A Confluent Trace Semantics for Probabilistic Lambda Calculus

Andrew Kenyon-Roberts

***Abstract*—Probabilistic lambda calculus has a weaker version of the confluence property of plain lambda calculus. However, in the usual formulation, this only applies in the sense of distributions, but not to individual traces of random choices. A new labelling scheme for random choices is introduced for PPCF, a simply-typed functional language with explicit recursion, real numbers and a random sampling operator, that allows a confluent version of trace semantics to be defined (with a restricted set of call-by-value reduction strategies).**

I. Introduction

Non-probabilistic lambda calculi are generally confluent, i.e. if a term A reduces to both B_1 and B_2, there is some C to which both B_1 and B_2 reduce, so the reduction order mostly doesn't matter. In the probabilistic case, this may not be true, because β-reduction can duplicate samples, so the outputs of the copies of the sample may be identical or independent, depending on whether the sample is taken before or after β-reduction. Consider for example the term $(\lambda x.x+x)\,\mathsf{sample}$, where sample reduces to a number chosen uniformly at random from the interval $I = [0, 1]$. If it is reduced in call-by-value order, first the sample reduces to some number r, then the β redex is evaluated, then r is added to itself, yielding $\underline{2}\underline{r}$. If it is reduced in call-by-name order instead, first the β redex is reduced, yielding $\mathsf{sample}+\mathsf{sample}$, then the samples are evaluated independently and added, yielding $\underline{r+r'}$. As r and r' are independent, the distribution of results is triangular, with support $[0, 2]$ and peak at 1, which is different from the uniform distribution of results in the CbV case.

The results obtained by CbV and CbN evaluation differ in a significant way, however, there are some cases where the order of evaluation doesn't matter. For example, in $\mathsf{sample}+\mathsf{sample}$, the order in which the samples are evaluated doesn't affect the final result, and in $(\lambda x.\mathsf{sample})\,\underline{0}$, the β redex and the sample can be evaluated in either order. In order to obtain the desired confluence result, we restrict our attention to a class of reduction strategies that are equivalent to CbV, as the CbN semantics is less expressive, being unable to force evaluation of a random choice and duplicate the result.

Even with such a restriction, a trace semantics in the usual style would not be entirely confluent. In the normal sort of trace semantics [1], there is a sequence of samples, the *trace*, selected at random from a trace space such as $I^{\mathbb{N}}$, then for every sample statement reduction, the next sample from the trace is used in order, so that the samples in the trace are effectively each labelled by a number corresponding to the execution order of the sample statements. Consider the evaluation of the term $\mathsf{sample}-\mathsf{sample}$ using one of these simple linear traces, $(1, 0, \dots)$. It would reduce to either 1 or -1 depending on the order of evaluation of the samples, as that determines which sample from the pre-selected sequence is used for each one. To fix this, rather than pre-selecting samples according to the order they'll be drawn in, they can be labelled according to the position in the term where they'll be used instead.

Further details, including all of the missing proofs, can be found in [2, §IV, §D].

A. Outline

First, the syntax of the language PPCF is introduced. Positions are defined as a way of addressing sample statements within a program independently of the reduction order. Next, a version of the reduction relation is presented that is nondeterministic, so that it allows a choice of what order to perform reductions in. The notion of positions is extended to potential positions, for samples which may appear later in the reduction sequence but not necessarily in the initial term. In order to allow potential positions in different reduction sequences to be considered equivalent, a relation $\sim^*$ is defined, and finally, all of these are used to construct a confluent version of the trace semantics, $\Rightarrow$, that is still nondeterministic in reduction order, but does specify the outcome of random choices.

