Metric Reasoning About λ-Terms: The General Case

Raphaëlle Crubillé, Ugo Dal Lago

Metric Reasoning About λ-Terms: The General Case*

Raphaëlle Crubillé[1] and Ugo Dal Lago[2]

[1] IRIF, Université Denis Diderot - Paris 7
rcrubille@pps.univ-paris-diderot.fr
[2] Università di Bologna & INRIA Sophia Antipolis
ugo.dallago@unibo.it

Abstract. In any setting in which observable properties have a quantitative flavor, it is natural to compare computational objects by way of *metrics* rather than *equivalences* or *partial orders*. This holds, in particular, for probabilistic higher-order programs. A natural notion of comparison, then, becomes context *distance*, the metric analogue of Morris' context *equivalence*. In this paper, we analyze the main properties of the context distance in fully-fledged probabilistic λ-calculi, this way going beyond the state of the art, in which only affine calculi were considered. We first of all study to which extent the context distance trivializes, giving a sufficient condition for trivialization. We then characterize context distance by way of a coinductively-defined, *tuple-based* notion of distance in one of those calculi, called $\Lambda_!^{\oplus}$. We finally derive pseudometrics for call-by-name and call-by-value probabilistic λ-calculi, and prove them fully-abstract.

1 Introduction

Probability theory offers computer science models which enable system abstraction (at the price of introducing uncertainty), but which can also be seen as a *a way to compute*, like in randomized computation. Domains in which probabilistic models play a key role include machine learning [27], robotics [34], and linguistics [24]. In cryptography, on the other hand, having access to a source of uniform randomness is essential to achieve security, e.g., in the public key setting [20]. This has stimulated the development of concrete and abstract programming languages, which most often are extensions of their deterministic siblings. Among the many ways probabilistic choice can be captured in programming, the simplest one consists in endowing the language of programs with an operator modeling the flipping of a fair coin. This renders program evaluation a probabilistic process, and under mild assumptions the language becomes universal for probabilistic computation. Particularly fruitful in this sense has been the line of work on the functional paradigm, both at a theoretical [22, 29, 26] and at a more practical level [21].

* This work is partially supported by the ANR projects 12IS02001 PACE, 14CE250005 ELICA, and 16CE250011 REPAS.

We are still far, however, from a satisfactory understanding of higher-order probabilistic computation. As an example, little is known about how much of the classic, beautiful, theory underlying the λ-calculus [1] can be lifted to probabilistic λ-calculi, although the latter have been known from forty years now [30]. Until the beginning of this decade, indeed, most investigations were directed towards domain theory, which has been proved to be much more involved in presence of probabilistic choice than in a deterministic scenario [23]. In the last ten years, however, some promising results have appeared. As an example, both quantitative semantics and applicative bisimilarity have been shown to coincide with context equivalence for certain kinds of probabilistic λ-calculi [14, 5]. This not only provides us with new proof methodologies for program equivalence, but also sheds new light on the very nature of probabilistic higher-order computation. As an example, recent results tell us that program equivalence in presence of probabilistic choice lies somehow in between determinism and non-determinism [5].

But are equivalences the most proper way to compare terms? Actually, this really depends on what the underlying observable is. If observables are boolean, then equivalences (and preorders) are indeed natural choices: two programs are dubbed equivalent if they give rise to the same observable (of which there are just two!) in any context. If, on the other hand, the observable is an element of a metric space, which happens for example when we observe (the probability of) convergence in a probabilistic setting, one may wonder whether replacing equivalences with metrics makes sense. This is a question that has recently been given a positive answer in the *affine* setting [6], i.e., in a λ-calculus in which copying is simply not available. More specifically, a notion of context distance has been shown to model differences between terms satisfactorily, and has also been shown to be characterized by notions of trace metrics, and to be approximated from below by behavioral metrics.

Affine λ-calculi are very poor in terms of the computations they are able to model. Measuring the distance between terms in presence of copying, however, is bound to be problematic. On the one hand, allowing contexts to copy their argument has the potential risk of *trivializing* the underlying metric. On the other hand, finding handier characterizations of the obtained notion of metric in the style of behavioral or trace metrics is inherently hard. A more thorough discussion on these issues can be found in Section 2 below.

In this paper, we attack the problem of analyzing the distance between λ-terms in its full generality. More specifically, the contributions of this paper are fourfold:

- First of all, we define a linear probabilistic λ-calculus, called $\Lambda^{!,\|}_{\oplus}$, in which copying and a nonstandard construct, namely Plotkin's parallel disjunction, are both available. A very liberal type system prevents deadlocks, but nevertheless leaves the expressive power of the calculus very high. This choice has been motivated by our will to put ourselves in the most general setting, so as to be able to talk about different fragments. The calculus is endowed with

a notion of context distance, in Morris' style. This is covered in Section 3 below.
- We study trivialization of the obtained notion(s) of metric for different fragments of $\Lambda^{!,\|}_{\oplus}$, showing that both parallel disjunction and strong normalization give us precisely the kind of discriminating power we need to arbitrarily amplify distances, while in the most natural fragment, namely $\Lambda^{!}_{\oplus}$, trivialization does *not* hold. This is the subject of Section 4.
- In Section 5, we prove that context distance can be characterized as a coinductively-defined distance on a labeled Markov chain of *tuples*. The way (tuples of) terms interact with their environment makes proofs of soundness laborious and different from their affine counterparts from [6]. An up-to-context notion of bisimulation is proved to be sound, and to be quite useful when evaluating the distance between concrete programs.
- Finally, we show that the results from Section 5 can be lifted back to ordinary probabilistic λ-calculi from the literature [10, 5]. Both when call-by-name evaluation and call-by-value are considered, our framework can be naturally adapted, and helps in facilitating concrete proofs. This is in Section 6.

More details can be found in a long version of this paper, available online [7].

2 Metrics and Trivialization, Informally

The easiest way to render the pure λ-calculus a universal probabilistic computation model [10] consists in endowing it with a binary construct $\oplus$ for probabilistic choice. The term $M \oplus N$ evolves as either M or N, each with probability $\frac{1}{2}$. The obtained calculus can be given meaning by an operational semantics which puts terms in correspondence with *distributions of* values. The natural notion of observation, at least in an untyped setting like the one we will consider in this paper, is thus the *probability of convergence* of the observed term M, which will be denoted as $\sum_{[\![M]\!]}$. One could then define a notion of *context equivalence* following Morris' pattern, and stipulate that two terms M and N should be equivalent whenever they terminate with *exactly* the same probability when put in *any* context:

$$M \equiv N \;\Leftrightarrow\; \forall C. \sum\nolimits_{[\![C[M]]\!]} = \sum\nolimits_{[\![C[N]]\!]}.$$

The anatomy of the obtained notion of equivalence has been recently studied extensively, the by-products of this study being powerful techniques for it in the style of bisimilarity and logical relations [9, 5, 3].

As observed by various authors (see, e.g., [25] for a nice account), probabilistic programs and processes are naturally compared by *metrics* rather than *equivalences*: the latter do not give any quantitative information about *how different* two non-equivalent programs are. Given that the underlying notion of observation is inherently quantitative, on the other hand, generalizing context equivalence to a *pseudometric* turns out to be relatively simple:

$$\delta(M, N) = \sup_C \left| \sum\nolimits_{[\![C[M]]\!]} - \sum\nolimits_{[\![C[N]]\!]} \right|.$$

Observe that the obtained notion of context *distance* between two terms is a real number between 0 and 1, which is minimal precisely when the considered terms are context equivalent. It is the least discriminating pseudometric which is non-expansive and adequate, and as such it provides some quite precise information about how far the two argument programs are, observationally. A similar notion has recently been studied by the authors [6], but only in a purely affine setting.

Let us now consider two prototypical examples of non-equivalent terms, namely $I = \lambda x.x$ (the identity) and Ω (the always-divergent term). The context distance $\delta^c(I, \Omega)$ between them is maximal: when applied, e.g., to the trivial context $[\cdot]$, they converge with probability 1 and 0, respectively. A term which is conceptually "in the middle" of them is $M = I \oplus \Omega$. Indeed, in a purely affine λ-calculus, $\delta^c(I, M) = \delta^c(M, \Omega) = \frac{1}{2}$.

If we render the three terms duplicable (by putting them in the scope of a !-operator), however, the situation becomes much more complicated. Consider the terms $!I$ and $!(I \oplus \Omega)$. One can easily define a family of contexts $\{C_n\}_{n\in\mathbb{N}}$ such that the probability of convergence of $C_n[!I]$ and $C_n[!(I \oplus \Omega)]$ tend to 1 and 0 (respectively) when n tends to infinity. It suffices to take C_n as $(\lambda!x.\ \underbrace{x \ldots x}_{n \text{ times}})[\cdot]$. Allowing contexts to have the capability to duplicate their argument seems to mean that they can arbitrarily *amplify* distances. Indeed, the argument above also works when $(I \oplus \Omega)$ is replaced by any term which behaves as Ω with probability ε and as I with probability $1 - \varepsilon$, provided of course $\varepsilon > 0$. But how about $!\Omega$ and $!(I \oplus \Omega)$? Are they at maximal distance, i.e. is it that $\delta^c(!\Omega, !(I \oplus \Omega)) = 1$? Apparently, this is *not* the case. The previously defined contexts C_n cannot amplify the "linear" distance between the two terms above, namely $\frac{1}{2}$, up to 1. But what is the distance between $!\Omega$ and $!(I \oplus \Omega)$, then? Evaluating it is hard, since you need to consider all contexts, which do not have a nice structure. In Section 5, we will introduce a different, better behaved, notion of distance, this way being able to prove that, indeed, $\delta^c(!\Omega, !(I \oplus \Omega)) = \frac{1}{2}$.

All this hints at even more difficult examples, like the one in which $M_\varepsilon = !(\Omega \oplus^\varepsilon I)$, where $\oplus^\varepsilon$ is the natural generalization of $\oplus$ to a possibly unfair coin flip, and one is interested in evaluating $\delta^c(M_\varepsilon, M_\mu)$. In that case, we can easily see that the "linear" distance between them is $|\varepsilon - \mu|$. In some cases, it is possible to amplify it: the most natural way is again to consider the contexts C_n defined above. Indeed, we see that the probabilities of convergence of $C_n[M_\varepsilon]$ and $C_n[M_\mu]$ are ε^n and μ^n, respectively. It follows that $\delta^c(M_\varepsilon, M_\mu) \geq \sup_{n\in\mathbb{N}} |\varepsilon^n - \mu^n|$. For some ε and μ (for example if $\varepsilon + \mu > 1$), the context distance can be greater than $|\varepsilon - \mu|$. But there is no easy way to know *how far* amplification can lead us. The terms M_ε and M_μ will be running examples in the course of this paper. Despite their simplicity, evaluating the distance between them is quite challenging.

We are also going to consider the case in which contexts can evaluate terms *in parallel*, converging if and only if at least one of the copies converges. This behavior is not expressible in the usual λ-calculus, but is captured by well-known constructs and in particular by Plotkin's parallel disjunction [28]. In Section 4 below, we prove that all this is not accidental: the presence of parallel disjunction

turns a non-trivializing metric into a trivializing one. The proof of it, by the way, relies on building certain amplifying contexts which are then shown to be universal using tools from functional analysis.

3 A Linear Probabilistic λ-Calculus

In this section, we present the syntax and operational semantics of our language $\Lambda^{!,\|}_{\oplus}$, on which we will later define metrics. $\Lambda^{!,\|}_{\oplus}$ is a probabilistic and linear λ-calculus, designed not only to allow copying, but to have a better control on it. It is based on a probabilistic variation of the calculus defined in [33], whose main feature is to never reduce inside exponential boxes. As we will see in Section 6, the calculus is capable of encoding both call-by-value and call-by-name fully-fledged probabilistic λ-calculi. We add a parallel disjunction construct to the calculus, being inspired by Plotkin's parallel disjunction [28]. Noticeably, it has been recently shown [8] that adding parallel disjunction to a (non-linear) λ-calculus increases the expressive power of contexts to the point of enabling coincidence between the contextual preorder and applicative similarity. The choice of studying a very general calculus is motivated by our desire to be as general as possible. This being said, many of our results hold only in *absence* of parallel disjunction.

