HAL
open science

On Randomised Strategies in the

λ

-Calculus

Ugo Dal Lago, Gabriele Vanoni

▶ **To cite this version:**

Ugo Dal Lago, Gabriele Vanoni. On Randomised Strategies in the

λ

-Calculus. 19th Italian Conference on Theoretical Computer Science, Sep 2018, Urbino, Italy. hal-01926512

HAL Id: hal-01926512

https://hal.science/hal-01926512v1

Submitted on 19 Nov 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

On Randomised Strategies in the λ-Calculus*

Ugo Dal Lago[1] and Gabriele Vanoni[2]

[1] Università di Bologna & INRIA Sophia Antipolis
ugo.dallago@unibo.it
[2] Politecnico di Milano
gabriele.vanoni@mail.polimi.it

Abstract. In this work we study randomised reduction strategies— a notion already known in the context of abstract reduction systems—for the λ-calculus. We develop a simple framework that allows us to prove a randomised strategy to be positive almost-surely normalising. Then we propose a simple example of randomised strategy for the λ-calculus that has such a property and we show why it is non-trivial with respect to classical deterministic strategies such as leftmost-outermost or rightmost-innermost. We conclude studying this strategy for the affine λ-calculus, where duplication is syntactically forbidden.

Keywords: λ-calculus · probabilistic rewriting · reduction strategies.

1 Introduction

There are different possible *strategies* you can follow to evaluate expressions. Some are better than others, and bring you to the result in a lower number of steps. Since programs in pure functional languages are essentially expressions, the problem of defining good strategies is particularly interesting. Finding *minimal* strategies, i.e. strategies that minimise the number of steps to normal form, seems even more interesting. However, the problem of picking the redex leading to the reduction sequence of minimal length has been proven undecidable for the λ-calculus [3, Section 13.5], *the* paradigmatic pure functional language. In the last decades several reduction strategies have been developed. Their importance is crucial in the study of evaluation orders in functional programming languages, which is one of their main characteristics and defines an important part of their semantics. The reader can think about the differences between `Haskell` (*call-by-need*) and `Caml` (*call-by-value*). The λ-calculus is a good abstraction to study reduction mechanisms because of its very simple structure. In fact, although *Turing-complete*, it can be seen as a rewriting system [13], where terms can be formed in only two ways, by abstraction and application, and only one rewriting rule, the β-rule, is present. Reduction strategies for the λ-calculus are typically defined according to the position of the contracted *redex* e.g. *leftmost-outermost*, *leftmost-innermost*, *rightmost-innermost*. As the following trivial examples show,

* This work was partially supported by ANR grant Elica ANR-14-CE25-0005 and Inria/JSPS EA Crecogi. An Extended Version with more details is available [9].

CEUR-WS.org/Vol-2243/paper9.pdf

the adopted strategy can indeed have a strong impact on evaluation performances, and possibly also on termination behaviour.

Example 1. Let $\boldsymbol{\omega} = \lambda x.xx$ and $\boldsymbol{\Omega} = \boldsymbol{\omega}\boldsymbol{\omega}$. We now consider the reduction of the term $(\lambda x.y)\boldsymbol{\Omega}$ according to two different reduction strategies, namely leftmost-outermost (LO) and rightmost-innermost (RI).

$$(\lambda x.y)\boldsymbol{\Omega} \longrightarrow_{\mathsf{LO}} y$$
$$(\lambda x.y)\boldsymbol{\Omega} \longrightarrow_{\mathsf{RI}} (\lambda x.y)\boldsymbol{\Omega} \longrightarrow_{\mathsf{RI}} (\lambda x.y)\boldsymbol{\Omega} \longrightarrow_{\mathsf{RI}} \cdots$$

The term $\boldsymbol{\Omega}$ is a looping combinator i.e. it reduces to itself. However, in $(\lambda x.y)\boldsymbol{\Omega}$ the argument $\boldsymbol{\Omega}$ is discarded since the function returns the constant y. Thus leftmost-outermost (akin to call-by-name in functional programming languages) yields a normal form in one step. Conversely, rightmost-innermost (akin to call-by-value) continues to evaluate the argument $(\lambda x.y)\boldsymbol{\Omega}$, though it is useless, and rewrites always the same term, yielding to a non-terminating process.

Example 2. Let $\boldsymbol{I} = \lambda x.x$. We now consider the reduction of the term $(\lambda x.xx)(\boldsymbol{I}\boldsymbol{I})$, according to LO and RI strategies, as above.

$$(\lambda x.xx)(\boldsymbol{I}\boldsymbol{I}) \longrightarrow_{\mathsf{LO}} (\boldsymbol{I}\boldsymbol{I})(\boldsymbol{I}\boldsymbol{I}) \longrightarrow_{\mathsf{LO}} \boldsymbol{I}(\boldsymbol{I}\boldsymbol{I}) \longrightarrow_{\mathsf{LO}} \boldsymbol{I}\boldsymbol{I} \longrightarrow_{\mathsf{LO}} \boldsymbol{I}$$
$$(\lambda x.xx)(\boldsymbol{I}\boldsymbol{I}) \longrightarrow_{\mathsf{RI}} (\lambda x.xx)\boldsymbol{I} \longrightarrow_{\mathsf{RI}} \boldsymbol{I}\boldsymbol{I} \longrightarrow_{\mathsf{RI}} \boldsymbol{I}$$

Here the argument $\boldsymbol{I}\boldsymbol{I}$ is duplicated and thus it is much more convenient to reduce it before it is copied, as in rightmost-innermost. Leftmost-outermost does, indeed, some useless work.