II. Syntax of Probabilistic PCF

The language PPCF is a call-by-value version of PCF with sampling of real numbers from the closed interval $[0, 1]$ [3–5]. Types and terms are defined as follows, where r is a real number, x is a variable, $f : \mathbb{R}^n \to \mathbb{R}$ is any measurable function, and Γ is an environment:

$$\begin{aligned} \textit{types}\ & A, B ::= \mathsf{R} \mid A \to B \\ \textit{values}\ & V ::= \lambda x.M \mid \underline{r} \\ \textit{terms}\ & M, N ::= V \mid x \mid M_1\, M_2 \mid \underline{f}(M_1, \dots, M_n) \mid \mathsf{Y}\, M \\ & \qquad \mid \mathsf{if}(M < 0, N_1, N_2) \mid \mathsf{sample} \end{aligned}$$

The typing rules are standard (see Fig. 1). The restriction to well-typed terms is only necessary here in order to avoid reaching terms which contain nonsense such as applying a number as though it were a function, so a more liberal type system would work just as well. Simple types are just used for simplicity. Terms are identified up to α-equivalence, as usual. The set of all terms is denoted Λ, and the set of closed terms is denoted Λ^0.

$$\frac{}{\Gamma; x : A \vdash x : A} \qquad \frac{\Gamma; x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \to B} \qquad \frac{\Gamma \vdash M : A \to B \quad \Gamma \vdash N : A}{\Gamma \vdash M\, N : B}$$

$$\frac{\Gamma \vdash M : (A \to B) \to (A \to B)}{\Gamma \vdash \mathsf{Y} M : (A \to B)} \qquad \frac{\Gamma \vdash M : \mathsf{R} \quad \Gamma \vdash N_1 : A \quad \Gamma \vdash N_2 : A}{\Gamma \vdash \mathsf{if}(M < 0, N_1, N_2) : A}$$

$$\frac{}{\underline{r} : \mathsf{R}} \qquad \frac{}{\Gamma \vdash \mathsf{sample} : \mathsf{R}} \qquad \frac{\Gamma \vdash M_1 : \mathsf{R} \quad \ldots \quad \Gamma \vdash M_n : \mathsf{R}}{\Gamma \vdash \underline{f}(M_1, \ldots, M_n) : \mathsf{R}}\ (f : \mathbb{R}^n \to \mathbb{R})$$

Figure 1. Typing rules of PPCF

III. Positions

A *position* is a finite sequence of steps into a term, defined inductively as

$$\begin{aligned} \alpha ::= \cdot \mid \lambda; \alpha \mid @_1; \alpha \mid @_2; \alpha \mid \underline{f}_i; \alpha \\ \mid \mathsf{Y}; \alpha \mid \mathsf{if}_1; \alpha \mid \mathsf{if}_2; \alpha \mid \mathsf{if}_3; \alpha. \end{aligned}$$

The *subterm of M at α*, denoted $M \mid \alpha$, is defined as

$$\begin{aligned} M \mid \cdot &= M \\ \lambda x.M \mid \lambda; \alpha &= M \mid \alpha \\ M_1\, M_2 \mid @_i; \alpha &= M_i \mid \alpha \quad \text{for } i = 1, 2 \\ \underline{f}(M_1, \ldots, M_n) \mid \underline{f}_i; \alpha &= M_i \mid \alpha \quad \text{for } i \le n \\ \mathsf{Y} M \mid \mathsf{Y}; \alpha &= M \mid \alpha \\ \mathsf{if}(M_1 < 0, M_2, M_3) \mid \mathsf{if}_i; \alpha &= M_i \mid \alpha \quad \text{for } i = 1, 2, 3 \end{aligned}$$

so that every subterm is located at a unique position, but not every position corresponds to a subterm (e.g. $x\, y \mid \lambda$ is undefined). A position such that $M \mid \alpha$ does exist is said to *occur* in M. *Substitution* of N at position α in M, written $M[N/\alpha]$, is defined similarly. For example, let $M = \lambda x\, y.y\, (\mathsf{if}(x < 0, y\, (\underline{f}(x)), \underline{3}))$ and $\alpha = \lambda; \lambda; @_2; \mathsf{if}_2; @_2$ then $M[\mathsf{sample}/\alpha] = \lambda x\, y.y\, (\mathsf{if}(x < 0, y\, \mathsf{sample}, \underline{3}))$.

Two subterms N_1 and N_2 of a term M, corresponding to positions α_1 and α_2, can overlap in a few different ways. If α_1 is a prefix of α_2 (written as $\alpha_1 \le \alpha_2$), then N_2 is also a subterm of N_1. If neither $\alpha_1 \le \alpha_2$ nor $\alpha_1 \ge \alpha_2$, the positions are said to be *disjoint*. The notion of disjointness is mostly relevant in that if α_1 and α_2 are disjoint, performing a substitution at α_1 will leave the subterm at α_2 unaffected.