Definition 1. *We assume a countable set of variables $\mathscr{X}$. The set of* terms *of $\Lambda^{!,\|}_{\oplus}$ (denoted $\mathscr{T}$) is defined by the following grammar:*

$$M \in \mathscr{T} ::= x \;\Big|\; MM \;\Big|\; \lambda x.M \;\Big|\; \lambda !x.M \;\Big|\; !M \;\Big|\; M \oplus M \;\Big|\; ([M \parallel M] \rightarrowtail M)\ ,$$

where $x \in \mathscr{X}$. The fragment of $\Lambda^{!,\|}_{\oplus}$ without the $([\cdot \parallel \cdot] \rightarrowtail \cdot)$ construct will be indicated as $\Lambda^{!}_{\oplus}$. Values *are those terms derived from the following grammar:*

$$V \in \mathscr{V} ::= \lambda x.M \;\Big|\; \lambda !x.M \;\Big|\; !M.$$

As already mentioned, $M \oplus N$ can evolve to either M or N, each with probability $\frac{1}{2}$. The term $!M$ is a duplicable version of M, often called an *(exponential) box*. We have two distinct abstraction operators: $\lambda x.M$ is a *linear* abstraction, while the *non-linear* abstraction $\lambda !x.M$ requires exponential boxes as arguments. The term $([M \parallel N] \rightarrowtail L)$ behaves as L if either M or N converges. Please observe that both abstractions and boxes are values—our notion of reduction is *weak* and *surface* [33].

We are now going to define an operational semantics for the closed terms of $\Lambda^{!,\|}_{\oplus}$ in a way similar to the one developed for a (non-linear) λ-calculus in [10]. We need to first define a family of *approximation semantics*, and then to take the semantics of a term as the least upper bound of all its approximations. The approximation semantics relation is denoted $M \Rightarrow \mathcal{D}$, where M is a closed term

of $\Lambda^{!,\|}_{\oplus}$, and $\mathcal{D}$ is a *(sub)distribution on values* with finite support, i.e., a function from $\mathscr{V}$ to $\mathbb{R}_{[0,1]}$ which sums to a real number $\sum_{\mathcal{D}} \leq 1$. For any distribution $\mathcal{D}$ on a set X, we call *support of* $\mathcal{D}$, and we note $\mathsf{S}(\mathcal{D})$, the set $\{x \in X \mid \mathcal{D}(x) > 0\}$. We say that $\mathcal{D}$ is *finite* if $\mathsf{S}(\mathcal{D})$ is a finite set.

The rules deriving the approximation semantics relation are given in Figure 1, and are based on the notion of an *evaluation context*, which is an expression generated from the following grammar:

$$E ::= [\cdot] \;\Big|\; EV \;\Big|\; ME \;\Big|\; ([M \parallel E] \rightarrowtail N) \;\Big|\; ([E \parallel M] \rightarrowtail N)\ .$$

As usual, $E[M]$ stands for the term obtained by filling the sole occurrence of $[\cdot]$ in E with M. In Figure 1 and elsewhere in this paper, we indicate the distribution assigning probability p_i to V_i for every $i \in \{1, \ldots, n\}$ as $\{V_1^{p_1}, \ldots, V_n^{p_n}\}$. We proceed similarly for the expression $\{V_i^{p_i}\}_{i \in I}$, where I is any countable index set. Observe how we first define a one-step reduction relation $\cdot \to \cdot$ between closed terms and *sequences* of terms, only later extending it to a small-step reduction relation $\cdot \Rightarrow \cdot$ between closed terms and *distributions* on values. A reduction step

$$\frac{}{M \oplus N \hookrightarrow M, N} \qquad \frac{}{(\lambda x.M)V \hookrightarrow M\{x/V\}} \qquad \frac{}{(\lambda !x.M)!N \hookrightarrow M\{x/N\}}$$

$$\frac{}{([V \parallel M] \rightarrowtail N) \hookrightarrow N} \qquad \frac{}{([M \parallel V] \rightarrowtail N) \hookrightarrow N} \qquad \frac{M \hookrightarrow N_1, \ldots, N_n}{E[M] \to E[N_1], \ldots E[N_n]}$$

$$\frac{}{V \Rightarrow \{V^1\}} \qquad \frac{}{M \Rightarrow \emptyset} \qquad \frac{M \to N_1, \ldots, N_n \qquad (N_i \Rightarrow \mathcal{D}_i)_{1 \leq i \leq n}}{M \Rightarrow \sum_{1 \leq i \leq n} \frac{1}{n} \cdot \mathcal{D}_i}$$

Fig. 1. Approximation Semantics for $\Lambda^{!}_{\oplus}$

can be a linear or non-linear β-reduction, or a probabilistic choice. Moreover, there can be more than one active redex in any closed term M, due to the presence of parallel disjunction. For any term M, the set of sub-distributions $\mathcal{D}$ such that $M \Rightarrow \mathcal{D}$ is a countable directed set. Since the set of sub-distributions (with potentially infinite support) is an ω-complete partial order, we can define the *semantics* of a term M as $[\![M]\!] = \sup\{\mathcal{D} \mid M \Rightarrow \mathcal{D}\}$. We could also define alternatively a big-step semantics, again in the same way as that of the probabilistic λ-calculus considered in [10].

Not all irreducible terms are values in $\Lambda^{!,\|}_{\oplus}$, e.g. $(\lambda !x.x)(\lambda x.x)$. We thus need a *type-system* which guarantees the absence of deadlocks. Since we want to be as general as possible, we consider recursive types as formulated in [2], which are expressive enough to type the image of the embeddings we will study in

Section 6. The grammar of *types* is the following:

$$\sigma \in \mathscr{A} ::= \alpha \;\Big|\; \mu\alpha.\sigma \multimap \sigma \;\Big|\; \mu\alpha.!\sigma \;\Big|\; \sigma \multimap \sigma \;\Big|\; !\sigma$$

Types are defined up to the equality $=^{\mathscr{A}}$, defined in Figure 2. $\sigma[\alpha \to \tau]$ stands for the type obtained by substituting all free occurrences of α by τ in σ. An

$$\frac{}{\mu\alpha.\sigma \multimap \tau =^{\mathscr{A}} \sigma[\alpha \to (\mu\alpha.\sigma \multimap \tau)] \multimap \tau[\alpha \to (\mu\alpha.\sigma \multimap \tau)]}$$

$$\frac{}{\mu\alpha.!\sigma =^{\mathscr{A}} !(\sigma[\alpha \to \mu\alpha.!\sigma])} \qquad \frac{\sigma =^{\mathscr{A}} \gamma[\alpha \to \sigma] \qquad \tau =^{\mathscr{A}} \gamma[\alpha \to \tau]}{\sigma =^{\mathscr{A}} \tau}$$

Fig. 2. Equality of Types

environment is a set of expressions in the form $x : \sigma$ or $!x :!\sigma$ in which any variable occurs at most once. Environments are often indicated with metavariables like $!\Gamma$, which stands for an environment in which all variables occur as $!x$, or Δ in which, on the contrary, variables can *only* occur with the shape x, so that Δ is of the form $x_1 : \sigma_1, \ldots, x_n : \sigma_n$. *Typing judgments* are thus of the form $!\Gamma, \Delta \vdash M : \sigma$. The typing system is given in Figure 3. The role of this type system is *not* to guarantee termination, but rather to guarantee a form of type soundness:

Lemma 1. *If $\vdash M : \sigma$ and $M \Rightarrow \mathcal{D}$, then $\vdash V : \sigma$ for every V in the support of $\mathcal{D}$. Moreover, if $\vdash M : \sigma$ and M is irreducible (i.e. $M \not\to N$ for every N), then M is value.*

Example 1. The term $I = \lambda x.x$ can be typed as $\vdash I : \sigma \multimap \sigma$ for every $\sigma \in \mathscr{A}$. We define $\Omega_!$ to be the term $(\lambda!x.x!x)(!(\lambda!x.x!x))$, which is the counterpart in our linear calculus of the prototypical diverging term of the λ-calculus, namely $\Omega = (\lambda x.xx)(\lambda x.xx)$. We can type this divergent term with any possible type: indeed, if we take $\tau ::= \mu\alpha.!\alpha \multimap \sigma$, then $\tau =^{\mathscr{A}} !\tau \multimap \sigma$ and $\vdash \lambda!x.x!x : \sigma$. Using that, we can see that $\vdash \Omega_! : \sigma$ for every type σ. We will see in Section 6 that, more generally, there are several ways to turn any pure λ-term M into a $\Lambda^!_\oplus$ term in such a way as to obtain meaningful typing and semantics: $\Lambda^!_\oplus$ is actually at least as powerful as the usual untyped probabilistic λ-calculus [10].

Termination could in principle be guaranteed if one considers *strictly positive* types, as we will do in Section 4.1 below. Let $\mathbb{D}$ be the set of dyadic numbers (i.e. those rational numbers in the form $\frac{n}{2^m}$ (with $n, m \in \mathbb{N}$ and $n \leq 2^m$). It is easy to derive, for every $\varepsilon \in \mathbb{D}$, a new binary operator on terms $\cdot \oplus^\varepsilon \cdot$ such that $[\![M \oplus^\varepsilon N]\!] = (1-\varepsilon)[\![M]\!] + \varepsilon[\![N]\!]$ for every closed M, N.

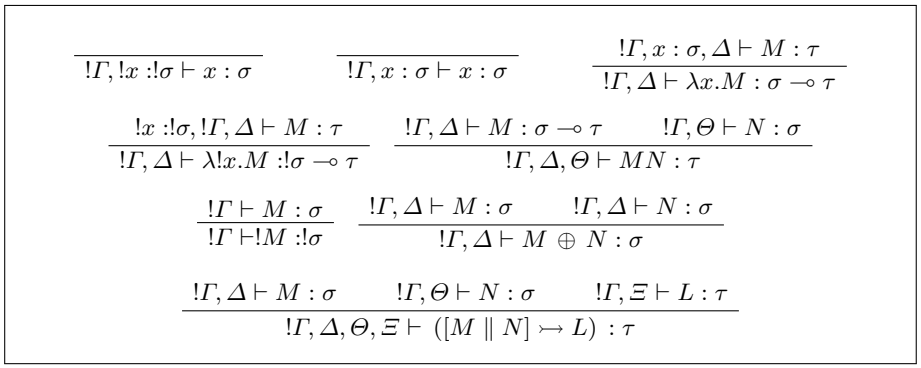

Fig. 3. Typing Rules

Example 2. We define here a family of terms that we use as a running example. We consider terms of the form $M_\varepsilon = !(\Omega_! \oplus^\varepsilon I)$, for $\varepsilon \in \mathbb{D}$. It holds that $\vdash M_\varepsilon : !(\sigma \multimap \sigma)$ for every σ. M_ε corresponds to a duplicable term, each copy of which behaves as I with probability ε, and does not terminate with probability $1 - \varepsilon$.

3.1 Some Useful Terminology and Notation

In this paper, we will make heavy use of sequences of terms and types. It is thus convenient to introduce some terminology and notation about them.

A finite (ordered) sequence whose elements are $e_1, \ldots, e_n$ will be indicated as $\boldsymbol{e} = [e_1, \ldots, e_n]$, and called an *$n$-sequence*. Metavariables for sequences are boldface variations of the metavariables for their elements. Whenever $E = \{i_1, \ldots, i_m\} \subseteq \{1, \ldots, n\}$ and $i_1 < \ldots < i_m$, the sub-sequence $[e_{i_1}, \ldots, e_{i_m}]$ of an n-sequence $\boldsymbol{e}$ will be indicated as $\boldsymbol{e}_E$. If the above holds, E will be called an *n-set*. If $\boldsymbol{e}$ is an n-sequence, and φ is a permutation on $\{1, \ldots, n\}$, we note e_φ the n-sequence $[e_{\varphi(1)}, \ldots, e_{\varphi(n)}]$. We can turn an n-sequence into an $(n+1)$-sequence by adding an element at the end: this is the role of the semicolon operator. We denote by $[e^n]$ the n-sequence in which all components are equal to e.

Whenever this does not cause ambiguity, notations like the ones above will be used in conjunction with syntactic constructions. For example, if $\boldsymbol{\sigma}$ is an n-sequence of types, then $!\boldsymbol{\sigma}$ stands for the sequence $[!\sigma_1, \ldots, !\sigma_n]$. As another example, if $\boldsymbol{\sigma}$ is an n-sequence of types and E is an n-set, then $\boldsymbol{x}_E : \boldsymbol{\sigma}_E$ stands for the environment assigning type σ_i to x_i for every $i \in E$. As a final example, if $\boldsymbol{M}$ is an n-sequence of terms and $\boldsymbol{\sigma}$ is an n-sequence of types, $\vdash \boldsymbol{M} : \boldsymbol{\sigma}$ holds iff $\vdash M_i : \sigma_i$ is provable for every $i \in \{1, \ldots, n\}$.

3.2 Context Distance

A *context of type σ for terms of type τ* is a term C which can be typed as $\mathit{hole} : \tau \vdash C : \sigma$, where *hole* is a distinguished variable. $\mathcal{C}^\tau_\sigma$ collects all such terms.

If $C \in \mathcal{C}^\tau_\sigma$ and M is a closed term of type τ, then the closed term $C\{hole/M\}$ has type σ and is often indicated as $C[M]$.