In general, innermost strategies are considered more efficient, because programs often need to copy their arguments (as in Example 2). However, as seen in Example 1, rightmost-innermost is not normalising: there exist terms which have a normal form which, however, can be missed by innermost strategies. Instead, a classical result by Curry and Feys [8] states that the leftmost-outermost strategy is *normalising*, i.e. it always rewrites terms to their normal norm, if it exists. Thus, leftmost-outermost is slower, but safer. Could we get, in a sense, the best of both worlds? All reduction strategies for the λ-calculus in the literature up to now are *deterministic*, i.e. they are (partial) functions on (possibly shared representations of) terms. There is however some work on probabilistic term rewriting systems [4, 10, 2], in particular regarding termination, and about randomised strategies in the abstract [5]. What would happen if the redexes to reduce were picked according to some probability distribution? How many steps would a term need to reach a normal form *on the average*?

In this work we consider a simple *randomised* reduction strategy P_ε, where the LO-redex is reduced with probability ε and the RI-redex is reduced with probability $1 - \varepsilon$. This is not necessarily the most interesting example, but certainly a good starting point in our investigation. The *uniform* randomised strategy which picks one between *all* the redexes in the term uniformly at random looks more natural, although much more difficult to analyse: there is no fixed lower bound on the probability of picking *the standard* redex, i.e. the leftmost-outermost one. The following are our main results:

- For every, $0 < \varepsilon \leq 1$, the strategy P_ε is positive almost-surely normalising on weakly normalising terms. That means that if a term M is weakly normalising, then the expected number of reduction steps from M to its normal form with strategy P_ε is finite. This is in contrast to the rightmost-innermost strategy, as can be seen from Example 1. Rightmost-innermost, in other words, is the only non normalising strategy in the family $\{\mathsf{P}_\varepsilon\}_{0 \leq \varepsilon \leq 1}$, namely P_0.
- The family of strategies $\{\mathsf{P}_\varepsilon\}_{0 < \varepsilon < 1}$ is shown to be non-trivial. In other words, there exists a class of terms and $0 < \mu < 1$ for which P_μ outperforms, on average, both LO and RI. This shows that randomisation can indeed be useful in this context. This is not surprising: in computer science there are a lot of examples where adding a random factor improves performances, e.g. in randomised algorithms, which are often faster (in average) than their deterministic counterparts [11].
- The expected number of reduction steps to normal form with strategy P_ε, seen as a function on ε, has minimum in 1 for terms in the affine λ-calculus λA.

The rest of this paper is structured as follows. In Section 2 basic definitions and results for the untyped λ-calculus are given. In Section 3 we present our model of fully probabilistic abstract reduction system and we give a sufficient condition to prove positive almost-sure termination. In Section 4 we apply this model to the λ-calculus, defining a randomised reduction strategy and collecting some results. Section 5 contains the conclusions with some ideas for further investigations on the subject.

Acknowledgements

Our interest in randomised strategies comes from some interesting and insightful discussions the first author had with Prakash Panangaden.

2 Basic Notions and Notations

The following definitions are standard and are adapted from [13].

Definition 1. *Assume a countable infinite set $\mathcal{V}$ of variables. The* λ-calculus *is the language of terms defined by the following grammar:*

$$M, N ::= x \in \mathcal{V} \;\Big|\; MN \;\Big|\; \lambda x.M$$

We denote by Λ the set of all λ-terms. As usual, λ-terms are taken modulo α-equivalence, which allows to appropriately define the capture-avoiding substitution of all the free occurrences of x for N in M, denoted $M\{N/x\}$.

Reduction will be defined based on the notion of a context, which needs to be given a formal status.

Definition 2. *We define (one-hole)* contexts *by the following grammar:*

$$C, D ::= \Box \;\Big|\; CM \;\Big|\; MC \;\Big|\; \lambda x.C$$

We denote with $\Lambda_\Box$ the set of all contexts.

Intuitively, contexts are λ-terms with a hole that can be filled with another λ-term. We indicate with $C[M]$ the term obtained by replacing $\Box$ with M in C. Those λ-terms in the form $R = (\lambda x.M)N$ are called *β-reducible expressions* or *β-redexes* and $M\{N/x\}$ is said to be the *contractum* of R. This is justified by the following definition.

Definition 3. *The relation of* β-reduction*, $\longrightarrow_\beta \subseteq \Lambda \times \Lambda$, is defined as*

$$\longrightarrow_\beta = \{(C[(\lambda x.M)N],\, C[M\{N/x\}]) \mid M, N \in \Lambda, C \in \Lambda_\Box\}.$$

The relation of ANF-β-reduction*, $\underset{\beta\mathsf{ANF}}{\longrightarrow} \subseteq \Lambda \times \Lambda$, is defined by*

$$\underset{\beta\mathsf{ANF}}{\longrightarrow} = \{(C[(\lambda x.M)N],\, C[M\{N/x\}]) \mid M, N \in \Lambda,\ N \text{ is in normal form},\ C \in \Lambda_\Box\}.$$

We denote by $\twoheadrightarrow_\beta$ the reflexive and transitive closure of $\longrightarrow_\beta$.