Thus we can define a nondeterministic reduction relation $\to$.

Definition III.1. The binary relation $\to$ is defined by the following rules, each is conditional on a redex occurring at position α in the term M:

$$\begin{aligned} \text{if } M \mid \alpha = (\lambda x.N)V,\ M \to M[N[V/x]/\alpha] \\ \text{if } M \mid \alpha = \underline{f}(\underline{r}_1, \ldots, \underline{r}_n),\ M \to M[\underline{f}(r_1, \ldots, r_n)/\alpha] \\ \text{if } M \mid \alpha = \mathsf{Y}\lambda x.N,\ M \to M[\lambda z.N[(\mathsf{Y}\lambda x.N)/x]z/\alpha] \\ \text{where } z \text{ is not free in } N \\ \text{if } M \mid \alpha = \mathsf{if}(\underline{r} < 0, N_1, N_2),\ M \to M[N_1/\alpha] \text{ where } r < 0 \\ \text{if } M \mid \alpha = \mathsf{if}(\underline{r} < 0, N_1, N_2),\ M \to M[N_2/\alpha] \text{ where } r \ge 0 \\ \text{if } M \mid \alpha = \mathsf{sample} \text{ and } \lambda \text{ does not occur after } @_2 \text{ or } \mathsf{Y} \text{ in } \alpha, \\ M \to M[\underline{r}/\alpha] \text{ where } r \in [0, 1]. \end{aligned}$$

In each of these cases, $M \mid \alpha$ is the *redex*, and the reduction *takes place at* α. Each subterm can be a redex in at most one way, but there can be multiple redexes at different positions.

The argument of a β redex and the body of a Y redex may be duplicated by those reductions. It is therefore these cases that need to be handled carefully to avoid duplicating samples at the wrong time. In both cases, the potentially duplicated part must already be a value, which excludes terms like sample or $\mathsf{sample}{+}\underline{1}$, which should be evaluated before being duplicated. In the other direction, if a sample occurs inside of a λ, it may need to be duplicated before being evaluated, which is why a sample reduction isn't allowed inside a λ inside a Y or the right side of an application. These restrictions are in some cases unnecessarily strict, for example, in $(\lambda x.x)((\lambda y.\mathsf{sample})\underline{0})$, it would be fine to evaluate the sample first, but they are at least sufficient to ensure confluence in terms of the distribution of results. Getting individual traces to behave correctly will take more work though.

IV. Skeletal Reduction Sequences

Labelling the pre-chosen samples by the positions in the term by using $I^{\{\alpha \mid (M \mid \alpha) = \mathsf{sample}\}}$ as the trace space would not be sufficient to solve the issue of different samples being used in corresponding locations in different reduction sequences because in some cases, a sample will be duplicated before being reduced, for example, in $(\lambda x.x\,\underline{0} + x\,\underline{0})(\lambda y.\mathsf{sample})$, both of the sample redexes that eventually occur originate at $@_2; \lambda$. It is therefore necessary to consider possible positions that may occur in other terms reachable from the original term. Even this is itself inadequate because some of the positions

in different reachable terms need to be considered the same, and the number of reachable terms is in general uncountable, which leads to measure-theoretic issues.

We are thus led to consider the reduction relation on skeletons. Define a *skeleton* to be a term but, instead of having real constants $\underline{r}$, it has a placeholder X, so that each term M has a skeleton $Sk(M)$, and each skeleton S can be converted to a term $S[r]$ given a vector r of n real numbers to substitute in, where n is the number of occurrences of X in S. Positions in a skeleton and the reduction relation $\to$ on skeletons can be extended from the definitions on terms in the obvious way, with $\mathsf{if}(\mathsf{X} < 0, A, B)$ reducing nondeterministically to both A and B, sample reducing to X, and X considered as (the skeletal equivalent of) a value, so that $(\lambda x.A)\,\mathsf{X}$ reduces to $A[\mathsf{X}/x]$. For example, we have $(\lambda x.\mathsf{if}(x < 0, x, \mathsf{X}))\,\mathsf{sample} \to (\lambda x.\mathsf{if}(x < 0, x, \mathsf{X}))\,\mathsf{X} \to \mathsf{if}(\mathsf{X} < 0, \mathsf{X}, \mathsf{X}) \to \mathsf{X}$.