The *context distance* [6] is the natural quantitative refinement of context equivalence. Intuitively, it corresponds to the maximum separation that contexts can induce between two terms. Following [6], we take as observable the probability of convergence: for any term M, we define its *observable* $\mathrm{Obs}(M)$ as $\sum_{[\![M]\!]}$. Then, for any terms M, N such that $\vdash M : \sigma$ and $\vdash N : \sigma$, we define:

$$\delta^c_{\sigma,!,\|}(M,N) = \sup_{C \in \mathcal{C}^\tau_\sigma} |\mathrm{Obs}(C[M]) - \mathrm{Obs}(C[N])|.$$

Please observe that this distance is a pseudometric, and that moreover we can recover context equivalence by considering its *kernel*, that is the set of pairs of terms which are at distance 0. The binary operator $\delta^c_{\sigma,!}$ is defined similarly, but referring to terms (and contexts) from $\Lambda^!_\oplus$.

Example 3. What can we say about $\delta^c_{\sigma,!,\|}(M_\varepsilon, M_\mu)$? Not much apparently, since *all* contexts should be considered. Even if we put ourselves in the fragment $\Lambda^!_\oplus$, the best we can do is to conclude that $\delta^c_{\sigma,!}(M,N) \geq \sup_{n\in\mathbb{N}} |\varepsilon^n - \mu^n|$, as explained in Section 2.

4 On Trivialization

As we have already mentioned, there can well be classes of terms such that the context distance collapses to context equivalence, due to the copying abilities of the language. The question of trivialization can in fact be seen as a question about the expressive power of contexts: given two duplicable terms, how much can a context amplify the observable differences between their behaviors?

More precisely, we would like to identify *trivializing* fragments of $\Lambda^{!,\|}_\oplus$, that is to say fragments such that for any pair of duplicable terms, their context distance (with respect to the fragment) is either 0 or 1. This is not the case in $\Lambda^!_\oplus$ (see Example 8 below).

In fact, a sufficient condition to trivialization is to require the existence of *amplification contexts*: for every observable type σ, for every $\alpha, \beta \in [0,1]$ distinct, for every $\gamma > 0$, we want to have a context $C^{\alpha,\beta,\gamma}_\sigma$ such that:

$$\left.\begin{array}{r} \vdash M, N : \sigma \\ \mathrm{Obs}(M) = \alpha \\ \mathrm{Obs}(N) = \beta \end{array}\right\} \Rightarrow \left|\mathrm{Obs}(C^{\alpha,\beta,\gamma}_\sigma[!M]) - \mathrm{Obs}(C^{\alpha,\beta,\gamma}_\sigma[!N])\right| \geq 1 - \gamma.$$

Fact 1 *Any fragment of $\Lambda^{!,\|}_\oplus$ admitting all amplifications contexts trivializes.*

4.1 Strictly Positive Types

First, let us consider the case of the fragment $\Lambda^{!,\downarrow}_\oplus$ of $\Lambda^!_\oplus$ obtained by considering strictly positive types, only (in a similar way to [2]), and by dropping parallel

disjunction. Every term M of $\Lambda^{!,\downarrow}_{\oplus}$ is terminating (i.e. $\sum[\![M]\!] = 1$), so we need to adapt our notion of observation: we define the type $\mathbb{B} = !\alpha \multimap !\alpha \multimap \alpha$, which can be seen as boolean type using a variant of the usual boolean encoding in λ-calculi. Our new notion of observation, defined only at type $\mathbb{B}$, is $\mathrm{Obs}(M) = \sum_{[\![M!I!\Omega_!]\!]}$, which corresponds to the probability that M evaluates to **true**. While this notion of observation uses the full power of $\Lambda^{!}_{\oplus}$, the context distance $\delta^c_{!,\downarrow}$ based on it only consider contexts in $\Lambda^{!,\downarrow}_{\oplus}$.

Theorem 1. *$\delta^c_{!,\downarrow}$ trivializes.*

The proof of Theorem 1 is based on the construction of amplification contexts. We are going to use Bernstein constructive proof of the Stone-Weierstrass theorem. Indeed, Bernstein showed that for every continuous function $f : [0,1] \to \mathbb{R}$, the following sequence of polynomials converges uniformly towards f:

$$P^f_n(x) = \sum\nolimits_{0 \leq k \leq n} f\left(\frac{k}{n}\right) \cdot B^n_k(x), \text{ where } B^n_k(x) = \binom{n}{k} \cdot x^k \cdot (1-x)^{n-k}.$$

Let us consider the following continuous function: we fix $f(\alpha) = 0$, $f(\beta) = 1$, and f defined elsewhere in such a way that it is continuous, that it has values in $[0,1]$, and that moreover $f(\mathbb{Q}) \subseteq \mathbb{Q}$. We can easily implement P^f_n by a context, i.e. define C such that for every M, $\mathrm{Obs}(C[M]) = P^f_n(\mathrm{Obs}(M))$. In $\Lambda^{!,\downarrow}_{\oplus}$, we can indeed copy an argument n times, then evaluate it, and then for every k between 0 and n, if the number of **true**s obtained is exactly k, return the term **false** $\oplus^{f(\frac{k}{n})}$ **true** (that corresponds to a term returning **true** with probability $f\left(\frac{k}{n}\right)$). Please observe that this construction works only because in $\Lambda^{!,\downarrow}_{\oplus}$ all terms converge with probability 1.

4.2 Parallel Disjunction

As we have seen, trivialization can be enforced by restricting the class of terms, but we can also take the opposite road, namely increasing the discriminating power of contexts. Indeed, consider the full language $\Lambda^{!,\|}_{\oplus}$, with the usual notion of observation.

We can first see how parallel disjunction increases the expressive power of the calculus on a simple example. Consider the following two terms: $M = !\Omega_!$ and $N = !(\Omega_! \oplus I)$. We will see later that these two terms are the simplest example of non-trivialization in $\Lambda^{!}_{\oplus}$: indeed $\delta^c_{!(\tau \multimap \tau),!}(M,N) = \frac{1}{2}$, while $\delta^c_{!(\tau \multimap \tau),!,\|}(M,N) = 1$. In $\Lambda^{!,\|}_{\oplus}$, we are able to define a family of contexts $(C_n)_{n \in \mathbb{N}}$ as follows:

$$C_n = (\lambda !x.\ ([x \parallel\ ([x \parallel \ldots] \rightarrowtail I)\,] \rightarrowtail I)\,)\,[\cdot].$$

Essentially, C_n makes n copies of its argument, and then converges towards I if *at least* one of these copies itself converges. When we apply the context C_n to M and N, we can see that the convergence probability of $C_n[M]$ is always 0 independently of n, whereas the convergence probability of $C_n[N]$ tends towards 1 when n tends to infinity.

Theorem 2. *$\delta^c_{!,\|}$ trivializes.*

The proof is based on the construction of amplification contexts $C^{\alpha,\beta,\gamma}_\sigma$. If $\max(\alpha,\beta)=1$, we can extend the informal argument from Section 2, by taking contexts that copy an arbitrary number of times their argument. If $\min(\alpha,\beta)=0$, we can use the same idea as in the example above, by taking contexts that do an arbitrary number of disjunctions. What remains to be done to obtain the trivialization result is treating the case in which $0<\alpha,\beta<1$. The overall idea is to somehow mix the contexts we use in the previous cases. More precisely, we define a family of contexts $(C^m_n)_{n,m\in\mathbb{N}}$ as follows:

$$C^m_n = \lambda !y.\left(\bigwedge\nolimits^n(\bigvee\nolimits^m(y,\ldots,y),\ldots,\bigvee\nolimits^m(y,\ldots,y))\right)[\cdot]$$

where

$$\bigvee\nolimits^n(M_1,\ldots M_n) = ([M_1 \parallel ([M_2 \parallel \ldots] \rightarrowtail I)] \rightarrowtail I)\ ;$$
$$\bigwedge\nolimits^n(M_1\ldots M_n) = (\lambda z_1.\lambda z_2.\ldots.\lambda y.(yz_1\ldots z_n))M_1\ldots M_n.$$

The term $\bigvee^n(M_1,\ldots,M_n)$ behaves as a n-ary disjunction: it terminates if *at least one* of the M_i terminates. On the other hand, $\bigwedge^n(M_1,\ldots,M_n)$ can be seen as a n-ary conjunction: it terminates if *all* the M_i terminates. The contexts C^α_n compute m-ary conjunctions of n-ary disjunctions. Now, let ι be such that $\alpha<\iota<\beta$. We need to show that for every n, we can choose $m(n,\iota)\in\mathbb{N}$ such that:

$$\lim_{n\to\infty}\mathrm{Obs}(C^{m(n,\iota)}_n[!M]) = \begin{cases}1 \text{ if } \mathrm{Obs}(M)>\iota;\\ 0 \text{ if } \mathrm{Obs}(M)<\iota.\end{cases}\quad .$$

We can express this problem purely in terms of functional analysis, by observing that $\mathrm{Obs}(C^m_n[!M]) = (1-(1-\mathrm{Obs}(M))^m)^n$. Then the result is proved by applying the dominated convergence theorem to a well-chosen sequence of functions.

5 Tuples and Full Abstraction

This section is structured as follows: first, we define a labeled Markov chain (LMC) which expresses the semantics of our calculus in an interactive way, and then we use it to give a coinductively-defined notion of distance on a labeled transition system (LTS) of *distributions*, which coincides with the context distance defined in Section 3.2. We are not considering parallel disjunction here: the motivations for that should be clear from Theorem 2.

5.1 A Labeled Markov Chain over Tuples

Labeled Markov chains are the probabilistic analogues to labeled transition systems. Formally, a LMC is a triple $\mathscr{M}=(\mathcal{S},\mathcal{A},\mathcal{P})$, where $\mathcal{S}$ is a countable set of

states, $\mathcal{A}$ is a countable set of *labels*, and $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \mathrm{Distr}(\mathcal{S})$ is a *transition probability matrix* (where $\mathrm{Distr}(X)$ is the set of all distributions over X).

Following [9], the interactive behavior of probabilistic λ-terms can be represented by a LMC whose states are the terms of the language, whose actions are values, and where performing the action V starting from a state M corresponds to applying the value V to M. This approach is bound *not* to work in presence of pairs when metrics take the place of equivalences, due to the unsoundness of projective actions. In [6], this observation led us to introduce a new LMC whose states are *tuples* of terms, and whose actions include one *splitting* a pair: $\mathcal{P}([\langle M, N\rangle])(\text{destruct}) = \{[M, N]^1\}$. This turns out to work well in an affine setting [6]. We are going to define a LMC $\mathscr{M}^!_\oplus = (\mathcal{S}_{\mathscr{M}^!_\oplus}, \mathcal{A}_{\mathscr{M}^!_\oplus}, \mathcal{P}_{\mathscr{M}^!_\oplus})$ which is an extension of the one from [6], and which is adapted to a language with copying capabilities. The idea is to treat exponentials in the spirit of Milner's Law: $!A \multimap A \otimes !A$.

States *Tuples* are pairs of the form $K = (\boldsymbol{M}, \boldsymbol{V})$ where $\boldsymbol{M}$ and $\boldsymbol{V}$ are a sequence of terms and values, respectively. The set of all such tuples is indicated as $\mathscr{U}$. The first component of a tuple is called its *exponential part*, while the second one is called its *linear part*. We write $\vdash (\boldsymbol{M}, \boldsymbol{V}) : (\boldsymbol{\sigma}, \boldsymbol{\tau})$ if $\vdash \boldsymbol{M} : \boldsymbol{\sigma}$ and $\vdash \boldsymbol{V} : \boldsymbol{\tau}$. We note $\mathbf{T}$ the set of pairs $A = (\boldsymbol{\sigma}, \boldsymbol{\tau})$, and we call *tuple types* the elements of $\mathbf{T}$. Moreover, we say that $(\boldsymbol{\sigma}, \boldsymbol{\tau})$ is a (n, m) *tuple type* if $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ are, respectively, an n-sequence and an m-sequence. To any term M, we associate a tuple in a natural way: we note $\dot{M}$ the tuple $([\,], [M])$, and similarly if σ is a type, we indicate the tuple type $([\,], [\sigma])$ as $\dot{\sigma}$. Please observe that if $\vdash M : \sigma$, then it holds that $\vdash \dot{M} : \dot{\sigma}$.