We can see the λ-calculus defined above as an *abstract reduction system* (ARS) [13] $(\Lambda, \longrightarrow_\beta)$, namely a countable set endowed with a relation, also called the reduction relation. We denote by Λ_{WN} the set of weakly normalising terms of Λ, i.e. terms which have *a* normal form.

2.1 Two Subcalculi of Λ

There are interesting subsets of the λ-calculus. In particular we focus our attention on two subsystems where terms satisfy a predicate on the number of occurrences of free variables. These systems are meaningful because they are *stable* w.r.t. β-reduction i.e. if $M \in S$ and $M \longrightarrow_\beta N$ then $N \in S$. A comprehensive treatment is in [12].

The λI-calculus. The λI-calculus was the original calculus studied by Alonzo Church in the '30 [7], and [3] contains a whole section dedicated to it. In λI-calculus there is no *cancellation*, in that variables have to occur free *at least once* when forming abstractions. Terms of the λI-calculus are not strongly normalising in general. As an example, $\boldsymbol{\Omega}$ is a λI-term. One can prove, however, that on λI-terms, weak normalisation implies strong normalisation: in other words, all strategies are qualitatively equivalent. This does *not* mean, however, that they are quantitatively equivalent.

The λA-calculus. The λA-calculus is the dual of λI-calculus and it is sometimes called *affine* λ-calculus in the literature. It is a very weak calculus in which variables bound by abstractions occur *at most once* free in the abstraction's body, thus forbidding *duplication*. The λA-calculus is strongly normalising, in a very strong sense: every reduction sequence from a term M has length bounded by the size of M.

2.2 Reduction Strategies

ARSs are sets endowed with a relation, and are thus inherently nondeterministic. The notion of reduction strategy allows us to fix one redex among the available ones, thus turning reduction into a deterministic process.

Definition 4 (Deterministic Strategies). *Given an ARS* $(A, \to)$*, a* deterministic reduction strategy *for* A *is a partial function* $\mathsf{S} : A \rightharpoonup A$ *such that* $\mathsf{S}(a)$ *is defined iff* a *is not in normal form and* $a \to \mathsf{S}(a)$ *whenever* $\mathsf{S}(a)$ *is defined.*

If $\sigma : a_0 \to a_1 \to \cdots \to a_n$ is a reduction sequence with strategy S and a_n is in normal form, we write $\mathsf{Steps}_{\mathsf{S}}(a_0) = n = |\sigma|$. If $\sigma : a_0 \to a_1 \to \cdots$ is infinite, we say that $\mathsf{Steps}_{\mathsf{S}}(a_0) = +\infty$. We define two reduction strategies for the λ-calculus that will be useful in the following sections.

Definition 5. Leftmost-outermost (LO) *is a deterministic reduction strategy in which* $\mathsf{LO}(M) = N$ *if and only if* $M \longrightarrow_\beta N$ *and the redex contracted in* M *is the* leftmost *among the ones in* M *(measuring the position of a redex by its beginning). If* $\mathsf{LO}(M) = N$*, we write* $M \longrightarrow_{\mathsf{LO}} N$*.*

Definition 6. Rightmost-innermost (RI) *is a deterministic reduction strategy in which* $\mathsf{RI}(M) = N$ *if and only if* $M \longrightarrow_\beta N$ *and the redex contracted in* M *is the* rightmost *among the ones in* M *(measuring the position of a redex by its beginning). Again, if* $\mathsf{RI}(M) = N$*, we write* $N \longrightarrow_{\mathsf{RI}} N$*.*

Lemma 1. *If* $M \longrightarrow_\beta N$*, then* $\mathsf{Steps}_{\mathsf{LO}}(N) \leq \mathsf{Steps}_{\mathsf{LO}}(M)$*.*

3 Probabilistic Abstract Reduction Systems as Strategies

We introduce now a framework suitable to define randomised strategies. In particular, we shift the notion of ARS to the fully probabilistic case. With respect to standard Markov chain theory, our construction is simpler and allows to reason better in an infinite state context. The first preliminary concept we need is that of a distribution.

Definition 7 (Distribution). *A* partial probability distribution *over a countable set* A *is a mapping* $\rho : A \to [0, 1]$ *such that* $|\rho| \leq 1$ *where* $|\rho| = \sum_{a \in A} \rho(a)$*. We denote the set of partial probability distributions over* A *by* $\mathsf{PDist}(A)$*. The* support *of a partial distribution* $\rho \in \mathsf{PDist}(A)$ *is the set* $\mathbf{Supp}(\rho) = \{a \in A \mid \rho(a) > 0\}$*. A* probability distribution *over a countable set* A *is a partial probability distribution* μ *such that* $|\mu| = 1$*.* $\mathsf{Dist}(A)$ *denotes the set of probability distributions over* A*.*

Strategies as from Definition 4 are inherently deterministic: the process of picking a reduct among the many possible ones can only have *one* outcome. But what if this process becomes *probabilistic*? This is captured by the following notion:

Definition 8 (Randomised Strategies). *Given an ARS* $(S, \to)$*, a* randomised reduction strategy P for $(S, \to)$ *is a partial function such that if* $s \in S$ *is in normal form, then* $\mathsf{P}(s) = \bot$*, otherwise* $\mathsf{P}(s) = \mu \in \mathsf{Dist}(S)$*, and* $\mathbf{Supp}(\mu) \subseteq \{t \mid s \to t\}$*.*

Please notice that if $(S, \to)$ is an ARS and P is a randomised reduction strategy for it, then (S, P) can be seen as a *fully* probabilistic abstract reduction system (FPARS), namely a probabilistic abstract reduction system [2] whose dynamics is purely probabilistic, without any nondeterminism. In the following, we will study randomised strategies as FPARS.

