

Lambda Calculus and Probabilistic Computation

Claudia Faggian
CNRS, France

Simona Ronchi della Rocca
Univ. Torino, Italy

Abstract—We introduce two extensions of the λ -calculus with a probabilistic choice operator, $\Lambda_{\oplus}^{\text{cbv}}$ and $\Lambda_{\oplus}^{\text{cbn}}$, modeling respectively call-by-value and call-by-name probabilistic computation. We prove that both enjoys confluence and standardization, in an extended way: we revisit these two fundamental notions to take into account the asymptotic behaviour of terms. The common root of the two calculi is a further calculus based on Linear Logic, $\Lambda_{\oplus}^!$, which allows us to develop a unified, modular approach.

I. INTRODUCTION

The pervasive role of stochastic models in a variety of domains (such as machine learning, natural language, verification) has prompted a vast body of research on probabilistic programming languages; such a language supports at least discrete distributions by providing an operator which models sampling. In particular, the functional style of probabilistic programming, pioneered by [29], attracts increasing interest because it allows for higher-order computation, and offers a level of abstraction well-suited to deal with mathematical objects. Early work [19], [25], [23], [27], [24] has evolved in a growing body of software development and theoretical research. In this context, the λ -calculus has often been used as a core language.

In order to model higher-order probabilistic computation, it is a natural approach to take the λ -calculus as general paradigm, and to enrich it with a probabilistic construct. The most simple and concrete way to do so ([10], [8], [13]) is to equip the untyped λ -calculus with an operator \oplus , which models flipping a fair coin. This suffices to have universality, as proved in [8], in the sense that the calculus is sound and complete with respect to *computable* probability distributions. The resulting calculus is however *non-confluent*, as it has been observed early (see [8] for an analysis). We revise the issue in Example 1. The problem with confluence is handled in the literature by fixing a deterministic reduction strategy, typically the leftmost-outermost strategy. This is not satisfactory both for theoretical and practical reasons, as we discuss later.

In this paper, we propose a more general point of view. Our goal is a foundational calculus, which plays the same role as the λ -calculus does for deterministic computation. More precisely, taking the point of view propounded by Plotkin in [26], we discriminate between a *calculus* and a *programming language*. The former defines the reduction rules, independently from any reduction strategy, and enjoys confluence and standardization, the latter is specified by a deterministic strategy (an abstract machine). *Standardization* is what relates the two: the programming language implements the standard

strategy associated to the calculus. Indeed, standardization implies the existence of a strategy (the standard strategy) which is guaranteed to reach the result, if it exists.

In this spirit, we consider a probabilistic *calculus* to be characterized by a specific calling mechanism; the reduction is otherwise only constrained by the need of discriminating between duplicating a function which samples from a distribution, and duplicating the result of sampling. Think of tossing a coin and duplicating the result, versus tossing the coin twice, which is indeed the issue at the core of confluence failure, as the following examples (adapted from [9], [8]) show.

Example 1 (Confluence). *Let us consider the untyped λ -calculus extended with a binary operator \oplus which models fair, binary probabilistic choice: $M \oplus N$ reduces to either M or N with equal probability $1/2$; we write this as $M \oplus N \rightarrow \{M^{\frac{1}{2}}, N^{\frac{1}{2}}\}$. Intuitively, the result of evaluating a probabilistic term is a distribution on its possible values.*

- 1) *Consider the term PQ , where $P = (\lambda z.z \text{ XOR } z)$, and $Q = (T \oplus F)$; XOR is the standard construct for exclusive OR, $T = \lambda xy.x$ and $F = \lambda xy.y$ code the boolean values.*
 - *If we first reduce Q , we obtain $(\lambda z.z \text{ XOR } z)T$ or $(\lambda z.z \text{ XOR } z)F$, with equal probability $1/2$. This way, PQ evaluates to $\{F^1\}$, i.e. F with probability 1.*
 - *If we reduce the outermost redex first, PQ reduces to $(T \oplus F) \text{ XOR } (T \oplus F)$, and the term evaluates to the distribution $\{T^{\frac{1}{2}}, F^{\frac{1}{2}}\}$.*
- 2) *The same phenomenon appears even if we restrict ourselves to call-by-value. Consider for example the reductions of PN with P as in 1), and $N = (\lambda xy.x \oplus y)$. We obtain the same two different distributions as above.*

In this paper, we define two probabilistic λ -calculi, respectively based on the call-by-value (CbV) and call-by-name (CbN) calling mechanism. Both enjoy confluence and standardization, *in an extended way*: indeed we revisit these two fundamental notions to take into account the asymptotic behaviour of terms. The common root of the two calculi is a further calculus based on Linear Logic, which is an extension of Simpson's linear λ -calculus [31], and which allows us to develop a unified, modular approach.

Content and Contributions: In Section IV, we introduce a call-by-value calculus, denoted $\Lambda_{\oplus}^{\text{cbv}}$, as a probabilistic extension of the call-by-value λ -calculus of Plotkin (where the β -reduction fires only in case the argument is a value, *i.e.* either a variable or a λ -abstraction). We choose to study in

detail *call-by-value* for two main reasons. First, it is the most relevant mechanism to probabilistic programming (most of the abstract languages we cited are call-by-value, but also real-world stochastic programs such as Church [17]). Second, call-by-value is a mechanism in which dealing with functions, and duplication of functions, is clean and intuitive, which allows us to address the issue at the core of confluence failure. The definition of value (in particular, a probabilistic choice is not a value) together with a suitable restriction of the evaluation context for the probabilistic choice, allow us to recover key results: confluence and a form of standardization (Section V). Let us recall that, in the classical λ -calculus, standardization means that there is a strategy which is *complete* for all reduction sequences, *i.e.*, for every reduction sequence $M \rightarrow^* N$ there is a *standard* reduction sequence from M to N . A standard reduction sequence with the same property exists also here. An unexpected result is that strategies which are complete in the classical case, are not so here, notably the leftmost strategy.

In Section VI we study the asymptotic behavior of terms. Our leading question is how the asymptotic behaviour of different sequences starting from the same term compare. We first analyze if and in which sense confluence implies that the result of a *probabilistically terminating* computation is unique. We formalize the notion of *asymptotic result* via *limit distributions*, and establish that there is a *unique* maximal one.