A sequence of the form $(E, F, \boldsymbol{\sigma}, \boldsymbol{\tau}, M, \gamma)$ is said to be an *applicative typing judgment* when $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ are, respectively, an n-sequence and an m-sequence of types, E and F are respectively an n-set and an m-set, and moreover it holds that $!\boldsymbol{x}_E : !\boldsymbol{\sigma}_E, \boldsymbol{y}_F : \boldsymbol{\tau}_F \vdash M : \gamma$. Intuitively, this means that if we have a tuple $K = (\boldsymbol{N}, \boldsymbol{V})$ of type $(\boldsymbol{\sigma}, \boldsymbol{\tau})$, we can replace free variables of M by *some* of the terms from K. More precisely, we can replace variables in linear position by the V_i with $i \in F$, and variables in non linear position by N_j, with $j \in E$. We note as $M[K]$ the closed term of type γ that we obtain this way. We note $\mathscr{J}$ the set of all applicative typing judgments. We are specially interested in those judgments $(E, F, \boldsymbol{\sigma}, \boldsymbol{\tau}, M, \gamma)$ in $\mathscr{J}$ such that for every tuple K, the resulting term $M[K]$ is a *value*: that is when either $M = y_i$ for a $i \in \mathbb{N}$, or M is of the form $\lambda z.N$, $\lambda !z.N$, or $!N$. We note $\mathscr{J}^{\mathscr{V}}$ the set of those judgments.

We are now in a position to define $\mathscr{M}^!_\oplus$ formally. The set of its states is indeed defined as $\mathcal{S}_{\mathscr{M}^!_\oplus} = \{(K, A) \mid K \in \mathscr{U},\ A \in \mathbf{T},\ \vdash K : A\}$.

Labels and Transitions How do states in $\mathcal{S}_{\mathscr{M}^!_\oplus}$ interact with the environment? This is captured by the labels in $\mathcal{A}_{\mathscr{M}^!_\oplus}$, and the associated probability matrix. We define $\mathcal{A}_{\mathscr{M}^!_\oplus}$ as the disjoint union of $\mathcal{A}_?$, $\mathcal{A}_!$ and $\mathcal{A}_@$, where:

$$\mathcal{A}_! = \mathcal{A}_? = \{i \mid i \in \mathbb{N}\}; \qquad \mathcal{A}_@ = \{(\kappa, i) \mid i \in \mathbb{N},\ \kappa \in \mathscr{J}^{\mathscr{V}}\}.$$

In order to distinguish actions in $\mathcal{A}_!$ and $\mathcal{A}_?$, we write the action $i \in \mathbb{N}$ as $(?^i)$ if it comes from $\mathcal{A}_?$, and as $(!^i)$ if it comes from $\mathcal{A}_!$. The action $(\kappa, i) \in \mathcal{A}_@$ is often indicated as $@^i_\kappa$. The probability matrix $\mathcal{P}_{\mathscr{M}^!_\oplus}$ is defined formally in Figure 4. We give below some intuitions about it. The general idea is that $\mathscr{M}^!_\oplus$ is designed to express every possible effect that a context can have on tuples. $\mathcal{A}_?$ and $\mathcal{A}_!$ are designed to model copying capabilities, while $\mathcal{A}_@$ corresponds to applicative interactions.

Actions in $\mathcal{A}_?$ take any term of the form $!M$ from the linear part of the underlying tuple, unbox it and transfer M to the exponential part of the tuple. Please observe that this action is in fact deterministic: the resulting tuple is uniquely determined. Labels in $\mathcal{A}_!$, on the other hand, model the act of *copying* terms in the exponential part. We call its elements *Milner's actions*. More specifically, the action $(!^i)$ takes the i-th term in the exponential part of the tuple, makes a copy of it, evaluates it and adds the result to the linear part. Please observe that, contrary to $(?^i)$, this action can have a probabilistic outcome: the transferred term is evaluated.

Labels in $\mathcal{A}_@$ are analogues of the applicative actions from applicative bisimulation over terms (see, e.g. [9]). As such, they model environments passing arguments to programs. Here, we have to adapt this idea to our tuple framework: indeed, we can see the tuple as a collection of programs available to the environment, who is free to choose *with which* of the programs to interact with by passing it an argument. This argument, however, could depend on other components of the tuple, which need to be removed from it if lying in its linear part. Finally, please observe that all this should respect types. Labels in $\mathcal{A}_@$ are indeed defined as a pair of an index i corresponding to the position in the tuple of the term the environment chooses, and an applicative typing judgment, used to specify the argument. Please observe that in the definition of the probability matrix for applicative actions, in Figure 4, the condition on i implies that the i-th linear component of the tuple is not used to construct the argument term.

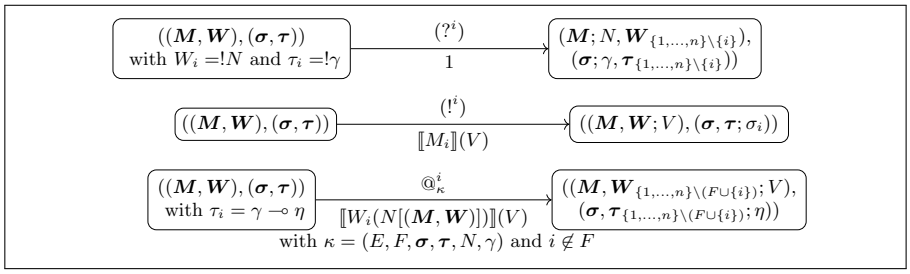

Fig. 4. Definition of $\mathcal{P}_{\mathscr{M}^!_\oplus}$

Example 4. We give in Figure 5 a fragment of $\mathscr{M}^!_\oplus$ illustrating our definitions on an example. Let τ be an element of $\mathscr{A}$. We consider terms of the form $M_\varepsilon = !(\Omega_! \oplus^\varepsilon I)$, for $\varepsilon \in \mathbb{D}$ and we look at *some* of the possible evolutions in $\mathscr{M}^!_\oplus$ from the associated state $(\dot{M}_\varepsilon, !(\tau \dot{\multimap} \tau)) = ([\,], [M_\varepsilon]), ([\,], [!(\tau \multimap \tau)])$. In Figure 5, we denote by σ the type $\tau \multimap \tau$.

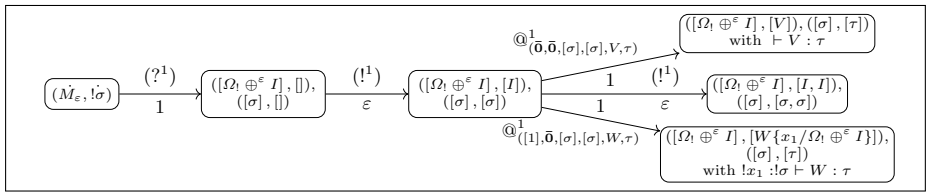

Fig. 5. A Fragment of $\mathcal{P}_{\mathscr{M}^!_\oplus}$

5.2 Distributions as States

Now that we have a LMC $\mathcal{P}_{\mathscr{M}^!_\oplus}$ modeling interaction between (tuple of) terms and their environment, we could define notions of metrics following one of the abstract definitions from the literature, e.g. by defining the *trace distance* or the *behavioral distance* between terms. This is, by the way, the approach followed in [6]. We prefer, however, to first turn $\mathcal{P}_{\mathscr{M}^!_\oplus}$ into a transition system $\mathscr{L}^!_\oplus$ whose states are *distributions* of tuples. This supports both a simple, coinductive presentation of the trace distance, but also up-to techniques, as we will see in Section 5.5 below. Both will be very convenient when evaluating the distance between concrete terms, and in particular for our running example.

It turns out that the usual notion of LTS is not sufficient for our purposes, since it lacks a way to *expose* the observables of each state, i.e., its sum. We thus adopt the following definition:

Definition 2. *A* weighted labeled transition system *(WLTS for short) is a quadruple in the form $\mathscr{L} = (\mathcal{S}, \mathcal{A}, \dot{\to}, w)$ where:*

- *$\mathcal{S}$ is a set of states and $\mathcal{A}$ is a countable set of actions,*
- *$\dot{\to}$ is a transition function such that, for every $t \in \mathcal{S}$ and $a \in \mathcal{A}$, there exists* at most *one $s \in \mathcal{S}$ such that $t \xrightarrow{a} s$,*
- *$w : \mathcal{S} \to [0, 1]$ is a function.*

Please observe how WLTSs are *deterministic* transition systems. We define the WLTS $\mathscr{L}^!_\oplus$ by the equations of Figure 6.

If $t = (\mathcal{D}, A)$ is in $\mathcal{S}_{\mathscr{L}^!_\oplus}$, we say that t is a A-state. It is easy to check that $\mathscr{L}^!_\oplus$ is nothing more than the natural way to turn $\mathcal{P}_{\mathscr{M}^!_\oplus}$ into a deterministic

$$\mathcal{S}_{\mathscr{L}^!_\oplus} = \bigcup_{A\in\mathbf{T}} (\mathrm{Distr}(\{K \mid \vdash K : A\}) \times \{A\}) \qquad \mathcal{A}_{\mathscr{L}^!_\oplus} = \mathcal{A}_{\mathscr{M}^!_\oplus} \cup \{A \mid A \in \mathbf{T}\} \qquad w(\mathcal{D}, A) = \sum_{\mathcal{D}}$$
$$(\mathcal{D}, A) \xrightarrow{A} (\mathcal{D}, A) \text{ for } A \in \mathbf{T} \qquad (\mathcal{D}, A) \xrightarrow{a} \sum_{K\in\mathscr{U}} \mathcal{D}(K) \cdot \mathcal{P}_{\mathscr{M}^!_\oplus}((K, A))(a) \text{ for } a \in \mathcal{A}_{\mathscr{M}^!_\oplus}.$$

Fig. 6. The WLTS $\mathscr{L}^!_\oplus = (\mathcal{S}_{\mathscr{L}^!_\oplus}, \mathcal{A}_{\mathscr{L}^!_\oplus}, \dot{\to}, w)$

transition system. We illustrate this idea in Figure 7, by giving a fragment of $\mathscr{L}^!_\oplus$ corresponding to (part of) the fragment of $\mathscr{M}^!_\oplus$ given in Example 4.

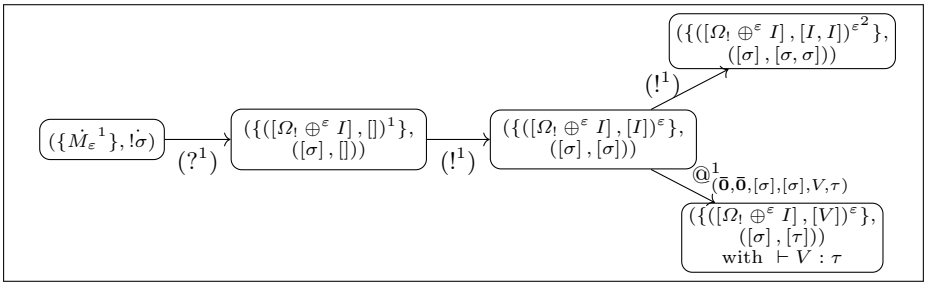

Fig. 7. A Fragment of $\mathscr{L}^!_\oplus$

5.3 A Coinductively-Defined Metric

Following Desharnais et al. [13], we use a quantitative notion of bisimulation on $\mathscr{L}^!_\oplus$ to define a distance between terms. The idea is to stipulate that, for any $\varepsilon \in [0, 1]$, a relation R is an ε-bisimulation if it is, somehow, *ε-close to a bisimulation*. The distance between two states t and s is just the smallest ε such that t and s are ε-bisimilar. However, while in [13] the notion of ε-bisimulation is *local*, we want it to be more restricted by the *global* deviation we may accept considering arbitrary sequences of actions.

Definition 3. *Let $\mathscr{L} = (\mathcal{S}, \mathcal{A}, \dot{\to}, w)$ be a WLTS. Let R be a symmetric and reflexive relation on $\mathcal{S}$, and $\varepsilon \in [0, 1]$. R is a ε-bisimulation whenever the following two conditions hold:*

- *if $t\,R\,s$, and $t \xrightarrow{a} u$, then there exists v such that $s \xrightarrow{a} v$, and it holds that $u\,R\,v$.*
- *if $t\,R\,s$, then $|w(t) - w(s)| \leq \varepsilon$.*

For every $\varepsilon \in [0, 1]$, there exists a largest ε-bisimulation, that we indicate as R^ε. Please observe that it is not an equivalence relation (since it is not transitive). We can now define a metric on $\mathcal{S}$: $\delta^b_{\mathscr{L}}(t, s) = \inf\{\varepsilon \mid t\,R^\varepsilon\,s\}$. The greatest lower bound is in fact reached as a $\delta^b_{\mathscr{L}}(t, s)$-bisimulation [7].

How can we turn $\delta^b_{\mathscr{L}}$ into a metric on *terms*? The idea is to consider the distributions on *tuples* one naturally gets when evaluating the term. To every term M of type σ, we define $\hat{s}_\sigma(M) \in \mathcal{S}_{\mathscr{L}^!_\oplus}$ as $(\{\dot{V}^{[\![M]\!](V)}\}_{V \in \mathscr{V}}, \dot{\sigma})$.

