrow \tau$, with $\vec{f}_j \mydef (f_{ji})_{i \in I}$, we form its supremum $\vec{f} \mydef (f_i)_{i \in I}$ pointwise, namely $f_i \mydef \sup_{j \in J} f_{ji}$. For every $(V_i)_{i \in I} \in \lror_{\sigma}$, $(f_{ji}(V_i))_{i \in I}$ is in $\lror_{\tau}$ for every $j \in J$, so by induction hypothesis $(f_i(V_i))_{i \in I}$ is also in $\lror_{\tau}$. It follows that $(f_i)_{i \in I}$ is in $\lror_{\tau} \rightarrow \tau$. For every directed family $(\vec{\nu}_j)_{j \in J}$ in $\lror_{\tau} \rightarrow \tau$, with $\vec{\nu}_j \mydef (\nu_{ji})_{i \in I}$, we form its supremum $\vec{\nu} \mydef (\nu_i)_{i \in I}$ pointwise, that is $\nu_i \mydef \sup_{j \in J} \nu_{ji}$. For all $(\vec{h})_{i \in I} \in \lror_{\tau}$, $(\int_{V \in [\int \mathbb{N}]} h_i(V)d\nu_i)_{i \in I} \in \lror_{\tau}$ for every $j \in J$, by induction hypothesis. We take suprema over $j \in J$. Since $\lror$ is closed under directed suprema, and integration
is Scott-continuous in the valuation, \((\int_{V \in \tau} h_i(V) \, d\nu_i)_{i \in I}\) is in \(\triangleright\). Since \((h_i)_{i \in I}\) is arbitrary, \((\nu_i)_{i \in I} \in \triangleright_{D\tau}\).

We also show that \((\bot_i)_{i \in I} \in \triangleright_{\tau}\) for every computation type \(\tau\). For function types, this is immediate. For types of the form \(D\tau\), we must check that \(\hat{0}\) is in \(\triangleright_{D\tau}\). For all \((h_i)_{i \in I} \in \triangleright_{\tau}^+\), we indeed have \((\int_{V \in \tau} h_i(V) \, d\nu_0)_{i \in I} \in \triangleright\), since \(\hat{0} \in \triangleright\).

Step 2. We claim that for all \((\nu_i)_{i \in I} \in \triangleright_{D\sigma}\), for all \((f_i)_{i \in I} \in \triangleright_{\sigma \rightarrow \tau_{D\tau}}\), \((f_i^i)_{i \in I} \in \triangleright_{D\tau}\). We wish to use the definition of \(\triangleright_{D\tau}\), so we consider an arbitrary tuple \((h_i)_{i \in I} \in \triangleright_{\tau}^+\), and we aim to prove that \((\int_{V \in \tau} h_i(V) \cdot f_i^i(\nu_i))_{i \in I}\) is in \(\triangleright\). For that, we use equation (2), to the effect that \(\int_{x \in \tau} h_i(V) \cdot f_i^i(\nu_i) = \int_{x \in \tau} (\int_{V \in \tau} h_i(V) \, d\nu_i(x)) \, d\nu_i\), for every \(i \in I\).

Let us define \(h_i^*(x) \overset{\text{df}}{=} \int_{V \in \tau} h_i(V) \, d\nu_i(x)\). We claim that \((h_i^*)_{i \in I} \in \triangleright_{\tau}^+\). Let \((x_i)_{i \in I} \in \triangleright_{\sigma}\). Then \((f_i(x_i))_{i \in I} \in \triangleright_{D\tau}\), and since \((h_i)_{i \in I} \in \triangleright_{\tau}^+\), \((h_i^*(x_i))_{i \in I} \in \triangleright\), by definition of \(\triangleright_{D\tau}\). Since \((x_i)_{i \in I} \in \triangleright_{\sigma}\) is arbitrary, \((h_i^*)_{i \in I} \in \triangleright_{\tau}^+\).

Since \((h_i^*)_{i \in I} \in \triangleright_{\tau}^+\) and \((\nu_i)_{i \in I} \in \triangleright_{D\sigma}\), by definition of \(\triangleright_{D\sigma}\) we obtain that \((\int_{x_i \in [\sigma]} h_i^*(x_i) \, d\nu_i)_{i \in I}\) is in \(\triangleright\), and this is exactly what we wanted to prove.

We now prove the claim by induction on \(M\). If \(M\) is a variable, this is by assumption. If \(M = 0\), this is trivial. If \(M\) is of the form \(\text{a}N\), then all the values \([N]_{\rho_i}\) are equal, hence also all the values \([M]_{\rho_i} = [N]_{\rho_i} + 1\). Similarly for terms of the form \(pN\). The case of applications is by definition of \(\triangleright_{\sigma \rightarrow \tau}\). In the case of abstractions \(\lambda x.\sigma.M\) with \(M\) : \(\tau\), we must show that, letting \(f_i\) be the map \(V \in [\sigma] \mapsto ([M]_{\rho_i}(x_{\sigma} \rightarrow V))\) \((i \in I)\), for all \((V_i)_{i \in I} \in \triangleright_{\sigma}\), \((f_i(V_i))_{i \in I} \in \triangleright_{\tau}\). This boils down to checking that \(([M]_{\rho_i}(x_{\sigma} \rightarrow V_i))_{i \in I} \in \triangleright\), for all \((V_i)_{i \in I} \in \triangleright_{\sigma}\), which follows immediately from the induction hypothesis and the easily checked fact that \((\rho_i[x_{\sigma} \rightarrow V_i])_{i \in I}\) is in \(\triangleright_{\sigma}\).

The case of terms of the form \(\text{rec}_{\tau} M\), where \(\tau\) is a computation type, is more interesting. Let \(f_i\) be the map \([M]_{\rho_i} : [\tau] \rightarrow [\tau]\). By induction hypothesis \((f_i)_{i \in I} \in \triangleright_{\tau}\), so for all \((a_i)_{i \in I} \in \triangleright_{\tau}\), \((f_i(a_i))_{i \in I} \in \triangleright_{\tau}\). Iterating this, we have \((f_n(\bot_i))_{i \in I} \in \triangleright_{\tau}\), for every \(n \in \mathbb{N}\). By Step 1, \((\bot_i)_{i \in I} \in \triangleright_{\tau}\). Hence \((f_n(\bot_i))_{i \in I} \in \triangleright_{\tau}\), for every \(n \in \mathbb{N}\). Since \(\triangleright\) is closed under directed suprema by Step 1, \((\text{fpn}[\tau] f_i)_{i \in I} = ([\text{rec}_{\tau} M]_{\rho_i})_{i \in I}\) is in \(\triangleright_{\tau}\).

For terms of the form \(M \overset{\text{df}}{=} \text{if } N = 0 \text{ then } P \text{ else } Q\) of type \(\tau\), by induction hypothesis \(([N]_{\rho_i})_{i \in I} \in \triangleright_{\text{int}}\), so all values \([N]_{\rho_i}\) are the same integer, say \(n\). (And this term exists because \(I\) is non-empty.) If \(n = 0\), then for every \(i \in I\), \([M]_{\rho_i}\) is then equal to \([P]_{\rho_i}\), so \(([M]_{\rho_i})_{i \in I} = ([P]_{\rho_i})_{i \in I}\) is in \(\triangleright_{\tau}\). We reason similarly if \(n \neq 0\).