In Section VII we address the question of how to find such greatest limit distribution, a question which arises from the fact that evaluation in $\Lambda_{\oplus}^{\text{cbv}}$ is non-deterministic, and different sequences may terminate with different probability. With this aim, we extend the notion of standardization to limits; this extension is non-trivial, and demands the development of new sophisticated proof methods.

We prove that the new notion of standardization supplies a family of *complete reduction strategies* which are guaranteed to reach the unique maximal result. Remarkably, we are able to show that, when evaluating programs, *i.e.*, closed terms, this family *does include* the leftmost strategy. As we have already observed, this is the deterministic strategy which is typically adopted in the literature, in either its call-by-value ([19], [7]) or its call-by-name version ([10], [13]), but without any completeness result with respect to *probabilistic* computation. Our result offers an “*a posteriori*” justification for its use!

The study of $\Lambda_{\oplus}^{\text{cbv}}$ allows us to develop a crisp approach, which we are then able to use in the study of different probabilistic calculi. Because the issue of duplication is central, it is natural to expect a benefit from the fine control over copies which is provided by Linear Logic. In Section IX we use our tools to introduce and study a probabilistic *linear* λ -calculus, $\Lambda_{\oplus}^!$. The linear calculus provides not only a finer control on duplication, but also a modular approach to confluence and standardization, which allow us to formalize a call-by-name version of our calculus, namely $\Lambda_{\oplus}^{\text{cbn}}$, in Section X. We prove that $\Lambda_{\oplus}^{\text{cbn}}$ enjoys properties analogous to those of $\Lambda_{\oplus}^{\text{cbv}}$, in particular confluence and standardization.

In Section II we provide the reader with some background and motivational observations. Basic notions of discrete prob-

ability and rewriting are reviewed in Section III.

An extended version of this paper, with more details and proofs, is available online [16].

Related Work: The idea of extending the λ -calculus with a probabilistic construct is not new; without any ambition to be exhaustive, let us cite [23], [27], [10], [13], [8], [5]. In all these cases, a specific reduction strategy is fixed; they are indeed languages, not calculi, according to Plotkin’s distinction.

The issue about confluence appears every time the λ -calculus is extended with a choice effect: quantum, algebraic, non-deterministic. The ways of framing the same problem in different settings are naturally related, and we were inspired by them. Confluence for an algebraic calculus is dealt with in [1] for the call-by-value, and in [32] for the call-by-name. In the quantum case we would like to cite [7], [6], which are based on Simpson’s calculus [31]. A probabilistic extension of Simpson’s calculus was first proposed in [11]. The language is similar to that of $\Lambda_{\oplus}^!$; however in [11] (similarly to [7], [6]) no reduction (not even β) is allowed in the scope of a $!$ -operator. The reduction there hence corresponds to surface reduction, which in Sec. IX we show to be the *standard strategy* for $\Lambda_{\oplus}^!$.

To our knowledge, the only proposal of a probabilistic λ -calculus in which the reduction is independent from a *specific strategy* is for *call-by-name*, namely the calculus of [20], in the line of work of differential [14] and algebraic [32] λ -calculus. The focus in [20] is essentially semantical, as the author want to study an equational theory for the λ -calculus, based on an extension of Böhm trees. [20] develops results which in their essence are similar to those we obtain for call-by-name in Sec. X, in particular confluence and standardization, even if his calculus –which internalizes the probabilistic behavior– is quite different from ours, and so are the proof techniques.

Finally, we wish to mention that proposals of a probabilistic λ -calculus could also be extracted from semantical models, such as the one in [3], which develops an idea earlier presented in [30], and in which the notion of graph models for λ -calculus has been extended with a probabilistic construct.

II. BACKGROUND AND MOTIVATIONAL OBSERVATIONS

In this section, we first review –in a non-technical way– the specific features of probabilistic programs, and how they differ from classical ones. We then focus on some motivational observations which are relevant to our work. First, we give an example of features which are lost if a programming language is characterized by a strategy which is not rooted in a more general calculus. Then, we illustrate some of the issues which appear when we study a general calculus, instead of a specific reduction strategy. Addressing these issues will lead us to develop new notions and tools.

A. Classical vs. Probabilistic Programs

A *classical* program defines a *deterministic input-output* relation; it terminates (on a given input), or does not; if it terminates, the program only runs for a finite number of steps. Instead, a *probabilistic* program generates a probability

distribution over possible outputs; it terminates (on a given input) with a certain probability; it may have runs which take infinitely many steps even when termination has probability 1.

A probabilistic program is a stochastic model. The intuition is that the probabilistic program P is executed, and random choices are made by sampling; this process defines a distribution over all the possible outputs of P . Even if the termination probability is 1 (*almost sure termination*), that degree of certitude is typically not reached in any finite number of steps, but it appears as a limit. A standard example is a term M which reduces to either the normal form T or M itself, with equal probability $1/2$. After n steps, M reduces to T with probability $\frac{1}{2} + \frac{1}{2^2} + \dots + \frac{1}{2^n}$. Only at the limit this computation terminates with probability 1.

Probabilistic vs. Quantitative: The notion of *probabilistic termination* is what sets apart probabilistic λ -calculus from other quantitative calculi such as those in [1], [14], [32], and from the non-deterministic λ -calculus [9]. For this reason, the *asymptotic behaviour* of terms will be the focus of this paper.

B. Confluence of the calculus is relevant to programming

Functional languages have their foundation in the λ -calculus and its properties, and such properties (notably, confluence and standardization) have theoretical and practical implications. A strength of *classical* functional languages -which is assuming growing importance- is that they are inherently parallel (we refer e.g. to [22] for discussion on *deterministic* parallel programming): every sub-expression can be evaluated in parallel, because of referential transparency; still, we can perform reasoning, testing and debugging on a program using a sequential model, because the result of the calculus is independent from the evaluation order. Not to force a sequential strategy impacts the implementation of the language, but also the *conception* of programs. As advocated by Harper, the parallelism of functional languages exposes the “*dependency structure of the computation by not introducing any dependencies that are not forced on us by the nature of the computation itself.*”

This feature of functional languages is rooted in the *confluence* of the λ -calculus, and is an example of what is lost in the probabilistic setting, if we give-up either confluence, or the possibility of non-deterministic evaluation.