Definition 4. *For every terms M and N such that $\vdash M : \sigma$ and $\vdash N : \sigma$, we set $\delta^b_{\sigma,!}(M,N) = \delta^b_{\mathscr{L}^!_\oplus}(\hat{s}_\sigma(M), \hat{s}_\sigma(N))$.*

Example 5. Consider again the terms M_ε and M_μ from Example 2. We fix a type τ, and define $\sigma = \tau \multimap \tau$. As mentioned in Example 2, it holds that $\vdash M_\varepsilon : !\sigma$. Let now ε, μ, α be in $[0,1]$, and let R be any α-bisimulation, such that $\hat{s}_{!\sigma}(M_\varepsilon)\, R\, \hat{s}_{!\sigma}(M_\mu)$. Let $\{t_i\}_{i\in\mathbb{N}}$ and $\{s_i\}_{i\in\mathbb{N}}$ be families from $\mathcal{S}_{\mathscr{L}^!_\oplus}$ such that $\hat{s}_{!\sigma}(M_\varepsilon) \xrightarrow{(?^1)} t_0 \xrightarrow{(!^1)} \ldots \xrightarrow{(!^1)} t_i \ldots$ and $\hat{s}_{!\sigma}(M_\mu) \xrightarrow{(?^1)} s_0 \xrightarrow{(!^1)} \ldots \xrightarrow{(!^1)} s_i \ldots$. Since R is an α-bisimulation, for every i, it holds that $t_i\, R\, s_i$. Looking at the definition of $\mathscr{L}^!_\oplus$, it is easy to realize that:

$$t_i = \{([\Omega_! \oplus^\varepsilon I], [I, \ldots, I]), ([\sigma], [\sigma, \ldots, \sigma])^{\varepsilon^i}\}_{i\in\mathbb{N}};$$
$$s_i = \{([\Omega_! \oplus^\mu I], [I, \ldots, I]), ([\sigma], [\sigma, \ldots, \sigma])^{\mu^i}\}_{i\in\mathbb{N}}.$$

By the definition of an α-bisimulation, we see that this implies that $\alpha \geq |\varepsilon^i - \mu^i|$. Since this reasoning can be done for every α such that M_ε and M_μ are α-bisimilar, it means that: $\delta^b_{!\sigma,!}(M_\varepsilon, M_\mu) \geq \sup_{i\in\mathbb{N}} |\varepsilon^i - \mu^i|$. Moreover, if we consider the special case where $\varepsilon = 0$, we can actually construct a μ-bisimulation by taking

$$R = (\hat{s}_{!\sigma}(M_0), \hat{s}_{!\sigma}(M_\mu)) \cup \{(t_0, s_0)\} \cup \{((0,A),(\mathcal{D},A) \mid \sum\nolimits_{\mathcal{D}} \leq \mu\}.$$

We can easily check that R is indeed a μ-bisimulation, which tells us that $\delta^b_{!\sigma,!}(M_0, M_\mu) = \mu$.

5.4 Full Abstraction

In this section, we prove that $\delta^b_{\sigma,!}$ coincides with $\delta^c_{\sigma,!}$. We first of all show that the metric $\delta^b_{\sigma,!}$ is *sound* with respect to $\delta^c_{\sigma,!}$, i.e. that $\delta^b_{\sigma,!}$ discriminates at least as much as $\delta^c_{\sigma,!}$:

Theorem 3 (Soundness). *For any terms M and N of $\Lambda^!_\oplus$, such that $\vdash M : \sigma$ and $\vdash N : \sigma$, it holds that $\delta^c_{\sigma,!}(M,N) \leq \delta^b_{\sigma,!}(M,N)$.*

Please remember that our definition of the tuple distance is based on the notion of ε-bisimulation. Proving the soundness theorem, thus, requires us to show that for any terms M and N of type σ such that $\hat{s}_\sigma(M)$ and $\hat{s}_\sigma(N)$ are ε-bisimilar, and for any context C, it holds that $\mid \sum_{[\![C[M]]\!]} - \sum_{[\![C[N]]\!]} \mid \leq \varepsilon$. Our proof strategy is based on the fact that we can decompose every evaluation path of a term in the form $C[L]$ into *external* reduction steps (that is, steps that do *not* affect L), and *internal* reduction steps (that is, reduction steps affecting L, but which can be shown to correspond *only* to actions from $\mathscr{L}^!_\oplus$). Intuitively, if we reduce in parallel $C[M]$ and $C[N]$, we are going to have steps where only the

context is modified (and the modification does not depend on whether we are considering the first program or the second), and steps where the internal part is modified, but these steps cannot induce too much of a difference between the two programs, since the two internal terms are ε-bisimilar.

We first of all need to generalize the notion of context to deal with tuples rather than with terms. In particular, we need contexts with multiple holes having types which match those of the tuple (or, more precisely, the A-state) they are meant to be paired with. More formally:

Definition 5 (Tuple Contexts). Tuple contexts *are triples of the form* (C, A, γ)*, where* C *is an open term,* $A = (\boldsymbol{\sigma}, \boldsymbol{\tau})$ *is a* (n, m)*-tuple type, and* γ *is a type such that* $!\boldsymbol{x}_{\{1,\ldots,n\}} :!\boldsymbol{\sigma}, \boldsymbol{y}_{\{1,\ldots,m\}} : \boldsymbol{\tau} \vdash C : \gamma$*. We note* $\mathscr{C}^{\mathbf{T}}$ *the set of tuple contexts. A tuple context* (C, A, γ) *is said to be an* open value *if* C *is of one of the following four forms:* $\lambda x.M$*,* $\lambda !x.M$*,* $!M$*,* y_i *(where* $i \in \mathbb{N}$*).*

We now want to define *when* a tuple context and an A-state can be paired together, and the operational semantics of such an object, which will be derived from that of $\Lambda^{!}_{\oplus}$-terms. This is the purpose of the following definition:

Definition 6 (Tuple Context Pairs). *We say that a pair* $u = (C, t)$ *is a* tuple context pair *iff* $t = (A, \mathcal{D})$ *is an* A*-state, and* $\exists \gamma \in \mathscr{A}$*,* $(C, A, \gamma) \in \mathscr{C}^{\mathbf{T}}$*. We indicate as* $\mathbf{C} \times \mathbf{\Delta}(\mathscr{U})$ *the set of tuple context pairs. Moreover, given such a pair* $u = (C, (A, \mathcal{D}))$*, we define* $\boldsymbol{F}(u)$ *as the (potentially infinite) distribution over* $\mathscr{T}$ *given by:*

$$\boldsymbol{F}(u) = \{C\{\boldsymbol{x}/\boldsymbol{M}\}\{\boldsymbol{y}/\boldsymbol{N}\}^{\mathcal{D}(\boldsymbol{M},\boldsymbol{N})}\}_{(\boldsymbol{M},\boldsymbol{N}) \in \mathcal{S}(\mathcal{D})}.$$

Giving a notion of context distance for A-states is now quite easy and natural, since we know how contexts for such objects look like. For the sake of being as general as possible, this notion of a distance is parametric on a set of tuple contexts $\mathscr{C} \subseteq \mathscr{C}^{\mathbf{T}}$.

Definition 7. *Let* $\mathscr{C} \subseteq \mathscr{C}^{\mathbf{T}}$*,* $A \in \mathbf{T}$*, and* t, s *be two* A*-states. We define:*

$$\delta^{c}_{\mathscr{C}}(t, s) = \sup_{(C,A,\sigma) \in \mathscr{C}} \left| \sum\nolimits_{[\![\boldsymbol{F}(C,t)]\!]} - \sum\nolimits_{[\![\boldsymbol{F}(C,s)]\!]} \right|$$

Unsurprisingly, the context distance between terms equals $\delta^{c}_{\mathscr{C}^{\mathbf{T}}}$ when applied to A-states obtained through $\hat{s}_{\sigma}(\cdot)$:

Proposition 1. *If* $\vdash M, N : \sigma$*, then* $\delta^{c}_{\sigma,!}(M, N) = \delta^{c}_{\mathscr{C}^{\mathbf{T}}}(\hat{s}_{\sigma}(M), \hat{s}_{\sigma}(N))$*.*

But why did we introduce $\mathbf{C} \times \mathbf{\Delta}(\mathscr{U})$? Actually, these pairs allow a fine analysis of how tuples behave when put in a context, which in turn is precisely what we need to prove Theorem 3. This analysis, however, is not possible without endowing $\mathbf{C} \times \mathbf{\Delta}(\mathscr{U})$ itself with an operational semantics, which is precisely what we are going to do in the next paragraphs.

Two relations need to be defined. On the one hand, we need a one-step *labeled* transition relation $\rightarrow_{\mathbf{C} \times \mathbf{\Delta}(\mathscr{U})}$ which turns an element of $\mathbf{C} \times \mathbf{\Delta}(\mathscr{U})$ into

a distribution over $\mathbf{C} \times \mathbf{\Delta}(\mathscr{U})$ by performing an action. Intuitively, one step of reduction in $\rightarrow_{\mathbf{C}\times\mathbf{\Delta}(\mathscr{U})}$ corresponds to *at most* one step of reduction in $\mathscr{L}^!_\oplus$. If that step exists, (i.e. if the *term* is reduced) then the label is the same, and otherwise (i.e., if only the *context* is reduced), the label is just τ. We also need a multi-step approximation semantics $\Rightarrow_{\mathbf{C}\times\mathbf{\Delta}(\mathscr{U})}$ between elements of $\mathbf{C} \times \mathbf{\Delta}(\mathscr{U})$ and subdistributions over the same set. The latter is based on the former, and both are formally defined in Figure 8, where

- E is an evaluation context;
- t is an (n, m)-state from $\mathcal{S}_{\mathscr{L}^!_\oplus}$;
- h is a tuple-context pair from $\mathbf{C} \times \mathbf{\Delta}(\mathscr{U})$;
- For every context C, $C_{\mathrm{remove}(E)}$ stands for the context

$$C\{y_1/y_{1-\#\{j|j\in E\wedge j<1\}}\}\ldots\{y_n/y_{n-\#\{j|j\in E\wedge j<n\}}\}$$

$$\frac{}{(E[(\lambda z.N)M],t)\xrightarrow{\tau}_{\mathbf{C}\times\mathbf{\Delta}(\mathscr{U})}\{(E[N\{z/M\}],t)^1\}} \qquad \frac{}{(E[(\lambda !z.N)!M],t)\xrightarrow{\tau}_{\mathbf{C}\times\mathbf{\Delta}(\mathscr{U})}\{(E[N\{z/M\}],t)^1\}}$$

$$\frac{}{(E[M\oplus N],t)\xrightarrow{\tau}_{\mathbf{C}\times\mathbf{\Delta}(\mathscr{U})}\{(E[M],t)^{1/2}\}+\{(E[N],t)^{1/2}\}}$$

$$\frac{t\xrightarrow{(!j)}s}{(E[x_j],t)\xrightarrow{(!j)}_{\mathbf{C}\times\mathbf{\Delta}(\mathscr{U})}\{(E[y_{m+1}],s)^1\}} \qquad \frac{t\xrightarrow{(?j)}s \qquad C=E[(\lambda !z.N)!x_{n+1}]_{\mathrm{remove}(\{j\})}}{(E[(\lambda !z.N)y_j],t)\xrightarrow{(?j)}_{\mathbf{C}\times\mathbf{\Delta}(\mathscr{U})}\{(C,s)^1\}}$$

$$\frac{t\xrightarrow{@^j_\kappa}s \qquad \begin{array}{c} t=(\mathcal{D},A)\wedge A=\boldsymbol{\sigma},\boldsymbol{\tau} \\ \boldsymbol{\tau}_j=\eta\multimap\iota \end{array} \qquad \begin{array}{c}\kappa=(\{1,\ldots,n\},F,\boldsymbol{\sigma},\boldsymbol{\tau},V,\eta)\in\mathscr{J}^{\mathscr{V}} \\ j\notin F\end{array}}{(E[y_jV],t)\xrightarrow{@^j_\kappa}_{\mathbf{C}\times\mathbf{\Delta}(\mathscr{U})}\{((E[y_{m+1}])_{\mathrm{remove}(F\cup\{j\})},s)^1\}}$$

$$\frac{h \text{ in normal form for } \dot{\rightarrow}_{\mathbf{C}\times\mathbf{\Delta}(\mathscr{U})}}{h\Rightarrow_{\mathbf{C}\times\mathbf{\Delta}(\mathscr{U})}\{t^1\}} \qquad \frac{}{h\Rightarrow_{\mathbf{C}\times\mathbf{\Delta}(\mathscr{U})}0} \qquad \frac{h\xrightarrow{a}_{\mathbf{C}\times\mathbf{\Delta}(\mathscr{U})}\mathcal{D} \qquad k\Rightarrow_{\mathbf{C}\times\mathbf{\Delta}(\mathscr{U})}\mathcal{E}_k}{h\Rightarrow_{\mathbf{C}\times\mathbf{\Delta}(\mathscr{U})}\sum_{k\in\mathsf{S}(\mathcal{D})}\mathcal{D}(k)\cdot\mathcal{E}_k}$$

Fig. 8. Rules for $\Rightarrow_{\mathbf{C}\times\mathbf{\Delta}(\mathscr{U})}$

We first show that this definition can indeed be related to the usual semantics for terms. This takes the form of the following lemma:

Lemma 2. *Let be $u \in \mathbf{C} \times \mathbf{\Delta}(\mathscr{U})$. Then:*

- $\{\mathcal{D} \mid u\Rightarrow_{\mathbf{C}\times\mathbf{\Delta}(\mathscr{U})}\mathcal{D}\}$ *is a directed set. We define* $[\![u]\!]^{\mathbf{C}\times\mathbf{\Delta}(\mathscr{U})}$ *as its least upper bound;*
- $\boldsymbol{F}(\cdot) : \mathit{Distr}(\mathbf{C} \times \mathbf{\Delta}(\mathscr{U})) \to \mathit{Distr}(\mathscr{T})$ *is continuous;*
- $[\![\boldsymbol{F}(u)]\!] = \boldsymbol{F}([\![u]\!]^{\mathbf{C}\times\mathbf{\Delta}(\mathscr{U})})$.