For terms of the form \(M \oplus N\), of type \(D\tau\), we consider an arbitrary tuple \((h_i)_{i \in I} \in \triangleright_{\tau}^+\). By induction hypothesis \(([M]_{\rho_i})_{i \in I} \text{ and } ([N]_{\rho_i})_{i \in I}\) are in \(\triangleright_{D\tau}\), so \((\int_{V \in \tau} h_i(V) \cdot [M]_{\rho_i})_{i \in I}\) and \((\int_{V \in \tau} h_i(V) \cdot [N]_{\rho_i})_{i \in I}\) are in \(\triangleright\). Since \(\triangleright\) is convex, and integration is linear in the valuation, \((\int_{V \in \tau} h_i(V) \cdot [M \oplus N]_{\rho_i})_{i \in I}\) is also in \(\triangleright\). Since \((h_i)_{i \in I}\) is arbitrary, \(([M \oplus N]_{\rho_i})_{i \in I}\) is in \(\triangleright_{D\tau}\).

For terms of the form \(\text{ret}_{\sigma} M\), we again consider an arbitrary tuple \((h_i)_{i \in I}\)
in $\triangleright_{\pi}$. By induction hypothesis, $\langle[M]\rho_i\rangle_{i\in I}$ is in $\triangleright_{\pi}$, so by definition of $\triangleright_{\pi}$, $(h_i([M]\rho_i))_{i\in I}$ is in $\triangleright$. Equivalently, $(\int_{V\in[e]} h_i(V)\delta_1 M_{\rho_i})_{i\in I}$ is in $\triangleright$, and that means that $\langle[\mathbf{ret}_\pi M]\rho_i\rangle_{i\in I}$ is in $\triangleright_{D\pi}$.

Finally, for terms $\mathbf{bind}_{\pi,M} N$, we have $\langle[M]\rho_i\rangle_{i\in I} \in \triangleright_{D\pi}$ and $\langle[N]\rho_i\rangle_{i\in I} \in \triangleright_{\pi \rightarrow D\pi}$ by induction hypothesis, so $\langle[\mathbf{bind}_{\pi,M} N]\rho_i\rangle_{i\in I} \in \triangleright_{D\pi}$ by Step 2. □

**Proposition 6.2** For every ground PCF$_P$ term $P$: Dint $\rightarrow$ Dint $\rightarrow$ Dint, $\llbracket\mathbf{portest} P\rrbracket \preceq 8/27 \cdot \delta_0$.

**Proof.** We specialize the construction of the logical relation $(\triangleright)_{\pi \text{type}}$ to $I \overset{\text{def}}{=} \{1, 2, 3\}$ and to $\triangleright$, defined as the downward closure in $\mathbb{R}^3_+$ of the convex hull $\{a \cdot (1, 0, 1) + b \cdot (1, 1, 0) + c \cdot (0, 1, 1) \mid a, b, c \in \mathbb{R}_+, a + b + c \leq 1\}$ of the three points $\vec{\alpha}_1 \overset{\text{def}}{=} (1, 0, 1), \vec{\alpha}_2 \overset{\text{def}}{=} (1, 1, 0),$ and $\vec{\alpha}_3 \overset{\text{def}}{=} (0, 1, 1)$. The relation $\triangleright$ has an alternate description as the set of those points $(a, b, c)$ of $\mathbb{R}^3_+$ such that $a, b, c \leq 1$ and $a + b + c \leq 2$. This is depicted on the right.

The relations $\triangleright$ and $\triangleright_{\pi}$ are ternary to account for the three calls to $f$ in the definition of $\mathbf{portest}$, and $\triangleright$ is designed so that $\triangleright_{\text{Dint}}$ is as small a relation as possible that contains the triples $(\delta_1, \delta_0, 0)$ and $(\delta_1, 0, \delta_0)$. Considering the three tests $f(\delta_1)(\delta_0) = \delta_1, f(\delta_0)(0) = \delta_0$ and $f(0)(\delta_0) = \delta_0$, the triple $(\delta_1, \delta_0, 0)$ consists of the first arguments to $f$ in those tests, and the triple $(\delta_1, 0, \delta_0)$ consists of the second arguments. Hence, with $f$ bound to $P$, the triple consisting of the three values of $f(\delta_1)(\delta_i), f(\delta_0)(0)$ and $f(0)(\delta_0)$ respectively will also be contained in $\triangleright_{\text{Dint}}$, by the basic lemma of logical relations (Proposition 6.1).

We will then show that the largest probability that those values are 1, 0 and 0 respectively is $8/27$, and this will complete the proof.

First, let us check that $(\delta_1, \delta_0, 0)$ and $(\delta_1, 0, \delta_0)$ are in $\triangleright_{\text{Dint}}$. To that end, we simplify the expression of $\triangleright_{\text{Dint}}$. For all $h_1, h_2, h_3 \in \mathcal{L}[\text{int}], (h_1, h_2, h_3) \in \triangleright_{\text{int}}$ if and only if for every $n \in \llbracket\text{int}\rrbracket$, $(h_1(n), h_2(n), h_3(n)) \in \triangleright$. Next, $(a_1\delta_0 + b_1\delta_1, a_2\delta_0 + b_2\delta_1, a_3\delta_0 + b_3\delta_1)$ is in $\triangleright_{\text{Dint}}$ if and only if for all $(h_1, h_2, h_3) \in \triangleright_{\text{int}}, (a_1h_1(0) + b_1h_1(1), a_2h_2(0) + b_2h_2(1), a_3h_3(0) + b_3h_3(1)) \in \triangleright$. Since $\triangleright$ is convex and downwards-closed, it suffices to check the latter when the triples $(h_1(0), h_2(0), h_3(0))$ and $(h_1(1), h_2(1), h_3(1))$ each range over the three points $\vec{\alpha}_i, 1 \leq i \leq 3$ (nine possibilities). Let us write $\vec{\alpha}_i$ as $(\alpha_{i1}, \alpha_{i2}, \alpha_{i3})$. Hence $(a_1\delta_0 + b_1\delta_1, a_2\delta_0 + b_2\delta_1, a_3\delta_0 + b_3\delta_1)$ is in $\triangleright_{\text{Dint}}$ if and only if the nine triples $(a_1\alpha_{i1} + b_1\alpha_{j1}, a_2\alpha_{i2} + b_2\alpha_{j2}, a_3\alpha_{i3} + b_3\alpha_{j3}) (1 \leq i, j \leq 3)$ are in $\triangleright$, namely consist of non-negative numbers $\leq 1$ that sum up to a value at most 2. Verifying that this holds for $(\delta_1, \delta_0, 0)$ $(a_1 \overset{\text{def}}{=} 0, b_1 \overset{\text{def}}{=} 1, a_2 \overset{\text{def}}{=} 1, b_2 \overset{\text{def}}{=} 0, a_3 \overset{\text{def}}{=} b_3 \overset{\text{def}}{=} 0)$ and $(\delta_1, 0, \delta_0)$ $(a_1 \overset{\text{def}}{=} 0, b_1 \overset{\text{def}}{=} 1, a_2 \overset{\text{def}}{=} b_2 \overset{\text{def}}{=} 0, a_3 \overset{\text{def}}{=} 1, b_3 \overset{\text{def}}{=} 0)$ means verifying that for all $i, j$ between 1 and 3, $(\alpha_{i1}, \alpha_{i2}, 0)$ and $(\alpha_{j1}, 0, \alpha_{j3})$ are in $\triangleright$, which is obvious since those are triples of numbers equal to 0 or to 1.