The dynamics of an FPARS can be handled by way of an appropriate notion of a configuration, on which an evolution function can be defined:

Definition 9 (Configurations, Computations). *Let (S, P) be an FPARS and $s, t \in S$ be two* states*. We define the probability $\mathbb{P}(s \to t)$ of a* transition *from s to t:*

$$\mathbb{P}(s \to t) = \begin{cases} \mu(t) & \text{if } \mathsf{P}(s) = \mu, \\ 0 & \text{if } \mathsf{P}(s) = \bot. \end{cases}$$

A configuration *of an FPARS (S, P) is a partial probability distribution $\rho \in \mathsf{PDist}(S)$. The* evolution *of an FPARS (S, P) from a configuration ρ is a function $\mathsf{E} : \mathsf{PDist}(S) \to \mathsf{PDist}(S)$ defined in the following way:*

$$\mathsf{E}(\rho) = \sigma \text{ where } \sigma(s) = \sum_{t \in S} \rho(t) \cdot \mathbb{P}(t \to s) \text{ for every } s \in S.$$

If $\mathsf{E}(\rho) = \sigma$ we write $\rho \rightsquigarrow \sigma$. A computation *is any sequence $(\rho_i)_{i \in \mathbb{N}}$, such that $\rho_i \rightsquigarrow \rho_{i+1}$.*

Remark 1. Those computations $(\rho_i)_{i \in \mathbb{N}}$ where ρ_0 is **Dirac** (i.e. there exists $s \in S$ such that $\rho_0(s) = 1$) are particularly interesting: they model the evolution of an FPARS starting from one state. We write in this case $\rho_0 = \mathbf{Dirac}(s)$.

Example 3. Consider an ARS $(S, \to)$, where $S = \{a, b\}$ and $\to = \{(a, a), (a, b)\}$. We define a randomised strategy P on top of $(S, \to)$. $\mathsf{P}(a) = \mu$ such that $\mu(a) = \mu(b) = \frac{1}{2}$, while $\mathsf{P}(b) = \bot$ since b is in normal form. A computation $(\rho_i)_{i \in \mathbb{N}}$ starting from $\rho_0 = \mathbf{Dirac}(a)$ has the following form.

$$\underset{\rho_0}{\begin{cases} 1 & a \\ 0 & b \end{cases}} \rightsquigarrow \underset{\rho_1}{\begin{cases} \frac{1}{2} & a \\ \frac{1}{2} & b \end{cases}} \rightsquigarrow \underset{\rho_2}{\begin{cases} \frac{1}{4} & a \\ \frac{1}{4} & b \end{cases}} \rightsquigarrow \cdots \rightsquigarrow \underset{\rho_k}{\begin{cases} \frac{1}{2^k} & a \\ \frac{1}{2^k} & b \end{cases}} \rightsquigarrow \cdots$$

How could we measure the *length* of a computation? It is natural to look for a definition capturing the *average* derivation length from s to its normal form.

Definition 10. *Let (S, P) be an FPARS and $\rho_0 = \mathbf{Dirac}(s)$, where $s \in S$. Given the computation $(\rho_i)_{i \in \mathbb{N}}$, $\mathsf{Steps}_{\mathsf{P}}(s) = \sum_{i=1}^{\infty} |\rho_i|$.*

Observe that the definition above collapses to the one given for the deterministic case when P is deterministic. That it is the expected value of a random variable capturing the number of reduction steps to normal form from s can be proved by standard arguments from the theory of Markov Chains, see [9] for details. Termination is a crucial problem in rewriting theory. Since we are in a probabilistic context, distinct such notions are possible. We define in our setting two classical termination properties.

Definition 11. *An FPARS is* almost-surely terminating *(AST) if for each initial configuration ρ_0, the computation $(\rho_i)_{i\in\mathbb{N}}$, is such that $\lim_{n\to+\infty} |\rho_n| = 0$.*

Definition 12. *An FPARS (S, P) is* positive almost-surely terminating *(PAST) if for each $s \in S$, $\mathsf{Steps}_{\mathsf{P}}(s) < +\infty$. In this case we say that P is a* positive almost-surely normalising *strategy.*

Example 4. Consider the same setting of Example 3. It is easy to see that (S, P) is AST since $\lim_{n\to+\infty} |\rho_n| = \lim_{n\to+\infty} \frac{1}{2^{n-1}} = 0$. Moreover (S, P) is PAST since $\mathsf{Steps}_{\mathsf{P}}(a) = \sum_{n=1}^{\infty} |\rho_n| = \sum_{n=1}^{\infty} \frac{1}{2^{n-1}} = 2$ and $\mathsf{Steps}_{\mathsf{P}}(b) = 0$.

It is well-known from Markov chain literature that AST does not imply PAST e.g. in the symmetric random walk on $\mathbb{Z}$. We recall a standard result in Markov chain theory that gives a sufficient condition for PAST and a bound on the average number of steps to normal form.

Notation 1. For $\varepsilon > 0$ we write $x >_\varepsilon y$ if and only if $x \geq y + \varepsilon$. This order is well-founded on real numbers with a lower bound.