Before proceeding, we need to understand how any reflexive and symmetric relation on $\mathbf{C} \times \mathbf{\Delta}(\mathscr{U})$ can be turned into a relation on *distributions* on $\mathbf{C} \times \mathbf{\Delta}(\mathscr{U})$. If R is a reflexive and symmetric relation on $\mathbf{C} \times \mathbf{\Delta}(\mathscr{U})$, we lift it to distributions over $\mathbf{C} \times \mathbf{\Delta}(\mathscr{U})$ by stipulating that $\mathcal{D}\ R\ \mathcal{E}$ whenever there exists a countable set I, a family $(p_i)_{i \in I}$ of positive reals of sum smaller than 1, and families $(h_i)_{i \in I}, (k_i)_{i \in I}$ in $\mathbf{C} \times \mathbf{\Delta}(\mathscr{U})$, such that $\mathcal{D} = \{{h_i}^{p_i}\}_{i \in I}$, $\mathcal{E} = \{{k_i}^{p_i}\}_{i \in I}$, and moreover $h_i\ R\ k_i$ for every $i \in I$.

We now want to precisely capture *when* a relation on $\mathbf{C} \times \mathbf{\Delta}(\mathscr{U})$ can be used to evaluate the distance between tuple-context pairs.

Definition 8. *Let R be a reflexive and symmetric relation on $\mathbf{C} \times \mathbf{\Delta}(\mathscr{U})$.*

- *We say that R is* preserved by $\rightarrow_{\mathbf{C} \times \mathbf{\Delta}(\mathscr{U})}$ *if, for any $h, k \in \mathbf{C} \times \mathbf{\Delta}(\mathscr{U})$ such that $h\ R\ k$, if $h \xrightarrow{a}_{\mathbf{C} \times \mathbf{\Delta}(\mathscr{U})} \mathcal{D}$, then there exists $\mathcal{E}$ such that $k \xrightarrow{a}_{\mathbf{C} \times \mathbf{\Delta}(\mathscr{U})} \mathcal{E}$, and that $\mathcal{D}\ R\ \mathcal{E}$.*
- *We say that R is* ε-bounding *if $h\ R\ k$ implies $|\sum_{\boldsymbol{F}(h)} - \sum_{\boldsymbol{F}(k)}| \leq \varepsilon$.*
- *Let $\mathscr{C}$ be a set of tuple contexts, and $t, s \in \mathcal{S}_{\mathscr{L}^!_\oplus}$ be two A-states. We say that R is* $\mathscr{C}$-closed with respect to t and s *if, for every C and γ such that $(C, A, \gamma) \in \mathscr{C}$, it holds that $(C, t)\ R\ (C, s)$.*

Please observe how any relation preserving $\rightarrow_{\mathbf{C} \times \mathbf{\Delta}(\mathscr{U})}$ and being ε-bounding can be seen somehow as an ε-bisimulation, but on tuple-context pairs. The way we defined the lifting, however, makes it even a *stronger* notion, i.e., the ideal candidate for an intermediate step towards Soundness.

As a first step, the conditions from Definition 8 are enough to guarantee that two terms are at context distance at most ε.

Proposition 2. *Let M, N be two terms of type σ. Suppose there exists a reflexive and symmetric relation R on $\mathbf{C} \times \mathbf{\Delta}(\mathscr{U})$, which is preserved by $\rightarrow_{\mathbf{C} \times \mathbf{\Delta}(\mathscr{U})}$, ε-bounding, and $\mathscr{C}^{\mathbf{T}}$-closed with respect to $\hat{s}_\sigma(M)$ and $\hat{s}_\sigma(N)$. Then $\delta^c_{\sigma,!}(M, N) \leq \varepsilon$.*

What remains to be done, then, is to show that if two terms are related by R^ε, then they themselves satisfy Definition 8. Compulsory to that is showing that any ε-bisimulation can at least be turned into a relation on $\mathbf{C} \times \mathbf{\Delta}(\mathscr{U})$. We need to do that, in particular, in a way guaranteeing the $\mathscr{C}$-closure of the resulting relation, and thus considering all possible tuple contexts from $\mathscr{C}$:

Definition 9. *Let R be a reflexive and symmetric relation on $\mathcal{S}_{\mathscr{L}^!_\oplus}$. Let be $\mathscr{C}$ a set of tuple contexts. We define its* contextual lifting to $\mathbf{C} \times \mathbf{\Delta}(\mathscr{U})$ with respect to $\mathscr{C}$ *as the following binary relation on $\mathbf{C} \times \mathbf{\Delta}(\mathscr{U})$:*

$$\widehat{R}^{\mathscr{C}}_A = \bigcup_{(C,A,\sigma) \in \mathscr{C}} \{((C, t), (C, s)) \mid t, s\ A\text{-states}, t\ R\ s\}; \qquad \widehat{R}^{\mathscr{C}} = \bigcup_{A \in \mathbf{T}} \widehat{R}^{\mathscr{C}}_A.$$

The following result tells us that, indeed, any ε-bisimulation can be turned into a relation satisfying Definition 8:

Proposition 3. *Let R be an ε-bisimulation. Then $\widehat{R}^{\mathscr{C}^{\mathbf{T}}}$ is preserved by $\rightarrow_{\mathbf{C} \times \mathbf{\Delta}(\mathscr{U})}$ and ε-bounding, and $\mathscr{C}^{\mathbf{T}}$-closed with respect to every t, s such that $t\ R\ s$.*

We are finally ready to give a proof of soundness:

Proof (of Theorem 3). Consider two terms M and N of type σ. Let ε be $\delta^b_{\sigma,!}(M,N)$. We take R^ε (defined in Definition 3 as the largest ε-bisimulation), and we see that $\hat{s}_\sigma(M)\, R^\varepsilon\, \hat{s}_\sigma(N)$. Proposition 3 tells us that we can apply Proposition 2 to M, N, and $\widehat{(R^\varepsilon)}^{\mathscr{C}^{\mathbf{T}}}$. Doing so we obtain that $\delta^c_{\sigma,!}(M,N) \leq \varepsilon$, which is the thesis. □

We can actually show (see [7]) that $\delta^b_{\sigma,!}$ is also complete with respect to the contextual distance:

Theorem 4 (Full Abstraction). *For every σ, $\delta^c_{\sigma,!} = \delta^b_{\sigma,!}$.*

5.5 On an Up-to-Context Technique

As we have just shown, context distance can be characterized as a coinductively-defined metric, which turns out to be useful when evaluating the distance between terms. In this section, we will go even further, and show how an *up-to-context* [31] notion of ε-bisimulation is precisely what we need to handle our running example.

We first of all need to generalize our definition of a tuple: an *open tuple* is a pair $(\boldsymbol{M}, \boldsymbol{N})$, where $\boldsymbol{M}$ and $\boldsymbol{N}$ are sequences of (not necessarily closed) typable terms.

Definition 10. *If $K = (\boldsymbol{M}, \boldsymbol{N})$ is an open tuple, and $A = (\boldsymbol{\gamma}, \boldsymbol{\eta})$ is a tuple type, we say that $(\boldsymbol{\sigma}, \boldsymbol{\tau}, K, A)$ is a* substitution judgment *iff:*

- $!\boldsymbol{x} : !\boldsymbol{\sigma} \vdash \boldsymbol{M}_i : \boldsymbol{\gamma}_i$*;*
- *if n and m are such that $\boldsymbol{\tau}$ is a n-sequence, and $\boldsymbol{N}$ a m-sequence, then there exists a partition $\{E_1, \ldots E_m\}$ of $\{1, \ldots, n\}$ such that $\boldsymbol{y}_{E_j} : \boldsymbol{\tau}_{E_j} \vdash N_j : \eta_j$ for every $j \in \{1, \ldots, m\}$.*

$\mathscr{J}^{subst}$ is the set of all substitution judgments.

If $\kappa = (\boldsymbol{\sigma}, \boldsymbol{\tau}, K, A) \in \mathscr{J}^{\text{subst}}$, and $H \in \mathscr{U}$ is of type $(\boldsymbol{\sigma}, \boldsymbol{\tau})$, then there is a natural way to form a tuple $\kappa[H]$, namely by substituting the free variables of K by the components of H. In the following, we restrict $\mathscr{J}^{\text{subst}}$ to those judgments κ such that for every H, terms in the linear part of $\kappa[H]$ are values. Observe that we always have $\vdash \kappa[H] : A$. We extend the notation $\kappa[H]$ to distributions over $\mathscr{U}$: if $\mathcal{D}$ is a distribution over tuples of type $(\boldsymbol{\sigma}, \boldsymbol{\tau})$, we note $\kappa[\mathcal{D}] = \{\kappa[H]^{\mathcal{D}(H)}\}_{H \in \mathscr{U}}$, which is a distribution over tuples of type A. Moreover, we want to be able to apply our substitution judgments to the states of $\mathscr{L}^!_\oplus$. If $t = (\mathcal{D}, (\boldsymbol{\sigma}, \boldsymbol{\tau})) \in \mathcal{S}_{\mathscr{L}^!_\oplus}$, and $\kappa = (\boldsymbol{\sigma}, \boldsymbol{\tau}, K, A)$, the state of $\mathscr{L}^!_\oplus$ defined by $(\kappa[\mathcal{D}], A)$ will be often indicated as $\kappa[t]$.

Example 6. We illustrate on a simple example the use of substitution judgments. Let be τ any type. Consider $\boldsymbol{\sigma} = [\tau \multimap \tau]$, and $\boldsymbol{\tau} = []$. Moreover, let $K = ([x_1], [I])$ and $A = ([\tau \multimap \tau], [\tau \multimap \tau])$. Then $\kappa = (\boldsymbol{\sigma}, \boldsymbol{\tau}, K, A)$ is a substitution judgment. We consider now a tuple of type $(\boldsymbol{\sigma}, \boldsymbol{\tau})$. In fact, we take here a tuple

that will be useful in order to analyze our running example: $H = ([\Omega_! \oplus^\varepsilon I], [])$. By substituting H in κ, we obtain $\kappa[H] = ([\Omega_! \oplus^\varepsilon I], [I])$, and we can see easily that we obtain indeed a tuple of type A.

The main idea behind up-to-context bisimulation is to allow for the freedom of discarding any context when proving a relation to be a bisimulation. This is captured by the following definition:

Definition 11. *Let R be a relation on $\mathcal{S}_{\mathscr{L}^!_\oplus}$. R is an* ε-bisimulation up-to-context *if for every t and s such that $t\ R\ s$, the following holds:*

- *there exists $C \in \mathbf{T}$ such that $t = (\mathcal{D}, C)$, $s = (\mathcal{E}, C)$, and $|\sum_{\mathcal{D}} - \sum_{\mathcal{E}}| \leq \varepsilon$.*
- *for any $a \in \mathcal{A}_{\mathscr{M}^!_\oplus}$, if $t \xrightarrow{a} u = (\mathcal{D}, A)$ and $s \xrightarrow{a} v = (\mathcal{E}, A)$, then there exists a finite set $I \subseteq \mathbb{N}$ such that:*
  - *there is a family of rationals $(p_i)_{i\in I}$ such that $\sum_{i\in I} p_i \leq 1$;*
  - *there are families $\boldsymbol{\sigma}^i$, $\boldsymbol{\tau}^i$, and K^i, such that $\kappa_i = (\boldsymbol{\sigma}^i, \boldsymbol{\tau}^i, K^i, A)$ is a substitution judgment for every $i \in I$;*
  - *there are distributions over tuples $\mathcal{D}_i$, $\mathcal{E}_i$ such that $(\mathcal{D}_i, B_i)\ R\ (\mathcal{E}_i, B_i)$;*

  and moreover $\mathcal{D} = \sum_{i\in I} p_i \cdot \kappa_i[\mathcal{D}_i]$, and $\mathcal{E} = \sum_{i\in I} p_i \cdot \kappa_i[\mathcal{E}_i]$.