C. The result of probabilistic computation

A ground for our approach is the distinction between calculus and language. Some of the issues which we will need to address do not appear when working with probabilistic languages, because they are based on a simplification of the λ -calculus. Programming languages only evaluate programs, i.e., *closed terms* (without free variables). A striking simplification appears from another crucial restriction, *weak evaluation*, which does not evaluate function bodies (the scope of λ -abstractions). In *weak call-by-value* (base of the ML/CAML family of probabilistic languages) values are normal forms.

What is the result of a probabilistic computation is well understood only in the case of *programming languages*: the result of a program is a distribution on its possible outcomes, which

are *normal forms* w.r.t. a chosen strategy. In the literature of probabilistic λ -calculus, two main deterministic strategies have been studied: weak left strategy in CbV [8] and head strategy in CbN [13], whose normal forms are respectively the closed values and the head normal forms.

When considering a calculus instead of a language, the identity between normal forms and results does not hold anymore, with important consequences in the definition of limit distributions. We investigate this issue in Sec. VI. The approach we develop is general and uniform to all our calculi.

III. TECHNICAL PRELIMINARIES

We review basic notions on discrete probability and rewriting which we use through the paper. We assume that the reader has some familiarity with the λ -calculus.

A. Basics on Discrete Probability

A *discrete probability space* is given by a pair (Ω, μ) , where Ω is a countable set, and μ is a *discrete probability distribution* on Ω , i.e. is a function from Ω to $[0,1] \subset \mathbb{R}$ such that $\|\mu\| := \sum_{\omega \in \Omega} \mu(\omega) = 1$. In this case, a probability measure is assigned to any subset $\mathcal{A} \subseteq \Omega$ as $\mu(\mathcal{A}) = \sum_{\omega \in \mathcal{A}} \mu(\omega)$. In the language of probability theory, a subset of Ω is called an *event*.

Let (Ω, μ) be as above. Any function $F : \Omega \rightarrow \Delta$, where Δ is another countable set, induces a probability distribution μ^F on Δ by composition: $\mu^F(d \in \Delta) := \mu(F^{-1}(d))$ i.e. $\mu\{\omega \in \Omega : F(\omega) = d\}$. In the language of probability theory, F is called a *discrete random variable* on (Ω, μ) .

- Example 2 (Die).** 1) Consider tossing a die once. The space of possible outcomes is the set $\Omega = \{1,2,3,4,5,6\}$. The probability measure μ of each outcome is $1/6$. The event “result is odd” is the subset $\mathcal{O} = \{1,3,5\}$, whose probability measure is $\mu(\mathcal{O}) = 1/2$.
2) Let Δ be a set with two elements $\{\text{Even}, \text{Odd}\}$, and F the obvious function from Ω to Δ . F induces a distribution on Δ , with $\mu^F(\text{Even}) = 1/2$ and $\mu^F(\text{Odd}) = 1/2$.

B. Subdistributions and $\text{DST}(\Omega)$

Given a countable set Ω , a function $\mu : \Omega \rightarrow [0,1]$ is a probability *subdistribution* if $\|\mu\| \leq 1$. We write $\text{DST}(\Omega)$ for the set of subdistributions on Ω . With a slight abuse of language, we will use the term *distribution* also for subdistribution. Subdistributions allow us to deal with partial results and non-successful computations.

Order: $\text{DST}(\Omega)$ is equipped with the standard order relation of functions : $\mu \leq \rho$ if $\mu(\omega) \leq \rho(\omega)$ for each $\omega \in \Omega$.

Support: The support of μ is $\text{Supp}(\mu) = \{\omega : \mu(\omega) > 0\}$.

Representation: We represent a distribution by explicitly indicating the support, and (as superscript) the probability assigned to each element by μ . We write $\mu = \{a_0^{p_0}, \dots, a_n^{p_n}\}$ if $\mu(a_0) = p_0, \dots, \mu(a_n) = p_n$ and $\mu(a_j) = 0$ otherwise.

C. Multidistributions

To syntactically represent the global evolution of a probabilistic system, we rely on the notion of multidistribution [2].

A *multiset* is a (finite) list of elements, modulo reordering, i.e. $[a, b, a] = [a, a, b] \neq [a, b]$; the multiset $[a, a, b]$ has three elements. Let \mathcal{X} be a countable set and \mathfrak{m} a multiset of pairs of the form pM , with $p \in]0, 1]$, and $M \in \mathcal{X}$. We call $\mathfrak{m} = [p_i M_i \mid i \in I]$ (where the index set I ranges over the elements of \mathfrak{m}) a *multidistribution on \mathcal{X}* if $\sum_{i \in I} p_i \leq 1$. We denote by $\text{MDST}(\mathcal{X})$ the set of all multidistributions on \mathcal{X} .

We write the multidistribution $[1M]$ simply as $[M]$. The sum of multidistributions is denoted by $+$, and it is the concatenation of lists. The product $q \cdot \mathfrak{m}$ of a scalar q and a multidistribution \mathfrak{m} is defined pointwise: $q \cdot [p_1 M_1, \dots, p_n M_n] = [(qp_1)M_1, \dots, (qp_n)M_n]$.

Intuitively, a multidistribution $\mathfrak{m} \in \text{MDST}(\mathcal{X})$ is a syntactical representation of a discrete probability space where at each element of the space is associated a probability and a term of \mathcal{X} . To the multidistribution $\mathfrak{m} = [p_i M_i \mid i \in I]$ we associate a probability distribution $\mu \in \text{DST}(\mathcal{X})$ as follows:

$$\mu(M) = \begin{cases} p & \text{if } p = \sum_{i \in I} p_i \text{ s.t. } M_i = M \\ 0 & \text{otherwise;} \end{cases}$$

we call μ the *probability distribution associated to \mathfrak{m}* .