Definition 13. *Given an FPARS (S, P), we define a function $V : S \to \mathbb{R}$ as Lyapunov if:*

- *There exists $b \in \mathbb{R}$ such that $V(s) \geq b$ for each $s \in S$.*
- *There exists $\varepsilon > 0$ such that for every $s \in S$ if $\mathsf{P}(s) = \mu$, then $V(s) >_\varepsilon V(\mu)$, where V is extended to partial distributions as follows:*

$$V(\mu) = \sum_{t\in S} V(t) \cdot \mu(t).$$

Remark 2. Without loss of generality, given a Lyapunov function V we can always consider a new Lyapunov function $W(s) \geq 0$ for each $s \in S$ simply adding a constant to V.

Theorem 1 (Foster [6]). *If we can define for an FPARS $\mathcal{P} = (S, \mathsf{P})$ a Lyapunov function V, then $\mathcal{P}$ is PAST and the average derivation length $\mathsf{Steps}_{\mathsf{P}}(s)$ of any sequence $(\rho_i)_{i\in\mathbb{N}}$ starting from any $s \in S$ is bounded by $\frac{V(s)}{\varepsilon}$.*

4 Randomised Strategies in the λ-calculus

We define here a randomised strategy P_ε for the ARS $(\Lambda, \longrightarrow_\beta)$. Given a reducible term M, $\mathsf{P}_\varepsilon(M) = \mu$ such that for each $N \in \Lambda$:

$$\mu(N) = \begin{cases} \varepsilon & \text{if } M \longrightarrow_\beta N \text{ and the } \mathsf{LO}\text{-redex is reduced,} \\ 1-\varepsilon & \text{if } M \longrightarrow_\beta N \text{ and the } \mathsf{RI}\text{-redex is reduced,} \\ 0 & \text{otherwise.} \end{cases}$$

In this way we have defined an FPARS $(\Lambda, \mathsf{P}_\varepsilon)$.

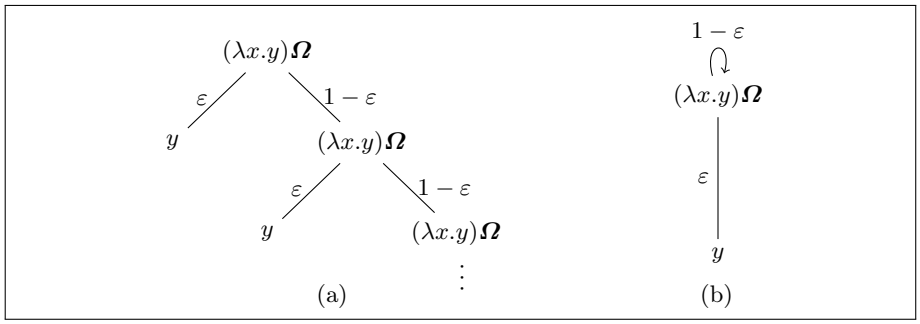

Fig. 1: The tree (a) and the cyclic graph (b) representing the reduction sequence of M.

Notation 2. For the FPARS $(\Lambda, \mathsf{P}_\varepsilon)$ we define the function $\mathsf{ExpLen}_M : [0,1] \to \mathbb{R}$ where $\mathsf{ExpLen}_M(\varepsilon) = \mathsf{Steps}_{\mathsf{P}_\varepsilon}(M)$.

Example 5. Let us consider the term $M = (\lambda x.y)\boldsymbol{\Omega}$ where $\boldsymbol{\Omega} = \boldsymbol{\omega}\boldsymbol{\omega}$ with $\boldsymbol{\omega} = \lambda x.xx$. There are two possible representations of the development of the strategy P_ε for this term, one as an infinite tree (Figure 1a) and another one as a cyclic graph (Figure 1b). According to the different representations, we can compute in different ways the probability of reaching normal form and the average derivation length. The results coincide yielding in both cases probability of termination equal to 1 and average derivation length equal to $\frac{1}{\varepsilon}$.

Theorem 2. *The FPARS $(\Lambda_{\mathsf{WN}}, \mathsf{P}_\varepsilon)$ is PAST whenever $\varepsilon > 0$.*

Proof. We use Foster's Theorem to prove the claim. Thus, we have to find a suitable Lyapunov function V. We consider $V = \mathsf{Steps}_{\mathsf{LO}}$. Certainly condition (1) is verified since $\mathsf{Steps}_{\mathsf{LO}}(M) \geq 0$ for each $M \in \Lambda_{\mathsf{WN}}$. We have to verify (2). Suppose $\mathsf{P}_\varepsilon(M) = \mu$. If $M \longrightarrow_{\mathsf{LO}} N$ and $M \longrightarrow_{\mathsf{RI}} L$, by Lemma 1 we can write:

$$\begin{aligned}
\mathsf{Steps}_{\mathsf{LO}}(\mu) &= \mathsf{Steps}_{\mathsf{LO}}(N) \cdot \varepsilon + \mathsf{Steps}_{\mathsf{LO}}(L) \cdot (1-\varepsilon) \\
&\leq (\mathsf{Steps}_{\mathsf{LO}}(M) - 1) \cdot \varepsilon + \mathsf{Steps}_{\mathsf{LO}}(M) \cdot (1-\varepsilon) \\
&= \varepsilon \cdot \mathsf{Steps}_{\mathsf{LO}}(M) - \varepsilon + \mathsf{Steps}_{\mathsf{LO}}(M) \cdot (1-\varepsilon) \\
&= \mathsf{Steps}_{\mathsf{LO}}(M) - \varepsilon.
\end{aligned}$$