The proof method we just introduced is indeed quite useful when handling our running example.

Example 7. We show that up-to bisimulations can handle our running example. Please recall the definition of M_ε given in Example 2. First, we can see that, for every a, for every type τ, $\hat{s}_{!(\tau\multimap\tau)}(M_a) = (\{([], [!\Omega_! \oplus^a I])^1\}, ([], [!\tau \multimap \tau]))$. We define a relation R on $\mathcal{S}_{\mathscr{L}^!_\oplus}$ containing $(\hat{s}_{!(\tau\multimap\tau)}(M_\varepsilon), \hat{s}_{!(\tau\multimap\tau)}(M_\mu))$, and we show that it is a γ-bisimulation up-to-context for an appropriate γ. In order to simplify the notations, we define $B = ([\tau \multimap \tau], [])$, and $t_n, s_n \in \mathcal{S}_{\mathscr{L}^!_\oplus}$ as:

$$t_n = (\{([(\Omega_! \oplus^\varepsilon I)], [])^{(\varepsilon^n)}\}, B), \qquad s_n = (\{([(\Omega_! \oplus^\mu I)], [])^{(\mu^n)}\}, B).$$

Then, we define the relation R as $R = \{(\hat{s}_\sigma(M), \hat{s}_\sigma(N))\} \cup \{(t_n, s_n) \mid n \in \mathbb{N}\}$. One can check that R is indeed a γ-bisimulation up-to-context (where $\gamma = \sup_{n\in\mathbb{N}}|\varepsilon^n - \mu^n|$) by carefully analyzing [7] every possible action. The proof is based on the following observations:

- The only action starting from $\hat{s}_\sigma(M)$ or $\hat{s}_\sigma(N)$ is $a = (?^1)$, passing a term to the exponential part of the tuple, then we end up in t_0 and s_0 respectively.
- If we start from t_n or s_n, the only relevant action is Milner's action $a = (!^1)$, consisting in taking a copy of the term in the exponential part, evaluating it, and putting the result in the linear part. We can see (using the substitution judgment κ defined in Example 6), that $t_n \xrightarrow{a} \kappa[t_{n+1}]$, and similarly $s_n \xrightarrow{a} \kappa[s_{n+1}]$, and the result follows.

Bisimulations up-to-context would be useless without a correctness result such as the following one:

Theorem 5. *If R is an ε-bisimulation up-to context, then $R \subseteq R^\varepsilon$.*

The proof is an extension of that of Theorem 3 (although technically more involved), and can be found in [7].

Example 8. We can exploit the soundness of up-to-context bisimulation to obtain the contextual distance for our running example. This allows us to conclude that $\delta^c_{!(\tau\multimap\tau),!}(M_\varepsilon, M_\mu) = \sup_{n\in\mathbb{N}}|\varepsilon^n - \mu^n|$. The context distance between M_ε and M_μ is thus *strictly* between 0 and 1 whenever $0 < |\varepsilon - \mu| < 1$.

6 Probabilistic λ-Calculi, in Perspective

The calculus $\Lambda^{!,\|}_\oplus$ we analyzed in this paper is, at least apparently, nonstandard, given the presence of parallel disjunction, but also because of the linear refinement it is based on. In this section, we will reconcile what we have done so far with calculi in the literature, and in particular with untyped probabilistic λ-calculi akin to those studied, e.g., in [9, 5].

We consider a language $\Lambda_\oplus$ defined by the following grammar:

$$M \in \Lambda_\oplus ::= x \;\Big|\; MM \;\Big|\; \lambda x.M \;\Big|\; M \oplus M.$$

6.1 On Stable Fragments of $\mathscr{M}^!_\oplus$.

Our objective in this section is to characterize various notions of context distance for $\Lambda_\oplus$ by way of appropriate embeddings into $\Lambda^!_\oplus$, and thus by the LMC $\mathscr{M}^!_\oplus$. It is quite convenient, then, to understand when any fragment of $\mathscr{M}^!_\oplus$ is sufficiently *robust* so as to be somehow self-contained:

Definition 12. *We say that the pair $(\hat{\mathcal{S}}, \hat{\mathcal{A}})$, where $\hat{\mathcal{S}} \subseteq \mathcal{S}_{\mathscr{M}^!_\oplus}$, and $\hat{\mathcal{A}} \subseteq \mathbf{T}\times\mathcal{A}_{\mathscr{M}^!_\oplus}$ is a* stable fragment of $\mathscr{M}^!_\oplus$ *iff for every pair $(A, a) \in \hat{\mathcal{A}}$, for every A-state t, and for every $s \in \mathcal{S}$ such that $\mathcal{P}_{\mathscr{M}^!_\oplus}(t, a, s) > 0$, it holds that $s \in \hat{\mathcal{S}}$.*

Using a stable fragment of $\mathscr{M}^!_\oplus$, we can restrict the WLTS $\mathscr{L}^!_\oplus$ in a meaningful way. The idea is that we only consider some of the states of $\mathscr{L}^!_\oplus$, and we are able to choose the possible actions depending on the type of the state of $\mathscr{L}^!_\oplus$ we consider.

Definition 13. *If $\mathscr{F} = (\hat{\mathcal{S}}, \hat{\mathcal{A}})$ is a stable fragment of $\mathscr{M}^!_\oplus$, we define a WLTS by $\mathscr{L}_\mathscr{F} = (\mathcal{S}_{\mathscr{L}_\mathscr{F}}, \mathcal{A}_{\mathscr{L}_\mathscr{F}}, \dot{\to}_\mathscr{F}, w_\mathscr{F})$, as*

$$\mathcal{S}_{\mathscr{L}_\mathscr{F}} = \bigcup\nolimits_{A\in\mathbf{T}} \mathit{Distr}(\{K \mid (K, A) \in \hat{\mathcal{S}}\}) \times \{A\}; \qquad \mathcal{A}_{\mathscr{L}_\mathscr{F}} = \bigcup\nolimits_{(A,a)\in\hat{\mathcal{A}}} \{a\} \cup \mathbf{T};$$

$$\dot{\to}_\mathscr{F} = \dot{\to} \cap \{((\mathcal{D}, A), a, s) \mid \mathcal{S}(\mathcal{D}) \subseteq \hat{\mathcal{S}},\ (A, a) \in \hat{\mathcal{A}}\};$$

and $w_\mathscr{F}$ is defined as expected.

We want to be able to define a notion of distance on a *fragment* of the original language $\Lambda^!_\oplus$, so that it verifies the soundness property for a *restricted* set of contexts. To do that, we need the restricted set of contexts $\mathscr{C}$ to be preserved by the stable fragment:

Definition 14. *Let $\mathscr{F} = (\hat{\mathcal{S}}, \hat{\mathcal{A}})$ be a stable fragment of $\mathscr{M}^!_\oplus$. Let $\mathscr{C}$ be a set of tuple contexts. We say that $\mathscr{C}$* is preserved by *$\mathscr{F}$ if the following holds: for any $(C, A, \gamma) \in \mathscr{C}$ that is not an open value and for any A-state t in $\mathcal{S}_{\mathscr{L}_\mathscr{F}}$, there exists a such that $(A, a) \in \hat{\mathcal{A}} \bigcup (\mathbf{T} \times \{\tau\})$, $(C, t) \xrightarrow{a}_{\mathbf{C} \times \mathbf{\Delta}(\mathscr{U})} \mathcal{E}$, and moreover:*

$$\mathcal{S}(\mathcal{E}) \subseteq \bigcup\nolimits_{B \in \mathbf{T}} \{(D, s) \mid s \text{ a } B\text{-state} \wedge \exists \eta \text{ s.t. } (D, B, \eta) \in \mathscr{C}\}$$

We are now able to provide guarantees that the contextual distance $\delta^c_\mathscr{C}$ with respect to our restricted set of contexts $\mathscr{C}$ is smaller that the distance defined on the WLTS $\mathscr{L}_\mathscr{F}$ induced by our stable fragment $\mathscr{F}$. This is the spirit of the following proposition.

Proposition 4. *Let $\mathscr{F} = (\hat{\mathcal{S}}, \hat{\mathcal{A}})$ be a stable fragment of $\mathscr{M}^!_\oplus$, $\mathscr{C}$ be a set of tuple contexts preserved by $\mathscr{F}$, and $t, s \in \mathcal{S}_{\mathscr{L}_\mathscr{F}}$. Then $\delta^c_\mathscr{C}(t, s) \leq \delta^b_{\mathscr{L}_\mathscr{F}}(t, s)$.*

In the following, we make use of Proposition 4 on stable fragments corresponding to embeddings of $\Lambda_\oplus$ into $\Lambda^!_\oplus$. We will consider two different encodings depending on the underlying notion of evaluation.

6.2 Call-by-Name

$$E ::= [\cdot] \mid EM \qquad \frac{}{M \oplus N \hookrightarrow M, N} \qquad \frac{}{(\lambda x.M)N \hookrightarrow M\{x/N\}} \qquad \frac{M \hookrightarrow N_1, \ldots, N_n}{E[M] \to E[N_1], \ldots E[N_n]}$$

Fig. 9. One-step Call-by-Name Semantics

$\Lambda_\oplus$ can first of all be endowed with call-by-name semantics, as in Figure 9. We use it to define an approximation semantics exactly in the same way as in Figure 1, and we take as usual the semantics of a term to be the least upper bound of its approximated semantics. Moreover, we denote by δ^c_{cbn} the context distance on $\Lambda_\oplus$, defined the natural way. We are going, in the remainder of this section, to use our results about $\Lambda^!_\oplus$ to obtain a characterization of δ^c_{cbn}.

The Call-By-Name Embedding Girard's translation [19] gives us an embedding $\langle\cdot\rangle^{\text{cbn}} : \Lambda_\oplus \to \Lambda^!_\oplus$, defined as follows:

$$\langle x \rangle^{\text{cbn}} = x \qquad \langle \lambda x.M \rangle^{\text{cbn}} = \lambda !x.\langle M \rangle^{\text{cbn}}$$
$$\langle MN \rangle^{\text{cbn}} = \langle M \rangle^{\text{cbn}} ! \langle N \rangle^{\text{cbn}} \qquad \langle M \oplus N \rangle^{\text{cbn}} = \langle M \rangle^{\text{cbn}} \oplus \langle N \rangle^{\text{cbn}}$$

Please observe that $\langle\cdot\rangle^{\text{cbn}}$ respects typing in the sense that, when we define $\sigma^{\text{cbn}} = \mu\alpha.!\alpha \multimap \alpha$, it holds that for every term M of $\Lambda_\oplus$ whose free variables are in $\{x_1, \ldots, x_n\}$, we can show that $!x_1 : !\sigma^{\text{cbn}}, \ldots, !x_n : !\sigma^{\text{cbn}} \vdash \langle M \rangle^{\text{cbn}} : \sigma^{\text{cbn}}$.

Metrics for $\Lambda_\oplus$ It is very tempting to define a metric on $\Lambda_\oplus$ just as follows: $\delta^b_{\mathrm{cbn}}(M,N) = \delta^b_{!_{\sigma^{\mathrm{cbn}}},!}(!\langle M\rangle^{\mathrm{cbn}}, !\langle N\rangle^{\mathrm{cbn}})$. We can easily see that it is sound with respect to the context distance for $\Lambda_\oplus$, since any context of this language can be seen, through $\langle\cdot\rangle^{\mathrm{cbn}}$, as a context in $\Lambda^!_\oplus$. However, it is not complete, as shown by the following example:

Example 9. We consider $M = \Omega \oplus (\lambda x.\Omega)$ and $N = (\lambda x.\Omega)$. We can see that $\delta^b_{!_{\sigma^{\mathrm{cbn}}},!}(!\langle M\rangle^{\mathrm{cbn}}, !\langle N\rangle^{\mathrm{cbn}}) = 1$: indeed, when we define a sequence of $\Lambda^!_\oplus$-contexts by $C_n = \lambda!x.((\lambda y_1 \ldots \lambda y_n.(\lambda z.zy_1, \ldots y_n))x \ldots x)\,[]$, we see that $\mathrm{Obs}(!\langle M\rangle^{\mathrm{cbn}}) = 1/2^n$ while $\mathrm{Obs}(!\langle N\rangle^{\mathrm{cbn}}) = 1$. But those contexts C_n have more expressive power than any context in $\langle\Lambda_\oplus\rangle^{\mathrm{cbn}}$, since they do something that none of the contexts from $\Lambda_\oplus$ can do: they evaluate a copy of the term, and then shift their focus to *another* copy of the term. It can be seen in the embedding: a term in $\langle\Lambda_\oplus\rangle^{\mathrm{cbn}}$ never has several redexes in linear position. It can actually be shown (see [7]) that $\delta^c_{\mathrm{cbn}}(M,N) = \frac{1}{2} < \delta^b_{\mathrm{cbn}}(M,N)$.