Given a closed term M, let $L_0(M)$ be the set of pairs, the first element of which is a $\to$-reduction sequence of skeletons starting at $Sk(M)$, and the second of which is a position in the final skeleton of the reduction sequence. As with the traces from $I^{\mathbb{N}}$ used to pre-select samples to use in the standard trace semantics, modified traces, which are elements of $I^{L_0(M)}$ (with one more caveat introduced after Def. V.2), will be used to pre-select a sample from I for each element of $L_0(M)$, which will then be used if a sample reduction is ever performed at that position.

A (skeletal) reduction sequence is assumed to contain the information on the locations of all of the redexes as well as the actual sequence of skeletons that occurs. For example, $(\lambda x.x)((\lambda x.x)\,\mathsf{X})$ could reduce to $(\lambda x.x)\,\mathsf{X}$ with the redex at either $\cdot$ or $@_2$, and these give different reduction sequences.

Example IV.1. Consider the terms

$$A[M] = \mathsf{if}(\mathsf{if}(M > 0, I, I)(\lambda y.\mathsf{sample})\,\underline{0} - \underline{0.5} > 0, \underline{0}, \Omega)$$
$$B = \mathsf{if}(\mathsf{sample} - \underline{0.5} > 0, \underline{0}, \Omega)$$

If terms rather than skeletons were used to label samples, the set of modified traces where $A[\mathsf{sample}]$ terminates would be

$$\bigcup_{r\in[0,1]} \{s \mid s(A[\mathsf{sample}], \mathsf{if}_1; \underline{-}_1; @_1; @_1; \mathsf{if}_1) = r, \\ s(A[\mathsf{sample}] \to A[\underline{r}] \to^* B, \mathsf{if}_1; \underline{-}_1) > 0.5\}.$$

This is a rather unwieldy expression, but the crucial part is that r occurs twice in the conditions on s: once as the value a sample must take, and once in the location of a sample. As this set is unmeasurable, the termination probability would not even be well-defined. Labelling samples by skeletons instead, this problem does not occur because there are only countably many skeleton, and at each step in a reduction sequence, only finitely many could have occurred yet. Although skeletal reduction sequences omit the information on what the results of sampling were, they still contain all the necessary information on how many, and which, reductions took place.

For this particular term, $Sk(A[\underline{r}])$ does not depend on the value of r, therefore the set where it terminates becomes simply the following, which is measurable.

$$\{s \mid s(Sk(A[\mathsf{sample}]) \to Sk(A[\underline{0}]) \to^* Sk(B), \mathsf{if}_1; \underline{-}_1) > 0.5\}$$

Reduction sequences are used rather than reachable skeletons because if the same skeleton is reached twice, different samples may be needed:

Example IV.2. Consider the term $M = \mathsf{Y}(\lambda f x.\,\mathsf{if}(\mathsf{sample} - \underline{0.5} < 0, f\,x, x))\,\underline{0}$, which reduces after a few steps to $N = \mathsf{if}(\mathsf{sample} - \underline{0.5} < 0, M, \underline{0})$. If we label samples by just skeletons and positions, and the pre-selected sample for $(Sk(N), \mathsf{if}_1; \underline{-}_1)$ is less than 0.5, N reduces back to M, then N again, then the same sample is used the next time, therefore it's an infinite loop, whereas if samples are labelled by reduction sequences, the samples for $M \to^* N$ are independent from the samples for $M \to^* N \to M \to^* N$, and so on.

The reduction sequences of skeletons will often be discussed as though they were just skeletons, identifying them with their final skeletons. With this abuse of notation, a reduction sequence N (actually $N_1 \to^* N_n = N$) may be said to reduce to a reduction sequence O, where the reduction sequence implicitly associated with the final skeleton O is $N_1 \to^* N_n \to O$.

V. Potential Positions

This is still not quite sufficient to attain confluence because sometimes the same samples must be used at corresponding positions in different reduction sequences.