Since $0 \leq \varepsilon \leq 1$, $\mathsf{Steps}_{\mathsf{LO}}(M) >_\varepsilon \mathsf{Steps}_{\mathsf{LO}}(\mu)$ for each normalising term M. Then, if $\varepsilon > 0$, $(\Lambda_{\mathsf{WN}}, \mathsf{P}_\varepsilon)$ is PAST and the average number of steps to normal form of a term M reduced with strategy P_ε is bounded by $\frac{\mathsf{Steps}_{\mathsf{LO}}(M)}{\varepsilon}$. □

The bound we obtain on $\mathsf{ExpLen}_M(\varepsilon)$ from the above proof is very loose and thus it does not give us any information on the actual nature of the function $\mathsf{ExpLen}_M(\varepsilon)$. We show, by means of an example, that the strategy P_ε is non-trivial i.e. there exists a term M and $0 < \varepsilon < 1$, such that $\mathsf{ExpLen}_M(\varepsilon) < \mathsf{ExpLen}_M(1) = \mathsf{Steps}_{\mathsf{LO}}(M) < \mathsf{ExpLen}_M(0) = \mathsf{Steps}_{\mathsf{RI}}(M)$.

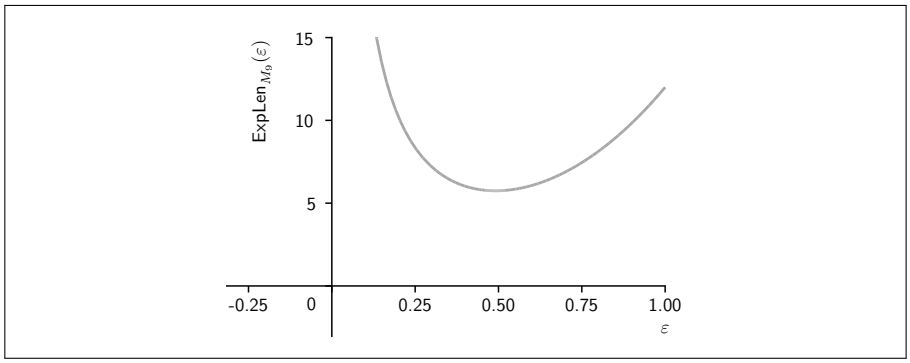

Fig. 2: The function $\mathsf{ExpLen}_{M_9}(\varepsilon)$.

Example 6. Let us consider a family of terms $M_n = NL_n$ where:

$$N = \lambda x.\underbrace{((\lambda y.z)\boldsymbol{\Omega})}_{P}x \qquad L_n = C_n\underbrace{((\lambda x.x)y)}_{S} \qquad C_n = \lambda x.\underbrace{xx\cdots x}_{n \text{ times}}$$

After quite simple computations one can derive $\mathsf{ExpLen}_{M_n}(\varepsilon) = (n-3)\varepsilon^3 + 4\varepsilon^2 + \frac{2}{\varepsilon}$. Clearly for $\varepsilon = 0$ the expression diverges. If $n \geq 2$ there is a minimum for $0 < \varepsilon < 1$, and thus $\mathsf{ExpLen}_{M_n}(\varepsilon) < \mathsf{ExpLen}_{M_n}(1) = \mathsf{Steps}_{\mathsf{LO}}(M_n) < \mathsf{ExpLen}_{M_n}(0) = \mathsf{Steps}_{\mathsf{RI}}(M_n) = +\infty$. $\mathsf{ExpLen}_{M_9}(\varepsilon)$ is plotted in Figure 2.

Studying the behaviour of $\mathsf{ExpLen}_M(\varepsilon)$ for *an arbitrary* term M is a difficult task, which goes outside the scope of this paper.

As a first step in the direction of a full understanding of the nature of $\mathsf{ExpLen}_M(\varepsilon)$, we study it in the case M is a term of the subcalculus λA we have previously introduced. In particular we show in the next section that $\mathsf{ExpLen}_M(\varepsilon)$ has minimum in $\varepsilon = 1$.

4.1 The Case of the λA-calculus

The following lemma is an easy consequence of Xi's combinatorial analysis of the Standardisation Theorem [14]:

Lemma 2. *Given a term $M \in \Lambda_A$ and a reduction sequence $M \overset{\sigma}{\twoheadrightarrow}_\beta N \longrightarrow_\beta L$, where σ is standard, we can construct a standard reduction sequence $\tau : M \twoheadrightarrow_\beta L$ such that $|\tau| \leq 1 + |\sigma|$.*

It is then easy to get the optimality of the leftmost-outermost strategy for λA-terms:

Theorem 3. *Given a reduction sequence $\sigma : M \twoheadrightarrow_\beta N$ if $M \in \Lambda_A$ and N is in normal form, then the reduction sequence $\tau : M \underset{\beta\mathsf{LO}}{\twoheadrightarrow} N$ is such that $|\tau| \leq |\sigma|$.*

Proof. By induction on $|\sigma|$. The case $|\sigma| = 0$ is trivial. So now let us suppose that the theorem holds for $|\sigma| \leq k$. Let us prove it for $|\sigma| = k+1$. We can assume that: $\sigma : M \overset{\xi}{\twoheadrightarrow}_\beta L \overset{R}{\longrightarrow}_\beta N$. By induction hypothesis we can construct a LO reduction sequence $\rho : M \twoheadrightarrow_\beta L$ such that $|\rho| \leq |\xi|$. Then, since LO reduction sequences are standard, by Lemma 2 we can build a standard reduction sequence $\tau : M \twoheadrightarrow_\beta N$ such that $|\tau| \leq 1 + |\rho| \leq 1 + |\xi| = |\sigma|$. The claim follows from the fact that standard reduction sequences to normal form are leftmost. □

Theorem 3 is a Theorem about a *deterministic* strategy, while our purpose here is to show the optimality of a *randomised* strategy. Some preliminary lemmas are necessary in order to appropriately lift it.