Example 3 (Distribution vs. multidistribution). If $\mathfrak{m} = [\frac{1}{2}a, \frac{1}{2}a]$, then $\mu = \{a^1\}$. Please observe the difference between *distribution* and *multidistribution*: if $\mathfrak{m}' = [1a]$, then $\mathfrak{m} \neq \mathfrak{m}'$, but $\mu = \mu'$.

D. Binary relations (notations and basic definitions)

Let \rightarrow_r be a binary relation on a set \mathcal{X} . We denote \rightarrow_r^* its reflexive and transitive closure. We denote $=_r$ the reflexive, symmetric and transitive closure of \rightarrow_r . If $u \in \mathcal{X}$, we write $u \not\rightarrow_r$ if there is no $t \in \mathcal{X}$ such that $u \rightarrow_r t$; in this case, u is in \rightarrow_r -normal form. **Figures convention:** as is standard, in the figures we depict \rightarrow^* as \Rightarrow ; solid arrows are universally quantified, dashed arrows are existentially quantified.

Confluence and Commutation: Let $r, s, t, u \in \mathcal{X}$. The relations \rightarrow_1 and \rightarrow_2 on \mathcal{X} *commute* if $(r \rightarrow_1^* s \text{ and } r \rightarrow_2^* t)$ imply there is u such that $(s \rightarrow_2^* u \text{ and } t \rightarrow_1^* u)$; they *diamond-commute* (\diamond -commute) if $(r \rightarrow_1 s \text{ and } r \rightarrow_2 t)$ imply there is u such that $(s \rightarrow_2 u \text{ and } t \rightarrow_1 u)$. The relation \rightarrow is *confluent* (resp. *diamond*) if it commutes (resp. \diamond -commutes) with itself. It is well known that \diamond -commutation implies commutation, and diamond implies confluence.

IV. CALL-BY-VALUE CALCULUS $\Lambda_{\oplus}^{\text{cbv}}$

We define $\Lambda_{\oplus}^{\text{cbv}}$, a CbV probabilistic λ -calculus.

A. Syntax of $\Lambda_{\oplus}^{\text{cbv}}$

1) *The language:* Terms and values are generated respectively by the grammars:

$$\begin{array}{ll} M, N, P, Q ::= x \mid \lambda x. M \mid MM \mid M \oplus M & \text{(terms } \Lambda_{\oplus}) \\ V, W ::= x \mid \lambda x. M & \text{(values } \mathcal{V}) \end{array}$$

where x ranges over a countable set of *variables* (denoted by x, y, \dots). Λ_{\oplus} and \mathcal{V} denote respectively the set of terms and of values. Free variables are defined as usual. $M[N/x]$ denotes the term obtained by capture-avoiding substitution of N for each free occurrence of x in M .

Contexts (C) and surface contexts (S) are generated by the grammars:

$$\begin{array}{ll} \mathbf{C} ::= \square \mid M\mathbf{C} \mid \mathbf{C}M \mid \lambda x. \mathbf{C} \mid \mathbf{C} \oplus M \mid M \oplus \mathbf{C} & \text{(contexts)} \\ \mathbf{S} ::= \square \mid M\mathbf{S} \mid \mathbf{S}M & \text{(surface contexts)} \end{array}$$

where \square denotes the *hole* of the term context. Given a term context \mathbf{C} , we denote by $\mathbf{C}(M)$ the term obtained from \mathbf{C} by filling the hole with M , allowing the capture of free variables. All surface contexts are contexts. Since the hole will be filled with a redex, *surface contexts* formalize the fact that the redex (the hole) is not in the scope of a λ -abstraction, nor of a \oplus .

$\text{MDST}(\Lambda_{\oplus})$ denotes the set of *multi-distributions* on Λ_{\oplus} .

2) *Reductions:* We first define reduction rules on terms (Fig. 1), and one-step reduction from terms to multidistributions (Fig. 2). We then lift the definition of reduction to a binary relation on $\text{MDST}(\Lambda_{\oplus})$.

Observe that, usually, a reduction step is given by the closure under context of the reduction rules. However, to define a reduction from term to term is not informative enough, because we still have to account for the probability. The meaning of $M \oplus N$ is that this term reduces to either M or N , with equal probability $\frac{1}{2}$. There are various way to formalize this fact; here, we use multidistributions.

a) *Reduction Rules and Steps:* The reduction rules on the terms of Λ_{\oplus} are defined in Fig. 1.

β_v -rule	Probabilistic rules
$(\lambda x. M)V \mapsto_{\beta_v} M[V/x] \text{ if } V \in \mathcal{V}$	$M \oplus N \mapsto_{l \oplus} M \quad M \oplus N \mapsto_{r \oplus} N$

Figure 1: Reduction Rules

The (one-step) reduction relations $\rightarrow_{\beta_v}, \rightarrow_{\oplus} \subseteq \Lambda_{\oplus} \times \text{MDST}(\Lambda_{\oplus})$ are defined in Fig. 2. Observe that the probabilistic rules $\mapsto_{r \oplus, l \oplus}$ are closed only under surface contexts, while the reduction rule \mapsto_{β_v} is closed under general context \mathbf{C} (hence $\Lambda_{\oplus}^{\text{cbv}}$ is a conservative extension of Plotkin's CbV λ -calculus, see IV-B). We denote by \rightarrow the union $\rightarrow_{\beta_v} \cup \rightarrow_{\oplus}$.

$(\lambda x. M)V \mapsto_{\beta_v} M[V/x] \quad V \in \mathcal{V}$
$\mathbf{C}((\lambda x. M)V) \rightarrow_{\beta_v} [\mathbf{C}(M[V/x])]$
$M \oplus N \mapsto_{l \oplus} M \quad M \oplus N \mapsto_{r \oplus} N$
$\mathbf{S}(M \oplus N) \rightarrow_{\oplus} [\frac{1}{2}\mathbf{S}(M), \frac{1}{2}\mathbf{S}(N)]$

Figure 2: Reduction Steps

b) *Lifting:* We lift the reduction relation $\rightarrow \subseteq \Lambda_{\oplus} \times \text{MDST}(\Lambda_{\oplus})$ to a relation $\Rightarrow \subseteq \text{MDST}(\Lambda_{\oplus}) \times \text{MDST}(\Lambda_{\oplus})$, as defined in Fig. 3. Observe that \Rightarrow is a reflexive relation.