Using Proposition 6.1 $(\llbracket P \rrbracket(\delta_1), \llbracket P \rrbracket(\delta_0)(0), \llbracket P \rrbracket(0)(\delta_0))$ is also in $\triangleright_{\text{Dint}}$. Let us write that triple as $(a_1\delta_0 + b_1\delta_1, a_2\delta_0 + b_2\delta_1, a_3\delta_0 + b_3\delta_1)$. Then $\llbracket\mathbf{portest} P\rrbracket$
is equal to $b_1a_2a_3 \cdot \delta_3$, as one can check. We wish to maximize $b_1a_2a_3$ subject to the constraint $(a_1\delta_0 + b_1\delta_1, a_2\delta_0 + b_2\delta_1, a_3\delta_0 + b_3\delta_1) \in \Delta_{12\text{int}}$. That constraint rewrites to the following list of twelve inequalities, not mentioning the relaxed constraints that say that each $a_i$ and each $b_i$ is non-negative:

- $a_1 + b_1$, $a_2 + b_2$, and $a_3 + b_3$ should be at most 1,

- and the nine values $a_1 + b_1 + a_2 + b_3, a_1 + b_1 + b_2 + a_3, a_1 + b_2 + a_3 + b_3, a_1 + b_1 + a_2 + b_3, a_1 + a_1 + a_2 + b_2 + b_3, a_1 + a_2 + a_3 + b_3, a_1 + a_2 + b_2 + a_3$ and $a_2 + b_2 + a_3 + b_3$ should be at most 2.

That is not manageable. To help us, we have run a Monte-Carlo simulation: draw a large number of values at random for the variables $a_i$ and $b_i$ so as to verify all constraints (using rejection sampling), and find those that lead to the largest value of $b_1a_2a_3$. That simulation gave us the hint that the maximal value of $b_1a_2a_3$ was indeed $8/27$, attained for $a_1 \overset{\text{def}}{=} 0, b_1 \overset{\text{def}}{=} 2/3, a_2 \overset{\text{def}}{=} 2/3, b_2 \overset{\text{def}}{=} 0, a_3 \overset{\text{def}}{=} 0, b_3 \overset{\text{def}}{=} 2/3$. We now have to verify that formally. Knowing which values of $a_i$ and $b_i$ maximize $b_1a_2a_3$ allows us to select which constraints are the important ones, and then one can simplify slightly further.

In order to obtain a formal argument, we therefore choose to maximize $b_1a_2a_3$ with respect to the relaxed constraints that $a_1 + b_1 + a_2 + b_2 + a_3 + b_3 \leq 2$ (an inequality implied by all the above constraints), all numbers being non-negative. This will give us an upper bound, which may fail to be optimal (but won’t).

In order to do so, we first maximize $c_1c_2c_3$ under the constraints $c_1, c_2, c_3 \geq 0$ and $c_1 + c_2 + c_3 \leq 2$. Rewrite $c_1$ as $d(1 - r)$, $c_2$ as $dr(1 - s)$, and $c_3$ as $drs$, where $d \leq 2$ and $r, s \in [0, 1]$. (Namely, let $d \overset{\text{def}}{=} c_1 + c_2 + c_3$; if $d = 0$, let $r$ and $s$ be arbitrary; otherwise, let $r \overset{\text{def}}{=} 1 - c_1/d$; if $r = 0$, then let $s$ be arbitrary; otherwise, let $s \overset{\text{def}}{=} c_3/(dr)$.) The maximal value of $c_1c_2c_3 = d^3(1 - r)^2 s(1 - s)$ is obtained by maximizing:

- $d$ (as 2),

- $(1 - r)^2$ when $r \in [0, 1]$ (value 4/27 obtained at $r \overset{\text{def}}{=} 2/3$, see Figure 5 left),

![Figure 5: Maximizing $(1 - r)r^2$ and $s(1 - s)$](image-url)
• and $s(1-s)$ when $s \in [0, 1]$ (value 1/4 obtained at $s = 1/2$, see Figure 5 right).

hence is equal to $2 \cdot (4/27) \cdot (1/4) = 8/27$. It follows that for all $a_1, b_1, a_2, b_2, a_3, b_3 \in [0, 1]$ such that $a_1+b_1+a_2+b_2+a_3+b_3 \leq 2$, $b_1a_2a_3 \leq (a_1+b_1)(a_2+b_2)(a_3+b_3) \leq 8/27$, by taking $c_i \overset{\text{def}}{=} a_i + b_i$ for each $i$.

We sum up our results as follows. Note that $\Pr[\text{portest } P \downarrow n] = 0$, for any $P$, if $n \neq 0$.

**Theorem 6.3** For every ground PCF$_P$ term $P : \text{Dint} \to \text{Dint} \to \text{Dint}$, the probability $\Pr[\text{portest } P \downarrow 0]$ that $P$ fools the parallel or tester never exceeds $8/27$. That bound is attained by taking $P \overset{\text{def}}{=} \text{pmor}$. □

7 Conclusion and Related Work

There is an extensive literature on the semantics of higher-order functional languages, and extensions that include probabilistic choice are now attracting attention more than ever.

Concerning denotational semantics, we should cite the following. *Probabilistic coherence spaces* provide a fully abstract semantics for a version of PCF with probabilistic choice, as shown by Ehrhard, Tasson, and Pagani [ETP14]. *Quasi-Borel spaces* and predomains have recently been used to give adequate semantics to typed and untyped probabilistic programming languages, see e.g. [VKS19]. *QCB spaces* form a convenient category in which various effects, including probabilistic choice, can be modeled [Bat06]. Comparatively, the domain-theoretic semantics we are using in this paper is rather mundane, and I have used similar models for further extensions that also include angelic [Gou15] and demonic [Gou19b] non-deterministic choice. In those papers, I obtain full abstraction at the price of adding some extra primitives, but also of considering a richer semantics that also includes forms of non-deterministic choice. The latter allows us to work in categories with nice properties. That is not available in the context of PCF$_P$, because there is no known Cartesian-closed category of continuous dcpos that is closed under $V_{\leq 1}$ [JT98].

Let me remind the reader that denotational semantics is only a tool here: the result we have presented concerns the operational semantics, and domain-theory is only used, through adequacy, in order to bound $\Pr[\text{portest } P \downarrow s]$. One may wonder whether a direct operational approach would work, but I doubt it strongly. Eventually, any operational approach would have to find suitable invariants, and such invariants will be hard to distinguish from an actual denotational semantics.

One may wonder whether such semantical proofs would be useful in the realm of probabilistic process algebras as well. In non-probabilistic process algebras, syntactic reasoning is usually enough, using bisimulations and up-to techniques. The case of probabilistic processes is necessarily more complex, and may benefit from such semantical arguments.
References

[AJ94]  Samson Abramsky and Achim Jung. Domain theory. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science vol. III*, pages 1–168. Oxford University Press, 1994.

[Bat06]  Ingo Battenfeld. Computational effects in topological domain theory. *Electronic Notes in Theoretical Computer Science*, 158:59–80, 2006.

[Bir98]  B. Bird. *Introduction to Functional Programming using Haskell*. Prentice-Hall Series in Computer Science, 1998.

[ETP14]  Thomas Ehrhard, Christine Tasson, and Michele Pagani. Probabilistic coherence spaces are fully abstract for probabilistic PCF. In Suresh Jagannathan and Peter Sewell, editors, *Proc. 41st Ann. ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL ’14)*, pages 309–320, 2014.