The following two lemmas tell us that the existence of a strictly partial probability distribution along a computation witnesses the existence of a *deterministic* computation leading to normal form:

Lemma 3. *Let (S, P) an FPARS and $(\rho_i)_{i\in\mathbb{N}}$ a computation, where $\rho_0 = \mathbf{Dirac}(s_0)$. For each $s \in S$, if there exists $k \geq 0$ such that $\rho_k(s) > 0$, then there exists a reduction sequence $s_0 \to s_1 \to \cdots \to s_{k-1} \to s$.*

Proof. We argue by induction on k. If $k = 0$, then the reduction sequence is trivially $s_0 \equiv s$. If $k = h$, $\rho_h(s) = \sum_{t\in S} \rho_{h-1}(t) \cdot \mathbb{P}(t \to s)$. Since $\rho_h(s) > 0$, there exists $t \in S$ such that $\rho_{h-1}(t) \cdot \mathbb{P}(t \to s) \neq 0$, i.e. $\rho_{h-1}(t) > 0$ and $\mathbb{P}(t \to s) > 0$. Thus, by induction hypothesis there exists a sequence $s_0 \to s_1 \to \cdots \to s_{h-2} \to t$, and $t \to s$. Hence there exists a reduction sequence $s_0 \to s_1 \to \cdots \to s_{h-2} \to t \to s$. □

Lemma 4. *Let (S, P) an FPARS and $(\rho_i)_{i\in\mathbb{N}}$ a computation, where $\rho_0 = \mathbf{Dirac}(s_0)$. If there exists $k \geq 1$ such that $|\rho_k| < 1$, then there exists a sequence $s_0 \to s_1 \to \cdots \to s_j$ such that s_j is in normal form and $j \leq k-1$.*

Proof. We argue by induction on k. If $k = 1$, since $|\rho_0|$ is $\mathbf{Dirac}(s_0)$ then $\mathsf{P}(s_0) = \bot$ (otherwise $|\rho_1| = 1$). Hence s_0 is in normal form. If $k = h$ and $|\rho_{h-1}| < 1$ by induction hypothesis we are done. So let us consider the case in which $|\rho_{h-1}| = 1$ and $|\rho_h| < 1$. We claim that there exists $t \in \mathbf{NF}(S)$ such that $\rho_{h-1}(t) > 0$.

$$
\begin{aligned}
|\rho_h| &= \sum_{s\in S}\sum_{t\in S} \rho_{h-1}(t) \cdot \mathbb{P}(t \to s) \\
&= \sum_{t\in S}\sum_{s\in S} \rho_{h-1}(t) \cdot \mathbb{P}(t \to s) = \sum_{t\in S}\left(\rho_{h-1}(t)\sum_{s\in S}\mathbb{P}(t \to s)\right) \\
&= \sum_{t\notin \mathbf{NF}(S)}\left(\rho_{h-1}(t)\sum_{s\in S}\mathbb{P}(t \to s)\right) + \sum_{t\in \mathbf{NF}(S)}\left(\rho_{h-1}(t)\sum_{s\in S}\mathbb{P}(t \to s)\right).
\end{aligned}
$$

If there was not $t \in \mathbf{NF}(S)$ such that $\rho_{h-1}(t) > 0$, then the second term in the sum would vanish and $|\rho_h| = \sum_{t\notin \mathbf{NF}(S)} \rho_{h-1}(t) = 1$. But $|\rho_h| < 1$ by hypothesis.

Hence there exists $t \in \mathbf{NF}(S)$ such that $\rho_{h-1}(t) > 0$ and thus by Lemma 3 there exist a sequence $s_0 \to s_1 \to \cdots \to s_{h-2} \to t$. □

We are almost done: the following lemma tells us that all configurations along a computation starting from a λA-term M are proper until the n-th configuration, where $n = \mathsf{Steps}_{\mathsf{LO}}(M)$.

Lemma 5. *Given the FPARS* $(\Lambda_A, \mathsf{P}_\varepsilon)$*, and a computation* $(\rho_i)_{i\in\mathbb{N}}$*, where* $\rho_0 = \mathbf{Dirac}(M_0)$*, for each* $k \leq \mathsf{Steps}_{\mathsf{LO}}(M_0)$*, then* $|\rho_k| = 1$*.*

Proof. Let $n = \mathsf{Steps}_{\mathsf{LO}}(M_0)$. By contradiction if there was $k \leq n$ such that $|\rho_k| < 1$, then by Lemma 4 would exist a sequence $M_0 \longrightarrow_\beta M_1 \longrightarrow_\beta \cdots \longrightarrow_\beta M_j$ such that $j \leq k-1$ and M_j is normal form. But this is impossible from Theorem 3. □

Corollary 1. *For each term* M *in* Λ_A*,* $\mathsf{ExpLen}_M(\varepsilon)$ *has minimum in* $\varepsilon = 1$*.*