$\frac{}{[M] \Rightarrow [M]} \quad L1$	$\frac{M \rightarrow \mathfrak{m}}{[M] \Rightarrow \mathfrak{m}} \quad L2$	$\frac{([M_i] \Rightarrow \mathfrak{m}_i)_{i \in I}}{[p_i M_i \mid i \in I] \Rightarrow \sum_{i \in I} p_i \cdot \mathfrak{m}_i} \quad L3$
---	--	--

Figure 3: Lifting of \rightarrow

We define in the same way the lifting of any relation $\rightarrow_r \subseteq \Lambda_{\oplus} \times \text{MDST}(\Lambda_{\oplus})$ to a binary relation \Rightarrow_r on $\text{MDST}(\Lambda_{\oplus})$. In particular, we lift $\rightarrow_{\beta_v}, \rightarrow_{\oplus}$ to $\Rightarrow_{\beta_v}, \Rightarrow_{\oplus}$.

c) *Reduction sequences:* A \Rightarrow -sequence (*reduction sequence*) from \mathfrak{m} is a sequence $\mathfrak{m} = \mathfrak{m}_0, \dots, \mathfrak{m}_i, \mathfrak{m}_{i+1}, \dots$ such that $\mathfrak{m}_i \Rightarrow \mathfrak{m}_{i+1} \ (\forall i)$. We write $\mathfrak{m} \Rightarrow^* \mathfrak{n}$ to indicate that there is a *finite* sequence from \mathfrak{m} to \mathfrak{n} , and $\langle \mathfrak{m}_n \rangle_{n \in \mathbb{N}}$ for an *infinite* sequence.

d) β_v *equivalence:* We write $=_{\beta_v}$ for the transitive, reflexive and symmetric closure of \Rightarrow_{β_v} ; abusing the notation, we will write $M =_{\beta_v} N$ for $[M] =_{\beta_v} [N]$.

e) Normal Forms: \mathcal{N} denotes the set of \rightarrow -normal forms. Given $\rightarrow_r \in \Lambda_{\oplus} \times \text{MDST}(\Lambda_{\oplus})$, a term M is in \rightarrow_r -normal form if $M \not\rightarrow_r$, i.e. there is no m such that $M \rightarrow_r m$. It is easy to check that all closed \rightarrow -normal forms are values, however a value is not necessarily a \rightarrow -normal form.

3) *Full Lifting:* The definition of lifting allows us to apply a reduction step \rightarrow to any number of M_i in the multidistribution $m = [p_i M_i \mid i \in I]$. If no M_i is reduced, then $m \Rightarrow m$ (the relation \Rightarrow is reflexive). Another important case is when *all* M_i for which a reduction step is possible are indeed reduced. This notion of *full* reduction, denoted by \Rightarrow , is defined as follows.

$$\frac{M \not\rightarrow}{[M] \Rightarrow [M]} \quad \frac{M \rightarrow m}{[M] \Rightarrow m} \quad \frac{([M_i] \Rightarrow m_i)_{i \in I}}{[p_i M_i \mid i \in I] \Rightarrow \sum_{i \in I} p_i \cdot m_i}$$

Obviously, $\Rightarrow \subseteq \rightarrow$. Similarly to lifting, also the notion of full lifting can be extended to any reduction. For any $\rightarrow_r \subseteq \Lambda_{\oplus} \times \text{MDST}(\Lambda_{\oplus})$, its full lifting is denoted by $\Rightarrow_r \subseteq \text{MDST}(\Lambda_{\oplus}) \times \text{MDST}(\Lambda_{\oplus})$. The relation \Rightarrow plays an important role in VII.

B. $\Lambda_{\oplus}^{\text{cbv}}$ and the λ -calculus

A comparison between $\Lambda_{\oplus}^{\text{cbv}}$ and the λ -calculus is in order.

Let Λ be the set of λ -terms; we denote by Λ^{cbn} the CbN λ -calculus, equipped with the reduction \rightarrow_{β} [4], and by Λ^{cbv} the CbV λ -calculus, equipped with the reduction \rightarrow_{β_v} [26].

$\Lambda_{\oplus}^{\text{cbv}}$ is a conservative extension of Λ^{cbv} . A translation $(\cdot)_{\lambda}$: $\Lambda_{\oplus} \rightarrow \Lambda$ can be defined as follows, where z is a fresh variable which is used by no term:

$$\begin{aligned} (x)_{\lambda} &= x \\ (M \oplus N)_{\lambda} &= z(M)_{\lambda}(N)_{\lambda} \end{aligned} \quad \left| \begin{aligned} (MN)_{\lambda} &= (M)_{\lambda}(N)_{\lambda} \\ (\lambda x. M)_{\lambda} &= \lambda x. (M)_{\lambda} \end{aligned} \right.$$

The translation is injective (if $(M)_{\lambda} = (N)_{\lambda}$ then $M = N$) and preserves values.

Proposition 4 (Simulation). *The translation is sound and complete. Let $M, N \in \Lambda_{\oplus}$.*

- 1) $M \rightarrow_{\beta_v} N$ implies $(M)_{\lambda} \rightarrow_{\beta_v} (N)_{\lambda}$;
- 2) $(M)_{\lambda} \rightarrow_{\beta_v} (N)_{\lambda}$ implies there is a (unique) N , with $Q = (N)_{\lambda}$ and $M \rightarrow_{\beta_v} N$.

C. Discussion (Surface Contexts)

The notion of surface context which we defined is familiar in the setting of λ -calculus: it corresponds to *weak evaluation*, which we discussed in II-C. In $\Lambda_{\oplus}^{\text{cbv}}$, the \rightarrow_{β_v} -reduction is *unrestricted*. Closing the \oplus -rules under surface context S expresses the fact that the \oplus -redex is not reduced under λ -abstraction, nor in the scope of another \oplus . The former is fundamental to confluence: it means that a function which samples from a distribution can be duplicated, but we cannot pre-evaluate the sampling. The latter is a technical simplification, which we adopt to avoid unessential burdens with associativity. To require no reduction in the scope of \oplus is very similar to allow no reduction in the branches of an if-then-else.