The way out consists in using the notion of stable fragment to refine the Markov Chain $\mathscr{M}^!_\oplus$ by keeping only the states and actions we are interested in.

Definition 15. *We define a stable fragment $\mathscr{F}^{cbn}$ as specified in Figure 10, and a distance δ_{cbn} on $\Lambda_\oplus$ as $\delta_{cbn}(M,N) = \delta^b_{\mathscr{L}_{\mathscr{F}^{cbn}}}(\hat{s}^{cbn}(M), \hat{s}^{cbn}(N))$, where $\hat{s}^{cbn}(M) = (\{([\langle M\rangle^{cbn}], [])^1\}, A^0)$.*

$$A^0 = ([\sigma^{\mathrm{cbn}}], []) \qquad A^1 = ([\sigma^{\mathrm{cbn}}], [\sigma^{\mathrm{cbn}}]) \qquad \hat{U}^{\mathrm{cbn}}(M) = (([\langle M\rangle^{\mathrm{cbn}}], []), A^0)$$
$$\mathcal{S}_{\mathscr{M}^{\mathrm{cbn}}_\oplus} = \left(\left\{\hat{U}^{\mathrm{cbn}}(M) \mid M \in \Lambda_\oplus\right\} \cup \left\{([\langle M\rangle^{\mathrm{cbn}}], [\langle V\rangle^{\mathrm{cbn}}]), A^1 \mid M \in \Lambda_\oplus, V \in \Lambda_\oplus \text{ a value } \right\}\right) \bigcap \mathcal{S}_{\mathscr{M}^!_\oplus}$$
$$\mathscr{J}^{\mathscr{V}}_{\mathrm{cbn}} = \mathscr{J}^{\mathscr{V}} \cap \{(E, F, \boldsymbol{\sigma}, \boldsymbol{\tau}, M, \gamma)\,,\ M \in \langle\Lambda_\oplus\rangle^{\mathrm{cbn}}\}$$
$$\mathcal{A}_{\mathscr{M}^{\mathrm{cbn}}_\oplus} = \{A^1\} \times \{@^1_\kappa \mid \kappa \in \mathscr{J}^{\mathscr{V}}_{\mathrm{cbn}}\} \cup \{A^0\} \times \mathcal{A}_!$$

Fig. 10. The Stable Fragment $\mathscr{F}^{\mathrm{cbn}} = (\mathcal{S}_{\mathscr{M}^{\mathrm{cbn}}_\oplus}, \mathcal{A}_{\mathscr{M}^{\mathrm{cbn}}_\oplus})$.

We need now to define a set of tuple contexts preserved by $\mathscr{F}^{\mathrm{cbn}}$, the aim of applying Proposition 4.

Definition 16. *$\mathscr{C}_{cbn}$ is the smallest set of tuple contexts such that:*
- *If $M \in \Lambda_\oplus$ with $FV(M) \subseteq \{x_1\}$, then $(\langle M\rangle^{cbn}, A^0, \sigma^{cbn}) \in \mathscr{C}_{cbn}$;*
- *If $(C, A^0, \sigma^{cbn}) \in \mathscr{C}_{cbn}$, and $C = E[x_1]$, it holds that $(E[y_1], A^1, \sigma^{cbn}) \in \mathscr{C}_{cbn}$.*

$\mathscr{C}_{\mathrm{cbn}}$ is designed to allow us to link δ^c_{cbn} and $\delta^c_{\mathscr{C}_{\mathrm{cbn}}}$: for any $M, N \in \Lambda_\oplus$ closed terms, it holds that $\delta^c_{\mathrm{cbn}}(M,N) = \delta^c_{\mathscr{C}_{\mathrm{cbn}}}(\hat{s}^{\mathrm{cbn}}(M), \hat{s}^{\mathrm{cbn}}(N))$. Moreover, $\mathscr{C}_{\mathrm{cbn}}$ is preserved by the stable fragment $\mathscr{F}^{\mathrm{cbn}}$ (the proof can be found in [7]).

Theorem 6 (Call-by-Name Full Abstraction). *δ^c_{cbn} and δ_{cbn} coincide.*

Proof. We first show that δ_{cbn} is at least as discriminating $\delta^{\mathrm{c}}_{\mathrm{cbn}}$. Let be $M, N \in \Lambda_{\oplus}$. By definition of $\mathscr{L}_{\mathscr{F}^{\mathrm{cbn}}}$, we know that $\hat{s}^{\mathrm{cbn}}(M), \hat{s}^{\mathrm{cbn}}(N) \in \mathcal{S}_{\mathscr{L}_{\mathscr{F}^{\mathrm{cbn}}}}$. Moreover, we know that $\mathscr{C}_{\mathrm{cbn}}$ is preserved by $\mathscr{F}^{\mathrm{cbn}}$. So we can apply Proposition 4, and we see that $\delta^{\mathrm{c}}_{\mathscr{C}_{\mathrm{cbn}}}(\hat{s}^{\mathrm{cbn}}(M), \hat{s}^{\mathrm{cbn}}(N)) \leq \delta_{\mathrm{cbn}}(M, N)$, and soundness follows. When proving completeness part, we rely on an "intrinsic" characterization of δ_{cbn}. The details can be found in [7]. □

6.3 Call-by-Value

In a similar way, we can endow $\Lambda_{\oplus}$ with a call-by-value semantics, and embed it into $\Lambda^{!}_{\oplus}$. We are then able to define a suitable fragment of $\mathscr{M}^{!}_{\oplus}$, a suitable set of tuple contexts preserving it, and a characterization of a call-by-value context distance for $\Lambda_{\oplus}$ follows [7]. While the construction of the stable fragment (and the set of tuple contexts to consider) are more involved than in the call-by-name case, we noticed that the characterization we obtain seems to have some similarities with the way environmental bisimulation for a call-by-value probabilistic λ-calculus was defined in [32].

7 Related Work

This is definitely *not* the first work on metrics in the context of programming languages semantics. A very nice introduction to the topic, together with a comprehensive (although outdated) list of references can be found in [35]. One of the many uses of metrics is as an alternative to order-theoretic semantics. This has also been applied to higher-order languages, and to *deterministic* PCF [15].

If one focuses on probabilistic programming languages, the first attempts at using metrics as a way to measure "how far" two programs are, algebraically or behaviorally, are due to Giacalone et al. [18], and Desharnais et al. [11, 12], who both consider process algebras in the style of Milner's CCS. Most of further work in this direction has focused on concurrent specifications. Among the recent advances in this direction (and without any hope of being comprehensive), we can cite Gebler et al.'s work on uniform continuity as a way to enforce compositionality in metric reasoning [17, 16]. Great inspiration for this work came from the many contributions on metrics for labeled Markov chains and processes appeared in the last twenty years (e.g. [36, 13]).

8 Conclusions

We have shown *how* the context distance can be characterized so as to simplify concrete proofs, and *to which extent* this metric trivializes. All this has been done in a universal linear λ-calculus for probabilistic computation. This clarifies to which extent refining equivalences into metrics is worth in such a scenario. The tuple-based techniques in Section 5.5 are potentially very interesting in view of possible applications to cryptography, as hinted in [4]. This is indeed what we are working on currently.

References

1. H. P. Barendregt. *The Lambda Calculus – Its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 1984.
2. H. P. Barendregt, W. Dekkers, and R. Statman. *Lambda Calculus with Types*. Perspectives in logic. Cambridge University Press, 2013.
3. A. Bizjak and L. Birkedal. Step-indexed logical relations for probability. In *Proc. of FoSSaCS*, pages 279–294, 2015.
4. A. Cappai and U. Dal Lago. On equivalences, metrics, and polynomial time. In *Proc. of FCT*, pages 311–323, 2015.
5. R. Crubillé and U. Dal Lago. On probabilistic applicative bisimulation and call-by-value λ-calculi. In *Proc. of ESOP*, pages 209–228, 2014.
6. R. Crubillé and U. Dal Lago. Metric reasoning about λ-terms: The affine case. In *Proc. of LICS*, pages 633–644, 2015.
7. R. Crubillé and U. Dal Lago. Metric reasoning about λ-terms: The general case (long version). Available at `http://arxiv.org/abs/1701.05521`, 2016.
8. R. Crubillé, U. Dal Lago, D. Sangiorgi, and V. Vignudelli. On applicative similarity, sequentiality, and full abstraction. In *Proc. of Correct System Design - Symposium in Honor of Ernst-Rüdiger Olderog on the Occasion of His 60th Birthday*, pages 65–82, 2015.
9. U. Dal Lago, D. Sangiorgi, and M. Alberti. On coinductive equivalences for higher-order probabilistic functional programs. In *Proc. of POPL*, pages 297–308, 2014.
10. U. Dal Lago and M. Zorzi. Probabilistic operational semantics for the lambda calculus. *RAIRO - Theor. Inf. and Applic.*, 46(3):413–450, 2012.
11. J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden. Metrics for labeled markov systems. In *Proc. of CONCUR*, 1999.
12. J. Desharnais, R. Jagadeesan, V. Gupta, and P. Panangaden. The metric analogue of weak bisimulation for probabilistic processes. In *Proc. of LICS*, pages 413–422, 2002.
13. J. Desharnais, F. Laviolette, and M. Tracol. Approximate analysis of probabilistic processes: Logic, simulation and games. In *Proc. of QEST*, pages 264–273, 2008.
14. T. Ehrhard, C. Tasson, and M. Pagani. Probabilistic coherence spaces are fully abstract for probabilistic PCF. In *Proc. of POPL*, pages 309–320, 2014.
15. M. Escardo. A metric model of PCF. Proceedings of the Workshop on Realizability Semantics and Applications. Available at `http://www.cs.bham.ac.uk/~mhe/papers/metricpcf.pdf`, 1999.
16. D. Gebler, K. G. Larsen, and S. Tini. Compositional metric reasoning with probabilistic process calculi. In *Proc. of FoSSaCS*, pages 230–245, 2015.
17. D. Gebler and S. Tini. SOS specifications of probabilistic systems by uniformly continuous operators. In *Proc. of CONCUR*, pages 155–168, 2015.
18. A. Giacalone, C. chang Jou, and S. A. Smolka. Algebraic reasoning for probabilistic concurrent systems. In *Proc. IFIP TC2*, pages 443–458. North-Holland, 1990.
19. J. Girard. Linear logic. *Theor. Comput. Sci.*, 50:1–102, 1987.
20. S. Goldwasser and S. Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.
21. N. D. Goodman, V. K. Mansinghka, D. M. Roy, K. Bonawitz, and J. B. Tenenbaum. Church: a language for generative models. In *UAI 2008*, pages 220–229, 2008.
22. C. Jones and G. D. Plotkin. A probabilistic powerdomain of evaluations. In *Prof. of LICS*, pages 186–195, 1989.

23. A. Jung and R. Tix. The troublesome probabilistic powerdomain. *Electr. Notes Theor. Comput. Sci.*, 13:70–91, 1998.
24. C. D. Manning and H. Schütze. *Foundations of statistical natural language processing*, volume 999. MIT Press, 1999.
25. R. Mardare. Logical foundations of metric behavioural theory for markov processes. Doctoral Thesis. In Preparation, 2016.
26. S. Park, F. Pfenning, and S. Thrun. A probabilistic language based on sampling functions. *ACM Trans. Program. Lang. Syst.*, 31(1), 2008.
27. J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.
28. G. D. Plotkin. LCF considered as a programming language. *Theor. Comput. Sci.*, 5(3):223–255, 1977.
29. N. Ramsey and A. Pfeffer. Stochastic lambda calculus and monads of probability distributions. In *Prof. of POPL*, pages 154–165, 2002.
30. N. Saheb-Djahromi. Probabilistic LCF. In *Proc. of MFCS*, pages 442–451, 1978.
31. D. Sangiorgi. On the bisimulation proof method. *Mathematical Structures in Computer Science*, 8:447–479, 1998.
32. D. Sangiorgi and V. Vignudelli. Environmental bisimulations for probabilistic higher-order languages. In *Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2016, St. Petersburg, FL, USA, January 20 - 22, 2016*, pages 595–607, 2016.
33. A. K. Simpson. Reduction in a linear lambda-calculus with applications to operational semantics. In *Proc. of RTA*, pages 219–234, 2005.
34. S. Thrun. Robotic mapping: A survey. *Exploring artificial intelligence in the new millennium*, pages 1–35, 2002.
35. F. van Breugel. An introduction to metric semantics: operational and denotational models for programming and specification languages. *Theor. Comput. Sci.*, 258(1-2):1–98, 2001.
36. F. van Breugel and J. Worrell. A behavioural pseudometric for probabilistic transition systems. *Theor. Comput. Sci.*, 331(1):115–142, 2005.