[GHK+03]  Gerhard Gierz, Karl Heinrich Hofmann, Klaus Keimel, Jimmie D. Lawson, Michael Mislove, and Dana Stewart Scott. *Continuous Lattices and Domains*, volume 93 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 2003.

[Gou13]  Jean Goubault-Larrecq. *Non-Hausdorff Topology and Domain Theory—Selected Topics in Point-Set Topology*, volume 22 of *New Mathematical Monographs*. Cambridge University Press, 2013.

[Gou15]  Jean Goubault-Larrecq. Full abstraction for non-deterministic and probabilistic extensions of PCF I: the angelic cases. *Journal of Logic and Algebraic Methods in Programming*, 84(1):155–184, January 2015.

[Gou19a]  Jean Goubault-Larrecq. Fooling the parallel or tester with probability 8/27. arXiv, 2019.

[Gou19b]  Jean Goubault-Larrecq. A probabilistic and non-deterministic call-by-push-value language. In Zhong Shao, editor, *Proceedings of the 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS’19)*, 2019. Full version on arXiv:1812.11573 [cs.LO].

[GPT07]  Jean Goubault-Larrecq, Catuscia Palamidessi, and AngeloTroina. A probabilistic applied pi-calculus. In Zhong Shao, editor, *Proceedings of the 5th Asian Symposium on Programming Languages and Systems (APLAS’07)*, volume 4807 of *Lecture Notes in Computer Science*, pages 175–290, Singapore, November-December 2007. Springer.

[Jon90]  Claire Jones. *Probabilistic Non-Determinism*. PhD thesis, University of Edinburgh, 1990. Technical Report ECS-LFCS-90-105.
A Soundness

There is a unique way of defining a denotational semantics $\llbracket C \rrbracket \rho$ of contexts $C$ in such a way that $\llbracket C[M] \rrbracket \rho = \llbracket C \rrbracket \rho(\llbracket M \rrbracket \rho)$ for every $M$ of the right type and every $\rho \in \text{Env}$. For $C \equiv E_0 E_1 \cdots E_n$, $\llbracket C \rrbracket \rho$ is the composition of the maps $\llbracket E_0 \rrbracket \rho, \llbracket E_1 \rrbracket \rho, \ldots, \llbracket E_n \rrbracket \rho$, where for each elementary or initial context $E$, $\llbracket E \rrbracket \rho$ is defined by:

- for every $N: \sigma$, $\llbracket \_ N \rrbracket \rho(f) \equiv f(\llbracket N \rrbracket \rho)$;
- $\llbracket p_\_ \rrbracket \rho(n) \equiv n - 1$, $\llbracket s_\_ \rrbracket \rho(n) \equiv n + 1$;
- $\llbracket \text{if } _ = 0 \text{ then } N \text{ else } P \rrbracket \rho(n) \equiv \llbracket N \rrbracket \rho$ if $n = 0$, $\llbracket P \rrbracket \rho$ otherwise;
- $\llbracket \text{bind}_{\sigma, \tau} \_ N \rrbracket \rho \equiv (\llbracket N \rrbracket \rho)^\dagger$;
- $\llbracket \text{ret}_{\tau} \_ \rrbracket \rho(n) \equiv \eta(n) = \delta_n$;
- $\llbracket \_ \rrbracket \rho(\nu) \equiv \nu$;

It is standard that $\llbracket M \rrbracket \rho$ only depends on the value of $\rho$ on the free variables of $M$ (if $\rho(x) = \rho'(x)$ for every free variable $x$ of $M$, then $\llbracket M \rrbracket \rho = \llbracket M \rrbracket \rho'$), and
that for every substitution $\theta \overset{\text{def}}{=} [x_1 := N_1, \ldots, x_n := N_n]$, $\|M\rho\| = \|M\| (\|\theta\| \rho)$, where $\|\theta\|$ is the environment that maps every $x_i$, $1 \leq i \leq n$, to $\|N_i\| \rho$ and all other variables $y$ to $\rho(y)$. In particular, $\|M[x_\sigma := N]\| \rho = \|M\| (\rho[x_\sigma \mapsto \|N\| \rho])$.

Finally, $\|(\lambda x_\sigma . M) N\| \rho$ is equal to $\|M[x_\sigma := N]\| \rho$. We have:

**Lemma A.1** Let $\rho$ be an environment.

1. For every rule of the form $s \overset{\tau}{\rightarrow} t$, $\|s\| \rho = \|t\| \rho$.

2. For every context $C$ of type $D \sigma \vdash \text{Dint}$, for all $M, N : D \sigma$, $\|C \cdot M \oplus N\| \rho = \frac{1}{2} \|C \cdot M\| \rho + \frac{1}{2} \|C \cdot N\| \rho$.

**Proof.** 1. All the cases are easily checked, except perhaps for the rule $C[\text{bind}_{\sigma, \tau, \_N}] \cdot \text{ret}_\sigma M \overset{\tau}{\rightarrow} C \cdot N M$. That reduces to showing the equality $\|C[\text{bind}_{\sigma, \tau \_N}] \cdot \text{ret}_\sigma M\| \rho = \|\text{ret}_\sigma M\| \rho$. The left-hand side is $(\|N\| \rho)^\dagger(\|\text{ret}_\sigma M\| \rho)$, which is equal to $\|N\| \rho(\|M\| \rho)$, by $\mathbb{1}$. In turn, that is $\|\text{ret}_\sigma M\| \rho$.

2. Let $C \overset{\text{def}}{=} E_0 E_1 \cdots E_n$. By inspection of types, all the elementary contexts $E_i$, $1 \leq i \leq n$, must be of the form $C[\text{bind}_{\sigma_i+1, \sigma_i \_N}]$ for some $N_i : \sigma_i+1 \rightarrow D \sigma_i$, $E_0 = \mathbb{1}$, and $\sigma_1 = \text{int}$.

We observe that $\|E_i\| \rho = (\|N_i\| \rho)^\dagger$ is a linear map. In fact, $f^\dagger$ is linear for every Scott-continuous map $f : X \rightarrow Y$, in the following sense: for all $a, b \in \mathbb{R}_+$ with $a + b \leq 1$, for all $\mu, \nu \in Y$, $f^\dagger (a \mu + b \nu) = a f^\dagger (\mu) + b f^\dagger (\nu)$. Indeed, for every $V \in \mathcal{O}_Y$, $f^\dagger (a \mu + b \nu) (V) = \int_{x \in X} f(x)(V) d(\mu + b \nu) = \int_{x \in X} f(x)(V) d\mu + b \int_{x \in X} f(x)(V) d\nu = a f^\dagger (\mu) (V) + b f^\dagger (\nu) (V)$.

It follows that $\|C\| \rho$ is also a linear map. Then $\|C \cdot M \oplus N\| \rho = \|C\| \rho (\frac{1}{2} \|M\| \rho + \frac{1}{2} \|N\| \rho) = \frac{1}{2} \|C\| \rho (\|M\| \rho) + \frac{1}{2} \|C\| (\|N\| \rho) = \frac{1}{2} \|C \cdot M\| \rho + \frac{1}{2} \|C \cdot N\| \rho$. \hfill $\square$

**Proposition A.2 (Soundness)** For every configuration $s$ of type $\text{Dint}$, for every $n \in \mathbb{Z}$, for every environment $\rho$, $\|s\| \rho(\{n\}) \geq \Pr[s \downarrow n]$.