V. CONFLUENCE AND STANDARDIZATION

A. Confluence

We prove that $\Lambda_{\oplus}^{\text{cbv}}$ is confluent. We modularize the proof using the Hindley-Rosen lemma. The notions of commutation and \diamond -commutation which we use are reviewed in Sec. III-D.

Lemma (Hindley-Rosen). Let \rightarrow_1 and \rightarrow_2 be binary relations on the same set \mathcal{R} . Their union $\rightarrow_1 \cup \rightarrow_2$ is confluent if both \rightarrow_1 and \rightarrow_2 are confluent, and \rightarrow_1 and \rightarrow_2 commute.

The following criterion allows us to *work pointwise* in proving commutation and confluence of binary relations on *multidistributions*, namely \Rightarrow_{β_v} and \Rightarrow_{\oplus} .

Lemma 5 (Pointwise Criterion). *Let $\rightarrow_o, \rightarrow_b \subseteq \Lambda_{\oplus} \times \text{MDST}(\Lambda_{\oplus})$ and $\Rightarrow_o, \Rightarrow_b$ their lifting (as defined in IV-A2b). Property (*) below implies that $\Rightarrow_o, \Rightarrow_b \diamond$ -commute.*

(*) *If $M \rightarrow_b n$ and $M \rightarrow_o s$, then $\exists r$ s.t. $n \Rightarrow_o r$ and $s \Rightarrow_b r$.*

Proof. We prove that (**) $m \Rightarrow_b n$ and $m \Rightarrow_o s$ imply exists r s.t. $n \Rightarrow_o r$ and $s \Rightarrow_b r$. Let $m = [p_i M_i \mid i \in I]$. By definition of lifting, for each M_i , we have $[M_i] \Rightarrow_b n_i$ and $[M_i] \Rightarrow_o s_i$, with $n = \sum p_i \cdot n_i$ and $s = \sum p_i \cdot s_i$. It is easily checked, using (*), that for each M_i , it exists r_i s.t. $n_i \Rightarrow_o r_i$ and $s_i \Rightarrow_b r_i$. Hence $r = \sum p_i \cdot r_i$ satisfies (**). \square

We derive confluence of \Rightarrow_{β_v} from the same property in the CbV λ -calculus [26], [28], using the simulation of Prop. 4.

Lemma 6. *The reduction \Rightarrow_{β_v} is confluent.*

We prove that \Rightarrow_{\oplus} is confluent (indeed, diamond).

Lemma 7. *The reduction \Rightarrow_{\oplus} is diamond.*

Proof. We prove that if $M \rightarrow_{\oplus} n$ and $M \rightarrow_{\oplus} s$, then $\exists r$ such that $n \Rightarrow_{\oplus} r$ and $s \Rightarrow_{\oplus} r$. The claim then follows by Lemma 5, by taking $\rightarrow_o = \rightarrow_b = \rightarrow_{\oplus}$. Let $M = S(P \oplus Q) = S'(P' \oplus Q')$, $n = [\frac{1}{2} S(P), \frac{1}{2} S(Q)]$ and $s = [\frac{1}{2} S'(P'), \frac{1}{2} S'(Q')]$. Because of definition of surface context, the two \oplus -redexes do not overlap: $P' \oplus Q'$ is a subterm of S and $P \oplus Q$ is a subterm of S' . Hence we can reduce those redexes in S and S' , to obtain r . \square

We prove commutation of \Rightarrow_{\oplus} and \Rightarrow_{β_v} by proving a stronger property: they \diamond -commute.

Lemma 8. *The reductions \Rightarrow_{β_v} and \Rightarrow_{\oplus} \diamond -commute.*

Proof. By using Lemma 5, we only need to prove that if $M \rightarrow_{\beta_v} n$ and $M \rightarrow_{\oplus} s$, then $\exists r$ such that $n \Rightarrow_{\oplus} r$ and $s \Rightarrow_{\beta_v} r$. The proof is by induction on M . Observe that M cannot be $\lambda x.P$ or $(\lambda x.P')V$ because neither can contain a \oplus -redex.

- Case $M = P \oplus Q$ is easy; M is the only possible \oplus -redex.
- Case $M = PQ$. If the β_v -redex is inside P , and the \oplus -redex inside Q , then $PQ \rightarrow_{\beta_v} [P'Q]$ and $PQ \rightarrow_{\oplus} [\frac{1}{2} PQ', \frac{1}{2} PQ'']$, with $P \rightarrow_{\beta_v} P'$ and $Q \rightarrow_{\oplus} [\frac{1}{2} Q', \frac{1}{2} Q'']$. Clearly $r = [\frac{1}{2} P'Q', \frac{1}{2} P'Q'']$ satisfies the claim. The dual case is similar.

If both redexes are inside Q (or P), we use induction. Let us write M as $S(Q)$; assume $Q \rightarrow_{\beta_v} [N]$, $Q \rightarrow_{\oplus} [\frac{1}{2} Q', \frac{1}{2} Q'']$, therefore $S(Q) \rightarrow_{\beta_v} [S(N)] = n$ and $S(Q) \rightarrow_{\oplus} [\frac{1}{2} S(Q'), \frac{1}{2} S(Q'')] = s$. We use the i.h. on Q to obtain $r' = [\frac{1}{2} R', \frac{1}{2} R'']$ such that $[N] \Rightarrow_{\oplus} [\frac{1}{2} R', \frac{1}{2} R'']$, $[Q'] \Rightarrow_{\beta_v} [R']$, $[Q''] \Rightarrow_{\beta_v} [R'']$. We conclude that for $r = [\frac{1}{2} S(R'), \frac{1}{2} S(R'')]$, it holds that $n \Rightarrow_{\oplus} r$ and $s \Rightarrow_{\beta_v} r$. \square

Theorem 9. *The reduction \Rightarrow is confluent.*

Proof. By Hindley-Rosen, from Lemmas 8, 6, and 7. \square

Let us call n an \mathcal{N} -multidistribution if $n \in \text{MDST}(\mathcal{N})$. The following fact is an immediate consequence of confluence:

"If m reduces to a \mathcal{N} -multidistribution, this is unique."