Proof. Let $n = \mathsf{Steps}_{\mathsf{LO}}(M) = \mathsf{ExpLen}_M(1)$.

$$\begin{aligned}\mathsf{ExpLen}_M(\varepsilon) &= \sum_{i=1}^{\infty} |\rho_i| = \sum_{i=1}^{n} |\rho_i| + \sum_{i=n+1}^{\infty} |\rho_i| \overset{\text{Lemma 5}}{=} \sum_{i=1}^{n} 1 + \sum_{i=n+1}^{\infty} |\rho_i| \\ &= n + \sum_{i=n+1}^{\infty} |\rho_i| = \mathsf{ExpLen}_M(1) + \sum_{i=n+1}^{\infty} |\rho_i| \geq \mathsf{ExpLen}_M(1).\end{aligned}$$

□

4.2 The Case of the λI-calculus

In terms of the λI-calculus all redexes are *needed* to compute normal forms and redexes can be duplicated. One might thus guess that the rightmost-innermost strategy could be optimal there. However, innermost strategies, which do not reduce a redex unless its argument is in normal form, are not optimal [1]. A simple counterexample is shown in Figure 3: intuitively, RI is not optimal in this case because the *virtual* redex yz is copied too early.

5 Conclusions

In this work we have initiated the study of randomised reduction strategies for the λ-calculus. We have defined a family of examples of such strategies, and we have shown that all of them, except one, are positive almost-surely normalising. Then we have studied those strategies for λA, the affine λ-calculus, proving optimality results.

Further work could consist in analysing the behaviour of the proposed strategies in the scope of the full λ-calculus. In particular one could try to characterize classes of λ-terms for which our strategies work strictly better than deterministic ones, and to develop some methods to tune the parameter ε in order to get good performances.

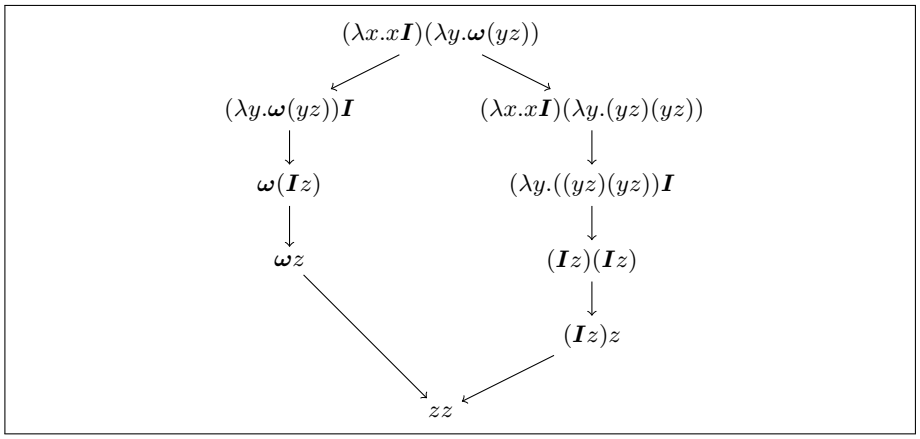

Fig. 3: The optimal reduction sequence of the term $M = (\lambda x.x\boldsymbol{I})(\lambda y.\boldsymbol{\omega}(yz))$, 4 steps, on the left, and its reduction under rightmost-innermost strategy, 5 steps, on the right.

References

1. Asperti, A., Guerrini, S.: The Optimal Implementation of Functional Programming Languages. Cambridge University Press (1998)
2. Avanzini, M., Dal Lago, U., Yamada, A.: On Probabilistic Term Rewriting. In: Proc. of 14th FLOPS. LNCS, vol. 10818, pp. 132–148. Springer (2018)
3. Barendregt, H.P.: The lambda calculus: its syntax and semantics. North-Holland (1984)
4. Bournez, O., Garnier, F.: Proving Positive Almost-Sure Termination. In: Proc. of 16th RTA. LNCS, vol. 3467, pp. 323–337. Springer (2005)
5. Bournez, O., Kirchner, C.: Probabilistic Rewrite Strategies. Applications to ELAN. In: Proc. of 13th RTA. LNCS, vol. 2378, pp. 252–266. Springer (2002)
6. Bremaud, P.: Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues. Springer-Verlag (1999)
7. Church, A.: An Unsolvable Problem of Elementary Number Theory. American Journal of Mathematics **58**(2), 345–363 (1936)
8. Curry, H.B., Feys, R.: Combinatory Logic. North-Holland (1958)
9. Dal Lago, U., Vanoni, G.: On Randomised Strategies in the λ-Calculus (Long Version) (2018), Available at: http://arxiv.org/abs/1805.03934
10. Ferrer Fioriti, L.M., Hermanns, H.: Probabilistic Termination: Soundness, Completeness, and Compositionality. In: Proc. of 42nd POPL. pp. 489–501. ACM (2015)
11. Motwani, R., Raghavan, P.: Randomized Algorithms. Cambridge University Press (1995)
12. Sinot, F.R.: Sub-λ-calculi, Classified. Electronic Notes in Theoretical Computer Science **203**(1), 123–133 (2008)
13. Terese: Term Rewriting Systems. Cambridge University Press (2003)
14. Xi, H.: Upper bounds for standardizations and an application. The Journal of Symbolic Logic **64**(1), 291–303 (1999)