**Proof.** It suffices to show that for every $r \in \mathbb{R}_+$ such that $r < \Pr[s \downarrow n]$, $r \leq \|s\| \rho(\{n\})$. We write $\Pr[s \downarrow V]$ as a possibly infinite sum. Since $r < \Pr[s \downarrow n]$, there is a finite subset of the summands which sum to at least $r$. In other words, there is a finite set of traces starting at $s$ and ending at $[\text{ret}_\text{int} \_n] \cdot n$, whose weights sum up to at least $r$. Let $N$ be some upper bound on the lengths of those traces. By induction on $N$, we show that the sum $\Pr_{\leq N}[s \downarrow n]$ of all weights of traces of length at most $N$, starting at $s$ and ending at $[\text{ret}_\text{int} \_n] \cdot n$, is less than or equal to $\|s\| \rho(\{n\})$, and this will prove the claim.

If $s = [\text{ret}_\text{int} \_n] \cdot n$, then $\|s\| \rho(\{n\}) = \delta_{\text{int}} \rho(\{n\}) = 1$. Therefore $\Pr_{\leq N}[s \downarrow n] \leq \|s\| \rho(\{n\})$.

From now on, we assume that $s$ is not of the form $[\text{ret}_\text{int} \_n] \cdot n$.

If $N = 0$, then there is no trace of length at most $N$ starting at $s$ and ending at $[\text{ret}_\text{int} \_n] \cdot n$, so $\Pr_{\leq N}[s \downarrow n] = 0 \leq \|s\| \rho(\{n\})$.

If $N \geq 1$, then we explore three cases.

If no rule applies to $s$, namely if $s \overset{a}{\rightarrow} t$ for no $a$ and no $t$, then $\Pr_{\leq N}[s \downarrow n] = 0 \leq \|s\| \rho(\{n\})$.  

18
If $s$ if of the form $C \cdot M \oplus N$ then $\Pr_{\leq N-1}(C \cdot M \downarrow n) \leq \|C \cdot M\| \rho(\{n\})$ and $\Pr_{\leq N-1}(C \cdot N \downarrow n) \leq \|C \cdot N\| \rho(\{n\})$, by induction hypothesis. Now $\Pr_{\leq N}(s \downarrow n) = \frac{1}{2} \Pr_{\leq N-1}(C \cdot M \downarrow n) + \frac{1}{2} \Pr_{\leq N-1}(C \cdot N \downarrow n)$, which is less than or equal to $\frac{1}{2} \|C \cdot M\| \rho(\{n\}) + \frac{1}{2} \|C \cdot N\| \rho(\{n\}) = \|C \cdot M \oplus N\| \rho(\{n\})$, by Lemma A.1 item 2.

In all other cases, $s \xrightarrow{t} t$ for some unique configuration $t$, so that $\Pr_{\leq N}(s \downarrow n) = \Pr_{\leq N-1}(t \downarrow n) \leq \|s\| \rho(\{n\})$, by induction hypothesis. By Lemma A.1 item 1, the latter is equal to $\|s\| \rho(\{n\})$.

## B Adequacy

The key to proving the converse of soundness is the design of a suitable logical relation $R^\text{def}_{\tau}$ of type $\tau$, where each $R_{\tau}$ is a binary relation between ground terms $M$ of type $\tau$ and elements of $\|\tau\|$. Since $\|M\| \rho$ does not depend on $\rho$ when $M$ is ground, we simply write $\|M\|$ in that case. We write $\|C\|$ similarly for ground contexts $C$.

The definition of $R_{\tau}$ is by induction on $\tau$, using auxiliary relations $R^1_{\tau}$ between ground contexts $C : D\tau \rightarrow \text{Dint}$ and Scott-continuous maps $h : \|D\tau\| \rightarrow \|\text{Dint}\|:

- for all ground $M : \text{int}$ and $n \in \mathbb{Z}$, $M R_{\text{int}} n$ if and only if $\{n\} : M \xrightarrow{\tau} \dom[n]$;
- for all types $\sigma, \tau$, for all ground $M : \sigma \rightarrow \tau$ and $f \in \|\sigma \rightarrow \tau\|$, $M R_{\sigma \rightarrow \tau} f$ if and only if for all $N R_{\sigma} a$, $MN R_{\tau} f(a)$ (we say “for all $N R_{\sigma} a$” instead of “for every ground $N : \sigma$ and for every $a \in \|\sigma\|$ such that $N R_{\sigma} a$”);
- for every type $\tau$, for all ground $M : D\tau$ and $\nu \in \|D\tau\|$, $M R_{D\tau} \nu$ if and only if for every ground context $C : D\tau \vdash \text{Dint}$, for every Scott-continuous map $h : \|D\tau\| \rightarrow \|\text{Dint}\|$ such that $C R^1_{D\tau} h$, for every $n \in \mathbb{Z}$, $\Pr[C \cdot M \downarrow n] \geq h(\nu)(\{n\})$;
- for every type $\tau$, for every ground context $C : D\tau \vdash \text{Dint}$, for every Scott-continuous map $h : \|D\tau\| \rightarrow \|\text{Dint}\|$, $C R^1_{D\tau} h$ if and only if for all $P R_{\tau} a$, for every $n \in \mathbb{Z}$, $\Pr[C \cdot \text{ret}_{\tau} P \downarrow n] \geq h(\nu(a))(\{n\})$.

**Lemma B.1** If $C : M \xrightarrow{\tau} C \cdot N$ by any sequence of rules except the rule $\dom[] : \text{ret}_{\text{int}} P \xrightarrow{\tau} \text{ret}_{\text{int}} \dom[] : P$, then for every context $C'$ of the expected type, $C' C : M \xrightarrow{\tau} C' C \cdot N$.

**Proof.** It suffices to show the claim under the assumption that $C : M \xrightarrow{\tau} C \cdot N$ by any other rule than the one we excluded. This is clear, since no rule except the one we excluded requires the context to have any specific shape. \qed

**Lemma B.2** For every context $C : \sigma \vdash \tau$, for every term $M : \sigma$,

1. $\dom[] : C[M] \xrightarrow{\tau} C \cdot M$ by using the exploration rules only;

19
2. the run starting at \([\_] \cdot C[M]\) must start with the trace \([\_] \cdot C[M] \xrightarrow{\beta} \ast C \cdot M\), followed by the run starting at \(C \cdot M\).

Proof. 1 is clear. 2 is because the operational semantics is deterministic, in the sense that \(s \xrightarrow{\beta} t_0\) and \(s \xrightarrow{\beta} t_1\) implies \(t_0 = t_1\). \(\square\)

**Lemma B.3** For every context \(C: \sigma \vdash \tau\), if \(\sigma\) is a computation type, then so is \(\tau\).

Proof. By inspection of the elementary contexts.

**Lemma B.4** For every configuration \(s\) of type \(\text{int}\), every trace \(\xrightarrow{\alpha} s'\) satisfies \(\alpha = 1\). Moreover, it does not use the rule \([\_] \cdot \text{ret}_\beta P \xrightarrow{\beta} [\text{ret}_\beta \_] \cdot P\).