1) *Discussion:* While immediate, the above fact is hardly useful, for two reasons. First, we know that probabilistic termination is not necessarily reached in a finite number of steps; the relevant notion is not that $m \Rightarrow^* n \in \text{MDST}(\mathcal{N})$, but rather that of a distribution which is defined as limit by the sequence $\langle m_n \rangle_{n \in \mathbb{N}}$. Secondly, in Plotkin's CbV calculus the result of computation is formalized by the notion of value, and considering normal forms as values is unsound ([26], page 135). In Section VI-B we introduce a suitable notion of limit distribution, and study the implications of confluence on it.

B. A Standardization Property

In this section, we first introduce surface and left reduction as strategies for \Rightarrow . In the setting of the CbV λ -calculus, the former corresponds to weak reduction, the latter to the standard strategy originally defined in [26]. We then establish a standardization result, namely that every *finite* \Rightarrow -sequence can be partially ordered as a sequence in which all surface reductions are performed first. A counterexample shows that in $\Lambda_{\oplus}^{\text{cbv}}$, a standardization result using left reduction fails.

1) *Surface and Left Reduction:* We remind the reader that in the λ -calculus, a *deterministic strategy* defines a function from terms to redexes, associating to every term the next redex to be reduced. More generally, we call reduction *strategy* for \rightarrow a reduction *relation* \rightarrow_a such that $\rightarrow_a \subseteq \rightarrow$. The notion of strategy can be easily formalized through the notion of context. With this in mind, let us consider surface and left contexts.

- *Surface contexts* \mathcal{S} have been defined in Sec.IV-A1.
- *Left contexts* \mathcal{L} are defined by the following grammar:

$$\mathcal{L} ::= \square \mid \mathcal{L}M \mid V\mathcal{L}$$

Note that in particular a left contexts is a surface context.

- We call *surface reduction*, denoted by \xrightarrow{s} (with lifting \xrightarrow{s}) and *left reduction*, denoted by \xrightarrow{l} (with lifting \xrightarrow{l}), the closure of the reduction rules in Fig. 1 under surface contexts and left contexts, respectively. It is clear that $\xrightarrow{s} = \xrightarrow{s}_{\beta_v} \cup \rightarrow_{\oplus}$. Observe that $\xrightarrow{l} \subseteq \xrightarrow{s}$.
- A reduction step $M \rightarrow m$ is *deep*, written $M \xrightarrow{d} m$, if it is not a surface step. A reduction step is *internal* (written $M \xrightarrow{\text{int}} m$) if it is not a left step. Observe that $\xrightarrow{d} \subseteq \xrightarrow{\text{int}}$.

Example 10. ($\xrightarrow{l} \not\subseteq \xrightarrow{s}$) Let $M = x(II)(II)$, where $I = \lambda x.x$. Then $M \xrightarrow{s} [xI(II)]$ and $M \xrightarrow{s} [x(II)I]$; instead, $M \xrightarrow{l} [xI(II)]$, $M \not\xrightarrow{l} [x(II)I]$.

Intuitively, left reduction chooses the leftmost of the surface redexes. More precisely, this is the case for closed terms (for example, the term $(xx)(II)$ has a \xrightarrow{s} -step, but no \xrightarrow{l} -step).

Surface Normal Forms: We denote by \mathcal{S}^{cbv} the set of \xrightarrow{s} -normal forms. We observe that all values are surface normal forms (but the converse does not hold): $\mathcal{V} \subseteq \mathcal{S}^{\text{cbv}}$. The situation is different if we restrict ourselves to closed term, in fact the following result holds, which is easy to check.

Lemma 11. If M is a closed term, the following three are equivalent: (i) M is a \xrightarrow{s} -normal form; (ii) M is a \xrightarrow{l} -normal form; (iii) M is a value.

2) *Finitary Surface Standardization:* The next theorem proves a standardization result, in the sense that every *finite* reduction sequence can be (partially) ordered in a sequence of surface steps followed by a sequence of deep steps.

Theorem 12 (Finitary Surface Standardization). In $\Lambda_{\oplus}^{\text{cbv}}$, if $m \Rightarrow^* n$ then exists r such that $m \xrightarrow{s}^* r$ and $r \xrightarrow{d}^* n$.

Proof. We build on an analogous result for CbV λ -calculus, which is folklore and is proved explicitly in [16]. We then only need to check that deep steps commute with \oplus -steps, which is straightforward technology ([16] gives the full proof). \square

a) *Finitary Left Standardization does not hold:* The following statement is *false* for $\Lambda_{\oplus}^{\text{cbv}}$.

"If $m \Rightarrow^* n$ then there exists r such that $m \xrightarrow{l}^* r$ and $r \xrightarrow{\text{int}}^* n$."

Example 13 (Counter-example). Let us consider the following sequence, where $I = \lambda x.x$ and $M = (II)((\lambda x.y \oplus z)I)$. $[M] \xrightarrow{\text{int}} [(II)(y \oplus z)] \rightarrow_{\oplus} [\frac{1}{2}(II)y, \frac{1}{2}(II)z] \Rightarrow_{\beta_v} [\frac{1}{2}Iy, \frac{1}{2}(II)z]$. If we anticipate the reduction of (II) , we have $M \xrightarrow{\beta_v} [I((\lambda x.y \oplus z)I)]$, from where we cannot reach $[\frac{1}{2}Iy, \frac{1}{2}(II)z]$. Observe that the sequence is already surface-standard!

VI. ASYMPTOTIC EVALUATION

The specificity of probabilistic computation is to be concerned with asymptotic behavior; the focus is not what happens after a finite number n of steps, but when n tends to infinity. In this section, we study the asymptotic behavior of \Rightarrow -sequences with respect to evaluation. The intuition is that a reduction sequence defines a distribution on the possible outcomes of the program. We first clarify what is the outcome of evaluating a probabilistic term, and then we formalize the idea of result "at the limit" with the notion of *limit distribution* (Def. 18). In Sec. VI-B we investigate how the asymptotic result of different sequences starting from the same m compare.