Example V.1. The term $M = \mathsf{sample} + \mathsf{sample}$ has the reachable skeletons $N_1 = \mathsf{X} + \mathsf{sample}$, $N_2 = \mathsf{sample} + \mathsf{X}$, $O = \mathsf{X} + \mathsf{X}$ and X, with reductions $M \to N_1 \to O \to \mathsf{X}$ and $M \to N_2 \to O \to \mathsf{X}$. In the reduction $M \to N_1$, the sample labelled $(M, \underline{+}_1)$ is used, and in the reduction $N_2 \to O$, the sample labelled $(M \to N_2, \underline{+}_1)$ is used. Each of these samples becomes the value of the first numeral in O in their respective reduction sequences, therefore in order for confluence to be attained, they must be the same. Which elements of $L_0(M)$ must match can be described by the relation $\sim^*$:

Definition V.2. The relation $\sim$ is defined as the union of the minimal symmetric relations $\sim_p$ ("p" for parent-child) and $\sim_c$ ("c" for cousin) satisfying

(i) If N reduces to O with the redex at position α, and β is a position in N disjoint from α, then $(N, \beta) \sim_p (O, \beta)$.

(ii) If N β-reduces to O at position α, β is a position in $N \mid \alpha; @_1; \lambda$ and $N \mid \alpha; @_1; \lambda; \beta$ is not the variable involved in the reduction, $(N, \alpha; @_1; \lambda; \beta) \sim_p (O, \alpha; \beta)$.

(iii) If N if-reduces to O at position α, with the first resp. second branch being taken, and $\alpha; \mathsf{if}_i; \beta$ occurs in N (where $i = 2$ resp. 3), $(N, \alpha; \mathsf{if}_i; \beta) \sim_p (O, \alpha; \beta)$.

(iv) If N, O_1 and O_2 match any of the following cases:

a) N contains redexes at disjoint positions α_1 and α_2, O_1 is N reduced first at α_1 then α_2 and O_2 is N reduced first at α_2 then at α_1.

b) $N \mid \alpha = \mathsf{if}(\underline{r} < 0, N_1, N_2)$, where $r < 0$ (or, respectively, $r \geq 0$), (N_2 resp. N_1) $\mid \beta$ is a redex, and O_1 is N reduced at α and O_2 is N reduced first at $\alpha; (\mathsf{if}_3$ resp. $\mathsf{if}_2); \beta$ then at α.

c) $N \mid \alpha = \mathsf{if}(\underline{r} < 0, N_1, N_2)$, where $r < 0$ (or, respectively, $r \geq 0$), (N_1 resp. N_2) $\mid \beta$ is a redex, and O_1 is N reduced first at α then at $\alpha; \beta$ and O_2 is N reduced first at $\alpha; (\mathsf{if}_2$ resp. $\mathsf{if}_3); \beta$ then at α.

d) $N \mid \alpha = (\lambda x.A)B$, there is a redex in A at position β, O_1 is N reduced first at α then at $\alpha; \beta$, and O_2 is N reduced first at $\alpha; @_1; \lambda; \beta$ then at α.

e) $N \mid \alpha = (\lambda x.A)B$, $B \mid \beta$ is a redex, $(\gamma_i)_i$ is a list of all the positions in A where $A \mid \gamma = x$, ordered from left to right, O_1 is N reduced first at $\alpha; @_2; \beta$ then at α, and O_2 is N reduced first at α then at $\alpha; \gamma_i; \beta$ for each i in order.

f) $N \mid \alpha = \mathsf{Y}(\lambda x.A)$, A reduced at β is A', $(\gamma_i)_i$ is a list of all the positions where $A' \mid \gamma = x$, ordered from left to right, O_1 is N reduced first at $\alpha; \mathsf{Y}; \lambda; \beta$ then at α, and O_2 is N reduced first at α then at $\alpha; \lambda; @_1; \gamma_i; Y; \lambda; \beta$ for each i in order where γ_i is left of β then at $\alpha; \lambda; @_1; \beta$ then at $\alpha; \lambda; @_1; \gamma_i; Y; \lambda; \beta$ for the remaining values of i.

(in which case O_1 and O_2 are equal as skeletons, but with different reduction sequences), O_1' and O_2' are the results of applying some reduction sequence to each of O_1 and O_2 (the same reductions in each case, which is always possible because they're equal skeletons), and δ is a position in O_1' (or equivalently O_2'), then $(O_1', \delta) \sim_c (O_2', \delta)$.