Proof. It is enough to show the claim under the assumption that \(s \xrightarrow{\alpha} s'\). Let us write \(s\) as \(C \cdot M\), where \(C\) is of type \(\sigma \vdash \beta\) and \(M: \sigma\). By Lemma B.3, \(\sigma\) cannot be a computation type. It follows that the rule that was used cannot be \(C \cdot P \oplus Q \xrightarrow{1/2} C \cdot P\) or \(C \cdot P \oplus Q \xrightarrow{1/2} C \cdot Q\), since \(P \oplus Q\) has a computation type. Similarly, it cannot be \([\_] \cdot \text{ret}_\beta P \xrightarrow{\beta} [\text{ret}_\beta \_] \cdot P\), again because \(\text{ret}_\beta P\) has a computation type.

**Lemma B.5** For all terms \(M: \tau\) and \(N: \sigma\), for every context \(C': \sigma \vdash \tau\), for every \(V \in \int\), if \([\_] \cdot M \xrightarrow{\beta} \ast C' \cdot N\) without using the rule \([\_] \cdot \text{ret}_\beta P \xrightarrow{\beta} [\text{ret}_\beta \_] \cdot P\), and if \(C' \cdot N \xrightarrow{\beta} V\), then \(M \xrightarrow{\beta} V\).

Proof. By induction on \(\tau\). If \(\tau = \text{int}\), \(C' \cdot N \xrightarrow{\beta} V\) means that \([\_] \cdot C' \cdot N \xrightarrow{\beta} \ast [\_] \cdot V\). By Lemma B.2, item 2, our trace starting at \([\_] \cdot C' \cdot N\) and ending at \([\_] \cdot V\) must factor as \([\_] \cdot C' \cdot N \xrightarrow{\beta} \ast [\_] \cdot V\). Hence \([\_] \cdot M \xrightarrow{\beta} \ast C' \cdot N \xrightarrow{\beta} \ast [\_] \cdot V\), showing that \(M \xrightarrow{\beta} V\).

For types of the form \(D\tau\), our task is to show that \(M \xrightarrow{\beta} V\), where \(\nu\) is any subprobability valuation in \([D\tau]\), knowing that \(C' \cdot N \xrightarrow{\beta} \nu\). We let \(C: D\tau \vdash \text{Dint}\) be an arbitrary ground context, \(h: [D\tau] \rightarrow [\text{Dint}]\) be an arbitrary Scott-continuous map such that \(C \xrightarrow{1/2} h\), and we wish to show that for every \(n \in \int\), \(\Pr[C \cdot M \downarrow n] \geq h(\nu)(\{n\})\). By Lemma B.1, \(C \cdot M \xrightarrow{\beta} \ast CC' \cdot N\). Since \(C' \cdot N \xrightarrow{\beta} \nu\), \(\Pr[C \cdot C' \cdot N \downarrow n] \geq h(\nu)(\{n\})\). By Lemma B.2, item 2, the run starting at \(C \cdot C' \cdot N\) must factor as a trace \(C \cdot C' \cdot N \xrightarrow{\beta} \ast CC' \cdot N\) followed by a run starting at \(CC' \cdot N\), so \(\Pr[C \cdot C' \cdot N \downarrow n] = \Pr[CC' \cdot N \downarrow n]\). Prepending instead the trace \(C \cdot M \xrightarrow{\beta} \ast CC' \cdot N\) (i.e., using Lemma 3.3, item 2), we see that \(\Pr[C \cdot M \downarrow n] = \Pr[CC' \cdot N \downarrow n]\). That is equal to \(\Pr[C \cdot C' \cdot N \downarrow n]\), which is larger than or equal to \(h(\nu)(\{n\})\) since \(C' \cdot N \xrightarrow{\beta} \nu\) and \(C \xrightarrow{1/2} h\).

For function types \(\sigma \rightarrow \tau\), we wish to show that \(M \xrightarrow{\beta} f\), where \(f \in [\sigma \rightarrow \tau]\), knowing that \(C' \cdot N \xrightarrow{\beta} \nu\). The latter means that for all \(P \xrightarrow{\beta} f\), \(C'[\_]P \xrightarrow{\beta} f(a)\). For every \(C\), there is a trace \(C \cdot MP \xrightarrow{\beta} C[\_] \cdot M \xrightarrow{\beta} \ast C[\_][P]C' \cdot N\), by Lemma B.1 with context \(C[\_]\), and this trace does not use
the rule \([_] \cdot \text{ret}_\beta Q \xrightarrow{1} [\text{ret}_\beta _] \cdot Q\). By induction hypothesis (using \([_] \cdot P[C']\) instead of \(C'\), \(MP R, f(b)\)). Since \(P\) and \(b\) are arbitrary, \(M R_\sigma \rightarrow f\). 

By taking \(C' \overset{\text{def}}{=} [\_]\), we obtain the following.

**Corollary B.6** Let \(M, N : \tau\), and \(V \in \llbracket \tau \rrbracket\). If \([_] \cdot M \xrightarrow{1} \tau [\_] \cdot N\) by any sequence of rules except \([_] \cdot \text{ret}_\text{int} P \xrightarrow{1} [\text{ret}_\text{int} _] \cdot P\), and if \(N R, V\) then \(M R, V\).

**Proof.** By induction on \(\tau\). When \(\tau = \text{int}\), this is obvious. Let us consider the case of types of the form \(D\tau\). For every ground context \(C: D\tau \vdash \text{Dint}\), for every Scott-continuous map \(h: \llbracket D\tau \rrbracket \rightarrow \llbracket \text{Dint} \rrbracket\) such that \(C R_D h\), for every \(n \in \llbracket \text{int} \rrbracket\), the set \(\Gamma_{C, h, n} \overset{\text{def}}{=} \{\nu \in \llbracket D\tau \rrbracket \mid h(\nu)\{\{n\}\} \leq \text{Pr}[C \cdot M \downarrow n]\}\) is Scott-closed: it is easily seen to be downwards-closed, and for every directed family \((\nu_i)_{i \in I}\) the set \(\sum_{i \in I} h(\nu_i)\{\{n\}\} \leq \text{Pr}[C \cdot M \downarrow n]\) is Scott-closed; it is easily seen to be downwards-closed, and for every directed family \((\nu_i)_{i \in I}\), \(\sum_{i \in I} h(\nu_i)\{\{n\}\} \leq \text{Pr}[C \cdot M \downarrow n]\), hence is Scott-closed as well. It also contains the least element \(\bot\) of \(\llbracket \tau \rrbracket\).

Additionally, for every \(\text{directed family}\) \((\nu_i)_{i \in I}\) in \(\Delta_{N,a}\), \((\sum_{i \in I} h(\nu_i)\{\{n\}\}) = \sum_{i \in I} h(\nu_i)\{\{n\}\} \leq \text{Pr}[C \cdot M \downarrow n]\), so \(\sup_{i \in I} \nu_i \in \Gamma_{C, h, n}\). \(M R_D\) is the intersection of all the sets \(\Gamma_{C, h, n}\), hence is Scott-closed as well.

When \(\sigma \rightarrow \tau\) is a computation type, \(\tau\) is one, too, and by induction hypothesis \(MN R_\tau \perp\) for all \(N R_\sigma a\). That means that \(MN R_\tau \perp_\sigma \) for all \(N R_\sigma a\), hence that \(M R_\sigma \rightarrow \perp_\sigma \rightarrow_{\tau}\).

**Corollary B.8** For every computation type \(\tau\), for all \(M R_\sigma \rightarrow \tau f\), \(\text{rec}_\tau M R_\tau \text{lf}_{\llbracket \tau \rrbracket} f\).

**Proof.** By the second part of Lemma B.7, \(\text{rec}_\tau M \theta R_\tau \perp\).

Additionally, for every \(a \in \llbracket \tau \rrbracket\), if \(\text{rec}_\tau M R_\tau a\) then \(M(\text{rec}_\tau M) R_\tau f(a)\), since \(M R_\sigma \rightarrow \tau f\). Using Corollary B.6 with the step \([_] \cdot \text{rec}_\tau M \xrightarrow{1} M(\text{rec}_\tau M)\), it follows that \(\text{rec}_\tau M R_\tau f(a)\).

Hence for every \(a \in \text{rec}_\tau M R_\tau f(a)\) is also in \(\text{rec}_\tau M R_\tau\). It follows that \(f^n(\perp_\tau)\) is in \(\text{rec}_\tau M R_\tau\) for every \(n \in \mathbb{N}\). By Lemma B.7, \(\sup_{n \in \mathbb{N}} f^n(\perp_\tau)\) must also be in \(\text{rec}_\tau M R_\tau\), and that is just \(\text{lf}_{\llbracket \tau \rrbracket} f\). 

**Lemma B.9** Let \(\sigma\) be a type. For all \(M R_\sigma a\), \(\text{ret}_\sigma M R_{\sigma \tau} \eta(a)\).

21
Proof. Relying on the definition of $R_{D\sigma}$, let $\beta$ be a basic type, $C : D\sigma \vdash D\beta$ be a ground context, $h : [D\sigma] \rightarrow [D\beta]$ be Scott-continuous, and assume that $C R_{D\sigma}^{\bot} h$. By definition of $R_{D\sigma}^{\bot}$, and since $M R_{\sigma} a$, we obtain $\Pr(C \cdot \text{ret}_\sigma M \downarrow V) \geq h(\eta(a))$, and that is what we wanted to show.

**Lemma B.10** Let $\sigma, \tau$ be types. For all $M R_{D\sigma} \nu$ and $N R_{\sigma \rightarrow D\tau} f$, we have $\text{bind}_{\sigma,\tau} MN R_{D\tau} f^!(\nu)$.

Proof. We plan to use the definition of $R_{D\sigma}$, and for that we fix an arbitrary ground context $C : D\tau \vdash \text{Dint}$, an arbitrary Scott-continuous map $h : [D\tau] \rightarrow [\text{Dint}]$ such that $C R_{D\tau}^{\bot} h$, and we wish to show: $(\ast)$ for every $n \in \mathbb{Z}$, $\Pr(C \cdot \text{bind}_{\sigma,\tau} MN \downarrow n) \geq h(f^!(\nu))(\{n\})$.

For all $P R_{\sigma} a$, by definition of $R_{\sigma \rightarrow D\tau}$, we have $NP R_{D\tau} f(a)$. Since $C R_{D\tau}^{\bot} h$, and using the definition of $R_{D\tau}$, we obtain that $\Pr(C \cdot NP \downarrow n) \geq h(f^!(\nu))(\{n\})$ for every $n \in \mathbb{Z}$. We note that $C[\text{bind}_{\sigma,\tau} \_ \cdot N] \cdot \text{ret}_\sigma P \vdash C \cdot NP$ and we use Lemma 3.3 item 2, so $\Pr(C[\text{bind}_{\sigma,\tau} \_ \cdot N] \cdot \text{ret}_\sigma P \downarrow n) \geq h(f^!(\nu))(\{n\})$. By Lemma 3.3, $f = f^!\tau P$, so $\Pr(C[\text{bind}_{\sigma,\tau} \_ \cdot N] \cdot \text{ret}_\tau P \downarrow n) \geq h(f^!(\eta(a)))(\{n\})$. Since $n, P$ and $a$ are arbitrary such that $P R_{\sigma} a$, we obtain that $C[\text{bind}_{\sigma,\tau} \_ \cdot N] R_{D\sigma}^{\bot} h \circ f^!$, by definition of $R_{D\sigma}^{\bot}$.

From that and $M R_{D\sigma} \nu$, it follows that, for every $n \in \mathbb{Z}$, $\Pr(C[\text{bind}_{\sigma,\tau} \_ \cdot N] \cdot MN \downarrow n) \geq h(f^!(\nu))(\{n\})$. Since $C \cdot \text{bind}_{\sigma,\tau} MN \vdash C[\text{bind}_{\sigma,\tau} \_ \cdot N].M$, and using Lemma 3.3 item 2, we obtain $\Pr(C \cdot \text{bind}_{\sigma,\tau} MN \downarrow n) \geq h(f^!(\nu))(\{n\})$. Since $n, C$ and $\tau$ are arbitrary such that $C R_{D\tau}^{\bot} h$, $\text{bind}_{\sigma,\tau} MN R_{D\tau} f^!(\nu).$ $\square$

The crucial property of logical relations is the following basic lemma of logical relations. For a ground substitution $\theta \overset{\text{def}}{=} [x_1 := N_1, \ldots, x_k := N_k]$ and an environment $\rho$, we write $\theta R_{\nu} \rho$ to mean that for every $i, 1 \leq i \leq k, N_i R_{\tau_i} \rho(x_i)$, where $\tau_i$ is the type of $x_i$. The following is the basic lemma of logical relations for the case at hand.

**Proposition B.11** For every PCF$_\varpi$ term $M : \tau$, for every ground substitution $\theta$ such that all the free variables of $M$ are in $\text{dom} \theta$, and for every environment $\rho$ such that $\theta R_{\nu} \rho, M \theta R_{\nu} [M] \rho$.

Proof. This is by induction on the structure of $M$. If $M = x_i$ for some $i$, $1 \leq i \leq n$ (where $\theta = [x_1 := N_1, \ldots, x_n := N_n]$), then this follows from the assumption $\theta R_{\nu} \rho$.

If $M$ is a constant $n \in \mathbb{Z}$, then $n R_{\text{int}} n$, because $[\_].n \rightarrow^{\ast} [\_].n$, trivially. If $M = sN$, then by induction hypothesis $N \theta R_{\text{int}} n$, where $n \overset{\text{def}}{=} [N] \rho$. Therefore $[\_].N \rightarrow^{\ast} [\_].n$. By Lemma B.4 that trace does not use the rule $[\_].\text{ret}_\beta P \rightarrow [\text{ret}_\beta \_].P$. We can therefore apply Lemma B.5 to the effect that $[s\_].N \rightarrow^{\ast} [s\_].n$. Then $[\_].sM \rightarrow [s\_].N \rightarrow^{\ast} [s\_].n \rightarrow^{\ast} [\_].n + 1 = [M] \rho$. We reason similarly if $M = pN$.

In the case of terms of the form if $M = 0$ then $N$ else $P$, we must show that if $M \theta = 0$ then $N \theta$ else $P \theta R_{\nu} [\text{if } M = 0 \text{ then } N \text{ else } P] \rho$, knowing that $M \theta R_{\text{int}} [M] \rho, N \theta R_{\nu} [N] \rho$ and $P \theta R_{\nu} [P] \rho$ by induction hypothesis.