Example V.3. In Ex. V.1, $(M, \underline{+}_1) \sim_p (M \to N_2, \underline{+}_1)$ by case i of $\sim_p$ (because the reduction $M \to N_2$ occurs at $\underline{+}_2$, which is disjoint from $\underline{+}_1$), and similarly, $(M, \underline{+}_2) \sim_p (M \to N_1, \underline{+}_2)$.

If we extend it to have three samples, $\sim_c$ becomes necessary as well: Let $M_{sss} = \mathsf{sample} + \mathsf{sample} + \mathsf{sample}$ (taking the three-way addition to be a single primitive function), $M_{Xss} = \mathsf{X} + \mathsf{sample} + \mathsf{sample}$, and so on. There are then reduction sequences $M_{sss} \to M_{Xss} \to M_{XXs} \to M_{XXX} \to \mathsf{X}$ and $M_{sss} \to M_{sXs} \to M_{XXs} \to M_{XXX} \to \mathsf{X}$. For the first two reductions, these reduction sequences take the same samples by $\sim_p$, case i, as in Ex. V.1 . The next reduction uses the samples labelled by $(M_{sss} \to M_{Xss} \to M_{XXs}, \underline{+}_3)$ and $(M_{sss} \to M_{sXs} \to M_{XXs}, \underline{+}_3)$, which are related by $\sim_c$, case a, therefore when these reduction sequences reach M_{XXX}, they still contain all the same numbers, as desired.

The reflexive transitive closure $\sim^*$ of this relation is used to define the set of *potential positions* $L(M) = L_0(M)/\sim^*$, and each equivalence class can be considered as the same position as it may occur across multiple reachable skeletons. If $(N, \alpha) \sim^* (O, \beta)$, then $N \mid \alpha$ and $O \mid \beta$ both have the same shape (i.e. they're either both the placeholder X, both variables, both applications, both samples etc.), therefore it's well-defined to talk of the set of *potential positions where there is a* sample, $L_s(M)$. The new sample space is then defined as $I^{L_s(M)}$, with the Borel σ-algebra and product measure. Since $I^{L_s(M)}$ is a countable product, the measure space is well-defined [6, Cor. 2.7.3].

VI. The Confluent Trace Semantics

Before defining the new version of the reduction relation, the following lemma is necessary for it to be well-defined.

Lemma VI.1. *The relation $\sim$ is defined on $L_0(M)$ with reference to a particular starting term M, so different versions, $\sim_M$ and $\sim_N$, can be defined starting at different terms. If $M \to N$, then $\sim_N^*$ is equal to the restriction of $\sim_M^*$ to $L_0(N)$.*

At each reduction step $M \to N$, the sample space must be restricted from $I^{L_s(M)}$ to $I^{L_s(N)}$. The injection $L_0(N) \to L_0(M)$ is trivial to define by appending $Sk(M) \to Sk(N)$ to each path, and using Lem. VI.1, this induces a corresponding injection on the quotient, $L(N) \to L(M)$. The corresponding map $L_s(N) \to L_s(M)$ is then denoted $i(M \to N)$.

Definition VI.2 ($\Rightarrow$ reduction)**.** This version of the reduction relation now specifies the results of sample reductions, but is still nondeterministic with respect to the order of reduction. It relates $\biguplus_{M \in \Lambda_0} I^{L_s(M)}$ to itself. We write an element of $\biguplus_{M \in \Lambda_0} I^{L_s(M)}$ as (M', s) where the term $M' \in \Lambda^0$ and $s \in I^{L_s(M')}$.

$(M, s) \Rightarrow (N, s \circ i(M \to N))$ if $M \to N$ at α and either
the redex is not sample, or
$M \mid \alpha = \mathsf{sample}$ and $N = M[\underline{s(Sk(M), \alpha)}/\alpha]$

This reduction relation now has all of the properties required of it. In particular, it can be considered an extension of the standard trace semantics (as will be seen later in Thm. VI.5), and also:

Lemma VI.3. *The relation $\Rightarrow$ is confluent.*

In order to show that $\Rightarrow$ behaves as expected, the following lemma is also necessary, in order to show that a sample is never used multiple times in the same reduction sequence:

Lemma VI.4. *If $M \to N$, with the redex at position α, then no position in any term reachable from N is related by $\sim^*$ to (M, α).*

The reduction relation $\Rightarrow$ is nondeterministic, so it admits multiple possible reduction strategies. A *reduction strategy* starting from a closed term M is a measurable partial function f from $Rch(M)$ to positions, such that for any reachable term N where f is defined, $f(N)$ is a position of a redex in N, and if $f(N)$ is not defined, N is a value. Using a reduction strategy f, a subset of $\Rightarrow$ that isn't nondeterministic, $\Rightarrow_f$, can be defined by $(N, s) \Rightarrow_f (N', s')$ just if $(N, s) \Rightarrow (N', s')$ and N reduces to N' with the redex at $f(N)$.

The usual call-by-value semantics can be implemented as one of these reduction strategies, given by (with V a value

and T a term that isn't a value and M a general term)

$$\begin{aligned}
\mathrm{cbv}(TM) &= @_1; \mathrm{cbv}(T) \\
\mathrm{cbv}(VT) &= @_2; \mathrm{cbv}(T) \\
\mathrm{cbv}(\underline{f}(V_1, \ldots, V_{k-1}, T, M_{k+1}, \ldots, M_n)) &= \underline{f}_k; \mathrm{cbv}(T) \\
\mathrm{cbv}(\mathsf{Y}T) &= \mathsf{Y}; \mathrm{cbv}(T) \\
\mathrm{cbv}(\mathsf{if}(T < 0, M_1, M_2)) &= \mathsf{if}_1; \mathrm{cbv}(T) \\
\mathrm{cbv}(V) &\text{ is undefined} \\
\mathrm{cbv}(T) &= \cdot \text{ otherwise}
\end{aligned}$$

(this last case covers redexes at the root position).

A closed term M terminates with a given reduction strategy f and samples s if there is some natural number n such that $(M, s) \Rightarrow_f^n (N, s')$ where f gives no reduction at N. The term is *almost surely terminating (AST) w.r.t. f* if it terminates with f for almost all s.

This reduction strategy allows the confluent trace semantics to be related to the standard version of the trace semantics with a fixed reduction order and linear traces. In [2], which gives the full definition of the standard trace semantics, this is used to prove the following theorems that allow termination results to be transferred from the confluent trace semantics to the standard trace semantics.

Theorem VI.5. *A closed term M is AST with respect to* cbv *iff it is AST.*

Theorem VI.6. *If M terminates with some reduction strategy f and trace s, it terminates with* cbv *and s.*

Corollary VI.7 (Reduction strategy independence)**.** *If M is AST with respect to any reduction strategy, it is AST.*

Proof. Suppose M is AST w.r.t. f. Let the set of samples with which it terminates with this reduction strategy be X. By Thm. VI.6, M also terminates with cbv and every element of X, and X has measure 1, by assumption, therefore M is AST with respect to cbv therefore by Thm. VI.5 it is AST. □

References

[1] J. Borgström, U. Dal Lago, A. D. Gordon, and M. Szymczak, "A lambda-calculus foundation for universal probabilistic programming," *ACM SIGPLAN Notices*, vol. 51, no. 9, pp. 33–46, 2016.

[2] A. Kenyon-Roberts and L. Ong, "Supermartingales, ranking functions and probabilistic lambda calculus," *arXiv preprint arXiv:2102.11164*, 2021.

[3] T. Ehrhard, M. Pagani, and C. Tasson, "Full Abstraction for Probabilistic PCF," *Journal of the ACM*, vol. 65, no. 4, pp. 1–44, apr 2018. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3208081.3164540

[4] ——, "Measurable cones and stable, measurable functions: a model for probabilistic higher-order programming," *PACMPL*, vol. 2, no. POPL, pp. 59:1–59:28, 2018. [Online]. Available: http://doi.acm.org/10.1145/3158147

[5] C. Mak, C.-H. L. Ong, H. Paquet, and D. Wagner, "Densities of almost surely terminating probabilistic programs are differentiable almost everywhere," in *ESOP 2021*, 2021, to appear. https://arxiv.org/abs/2004.03924.

[6] R. B. Ash and C. Doléans-Dade, *Probability and measure Theory*. Harcourt Academic Press, 2000.