22
Let $n \defeq \|M\| \rho$. Since $M \theta R_{\text{int}} n$, we have a trace $\lfloor \cdot \rfloor \cdot M \xrightarrow{1}^* \lfloor \cdot \rfloor \cdot n$, which cannot use the rule $\lfloor \cdot \rfloor \cdot \text{ret}_\beta Q \xrightarrow{1} \lfloor \text{ret}_\beta \cdot \rfloor \cdot Q$ by Lemma B.4. Hence $\lfloor \cdot \rfloor \cdot M \xrightarrow{1}^* \lfloor \cdot \rfloor \cdot n$, and therefore $M \theta = 0$ or $N \theta$ else $P \theta \xrightarrow{1}^* \lfloor \cdot \rfloor \cdot n$ by using an additional instance of the leftmost exploration rule. If $n = 0$, by doing one more computation step, we obtain that $M \theta = 0$ or $N \theta$ else $P \theta \xrightarrow{1}^* \lfloor \cdot \rfloor \cdot N \theta$, still not using the rule $\lfloor \cdot \rfloor \cdot \text{ret}_\beta Q \xrightarrow{1} \lfloor \text{ret}_\beta \cdot \rfloor \cdot Q$. We now use Lemma B.5 and we obtain that if $M \theta = 0$ then $N \theta$ else $P \theta R_{\tau} \lfloor N \rfloor \rho = \|M\| \rho$. When $n \neq 0$, we reason similarly and we obtain that if $M \theta = 0$ then $N \theta$ else $P \theta R_{\tau} \lfloor P \rfloor \rho = \|M\| \rho$.

In the case of applications, we must show that $(MN) \theta R_{\tau} \lfloor M \rfloor \rho (\lfloor N \rfloor \rho)$. This follows from the definition of $R_{\sigma \rightarrow \tau}$, since by induction hypothesis $M \theta R_{\sigma \rightarrow \tau} \lfloor M \rfloor \rho$ and $N \theta R_{\tau} \lfloor N \rfloor \rho$.

In the case of abstractions, we must show that $(\lambda x_\sigma.M) \theta R_{\sigma \rightarrow \tau} \lfloor \lambda x_\sigma.M \rfloor \rho$. We write $\theta$ as $[x_1 := N_1, \ldots, x_k := N_k]$, fix an arbitrary ground term $N : \sigma$, and a value $a \in \|\sigma\|$ such that $N \sigma a$. We rename $x_\sigma$ to a fresh variable if necessary, and we define $\theta'$ as $[x_1 := N_1, \ldots, x_k := N_k, x_\sigma := N]$, so that $(\lambda x_\sigma.M) \theta = \lambda x_\sigma.M \theta$ and $M \theta' = M \theta[x_\sigma := N]$. We must show that $(\lambda x_\sigma.M) \theta R_{\tau} \lfloor M \rfloor (\rho[x_\sigma \mapsto a])$. By induction hypothesis, $M \theta' R_{\tau} \lfloor M \rfloor (\rho[x_\sigma \mapsto a])$. We now apply Corollary B.6 noticing that $\lfloor \cdot \rfloor \cdot (\lambda x_\sigma.M) \theta) N \xrightarrow{1} \lfloor N \rfloor \cdot \lambda x_\sigma.M \theta \xrightarrow{1} \lfloor \cdot \rfloor \cdot M \theta[x_\sigma := N] = M \theta'$. This allows us to conclude that $(\lambda x_\sigma.M) \theta R_{\tau} \lfloor M \rfloor (\rho[x_\sigma \mapsto a])$, as desired.

Let us deal with terms of the form $M \oplus N$, of type $D\tau$. We must show that for every ground context $C : D\tau \vdash D\text{int}$, for every Scott-continuous map $h : \|D\tau\| \rightarrow \|D\text{int}\|$ such that $C R_{\tau \rightarrow \tau} h$, for every $n \in \mathbb{Z}$, $\Pr[C \cdot M \oplus N \downarrow n] \geq h(\nu)(\{n\})$. By induction hypothesis, $M \theta R_{D\tau} \lfloor M \rfloor \rho$, so $\Pr[C \cdot M \downarrow n] \geq h(\nu)(\{n\})$. Similarly, $\Pr[C \cdot N \downarrow n] \geq h(\nu)(\{n\})$. By Lemma 3, item 3,

$$\Pr[C \cdot (M \oplus N) \theta \downarrow n] = \frac{1}{2} \Pr[C \cdot M \theta \downarrow n] + \frac{1}{2} \Pr[C \cdot N \theta \downarrow n] \geq \frac{1}{2} h(\nu)(\{n\}) + \frac{1}{2} h(\nu)(\{n\}) = h(\nu)(\{n\}).$$

The case of terms of the form $\text{rec}_\tau M$, $\text{ret}_\tau M$ and $\text{bind}_{\alpha,\tau} M$ follow from Corollary B.8, Lemma B.9 and Lemma B.10 respectively. \hfill $\square$

**Lemma B.12** $\lfloor \cdot \rfloor R_{D\text{int}}^p \text{id}[D\text{int}]$.

**Proof.** We must show that for all $P R_{\text{int}} a$, for every $n \in \|\text{int}\|$, $\Pr[\lfloor \cdot \rfloor \cdot \text{ret}_{\text{int}} P \downarrow n] \geq \eta(a)(\{n\})$. By definition of $R_{\text{int}}$, and since $P R_{\text{int}} a$, $\lfloor \cdot \rfloor \cdot P \xrightarrow{1}^* \lfloor \cdot \rfloor \cdot a$. By Lemma B.4 that trace does not use the rule $\lfloor \cdot \rfloor \cdot \text{ret}_{\text{int}} Q \xrightarrow{1} \lfloor \text{ret}_{\text{int}} \cdot \rfloor \cdot Q$. We can therefore use Lemma B.1 and we obtain $\lfloor \text{ret}_{\text{int}} \cdot \rfloor \cdot P \xrightarrow{1}^* \lfloor \text{ret}_{\text{int}} \cdot \rfloor \cdot a$. Together with $\lfloor \cdot \rfloor \cdot \text{ret}_{\text{int}} P \xrightarrow{1} \lfloor \text{ret}_{\text{int}} \cdot \rfloor \cdot P$, we obtain that $\lfloor \cdot \rfloor \cdot \text{ret}_{\text{int}} P \xrightarrow{1}^* \lfloor \text{ret}_{\text{int}} \cdot \rfloor \cdot a$. That is, $\Pr[\lfloor \cdot \rfloor \cdot \text{ret}_{\text{int}} P \downarrow n]$ is equal to 1 if $n = a$, 0 otherwise. This is precisely $\eta(a)(\{n\})$. \hfill $\square$
Theorem B.13 (Adequacy) For every ground term $M : D\text{int}$, for every $n \in \mathbb{Z}$, $\llbracket M \rrbracket (\{n\}) = \Pr[M \downarrow n]$.

Proof. By soundness (Proposition A.2), $\llbracket M \rrbracket (\{n\}) \geq \Pr[M \downarrow n]$. In the converse direction, we use Proposition B.11 with $\theta \overset{\text{def}}{=} \emptyset$ and we obtain $M R_{D\text{int}} \llbracket M \rrbracket$. By Lemma B.12, $\llbracket \_ \_ \rrbracket R_{D\text{int}} \overset{\text{id}_{D\text{int}}}{\Rightarrow}$. Hence, using the definition of $R_{D\text{int}}$, for every $n \in \mathbb{Z}$, $\Pr[M \downarrow n] \geq \text{id}_{D\text{int}}(\llbracket M \rrbracket)(\{n\}) = \llbracket M \rrbracket (\{n\})$. \qed