We recall that to each multidistribution m on Λ_{\oplus} is associated a probability distribution $\mu \in \text{DST}(\Lambda_{\oplus})$ (see Sec.III-C). We use the following **letter convention**: given a multidistribution m, n, r, \dots we denote the associated distribution by the corresponding Greek letter μ, ν, ρ, \dots . If $\langle m_n \rangle_{n \in \mathbb{N}}$ is a \Rightarrow -sequence, then $\langle \mu_n \rangle_{n \in \mathbb{N}}$ is the sequence of associated distributions.

A. Probabilistic Evaluation

We start by studying the property of being *valuable* (VI-A1) and by analyzing some examples (VI-A2). This motivates the more general approach we introduce in VI-A3.

1) *To be valuable:* In the CbV λ -calculus, the key property of a term M is *to be valuable*, i.e., M can reduce to a value. *To be valuable* is a yes/no property, whose probabilistic analogous is the probability to reduce to a value. If m describes the result of a computation step, the probability that such a result is a value is simply $\mu(\mathcal{V}) := \sum_{V \in \mathcal{V}} \mu(V)$, i.e. the

- 2) Let $\langle m_n \rangle_{n \in \mathbb{N}}$ be the computation in Example 15.2, starting from $[QQ]$. Then $\langle m_n \rangle_{n \in \mathbb{N}} \Downarrow_{\text{obs}} \{\mathbf{T}^{\frac{1}{2}}, \mathbf{F}^{\frac{1}{2}}\}$.
- 3) Let $\langle m_n \rangle_{n \in \mathbb{N}}$ be the computation in Example 15.3, starting from $[NN]$. Then $\langle m_n \rangle_{n \in \mathbb{N}} \Downarrow_{\text{obs}} \{\mathbf{T}^{\frac{1}{2}}\}$.
- Let $\langle m_n \rangle_{n \in \mathbb{N}}$ be the reduction sequence in Example 16, starting with $[(\lambda x.x)\lambda x.\omega_3\omega_3]$. By taking as set of observations \mathcal{V}_{\sim} , the sequence converges to $\{\lambda x.\omega_3\omega_3^1\}$.

b) *Discussion:* Each observation expresses a result of interest for the evaluation of the term M . To better understand this, let us examine what become our notions of observation in the case of usual (non-probabilistic) CbV λ -calculus. Let $M \rightarrow^* N \in \mathbf{U}$ and $\mathbf{U} \in \text{Obs}$; if $\mathbf{U} \in \{\mathcal{V}\}$ then M is valuable, if $\mathbf{U} \in \mathcal{V}_{\sim}$, then M reduces to the value N up to β_v -equivalence, if $\mathbf{U} \in \{\mathcal{N}\}$, then M normalizes, finally $\mathbf{U} = \{N\} \in \mathcal{N}_{\{\cdot\}}$ means that M has normal form N . We say that $\mathbf{U} \in \text{Obs}$ is a result of evaluating M , if $M \rightarrow^* N \in \mathbf{U}$. Clearly, fixed Obs , confluence implies that the result of evaluating M , if any, is unique.

B. Uniqueness and Adequacy of the Evaluation

In this section, we adapt similar results from [15], to which we refer for details. We assume a set Obs to be fixed, hence we omit the index. For concreteness, think of \mathcal{V}_{\sim} , but the results only depend on the properties in Def. 17, and on confluence.

How do different reduction sequences from the same initial m compare? More precisely, assume $m \Rightarrow^{\infty} \rho$ and $m \Rightarrow^{\infty} \mu$, how do ρ and μ compare? Intuitively, the limit distributions of m (which are the result of a *probabilistically terminating* sequence) play the role of normal forms in finitary termination. As confluence implies uniqueness of normal forms, a similar property holds when considering *probabilistic termination* and limits, in the sense that each m has a *unique maximal limit distribution* (Thm. 22). While the property is similar, the proof is not as immediate as in the finitary case. The key result is Lemma 21 which implies both that $\text{Lim}(m)$ has a greatest element (Thm. 22), and adequacy of the evaluation (Thm. 23).

Recall that the order \leq on distributions is defined pointwise (Sec. III-A).

Lemma 21 (Main Lemma). $\Lambda_{\oplus}^{\text{cbv}}$ has the following property: $\forall m, s$, if $\mu \in \text{Lim}(m)$, and $m \Rightarrow^* s$, then $s \Rightarrow^{\infty} \sigma$ with $\mu \leq \sigma$. Moreover, if μ is maximal in $\text{Lim}(m)$ then $\sigma = \mu$.

Proof. Let $\mu \in \text{Lim}(m)$, and $\langle m_n \rangle_{n \in \mathbb{N}}$ be a sequence from $m = m_0$ which converges to μ . Assume $m \Rightarrow^* s$. As illustrated in Fig. 4, from s we build a sequence $s = s_{m_0} \Rightarrow^* s_{m_1} \Rightarrow^* s_{m_2} \dots$, where each segment $s_{m_i} \Rightarrow^* s_{m_{i+1}}$ ($i \geq 0$) is given by confluence from $m_i \Rightarrow^* s_{m_i}$ and $m_i \Rightarrow m_{i+1}$. Let $\langle s_n \rangle_{n \in \mathbb{N}}$ be the concatenation of all such segments and let σ be its limit distribution. Clearly, $\sigma \in \text{Lim}(m)$. Since by construction $m_i \Rightarrow^* s_{m_i}$, then for each $\mathbf{V} \in \text{Obs}$, $\mu_i(\mathbf{V}) \leq \sigma(\mathbf{V})$ (because $\mu_i(\mathbf{V}) \leq \sigma_{m_i}(\mathbf{V})$ by definition of observation). Therefore $\sup_n \{\mu_n(\mathbf{V})\} = \mu(\mathbf{V}) \leq \sigma(\mathbf{V})$. If μ is maximal, then $\sigma = \mu$. \square

Theorem 22 (Greatest Limit Distribution). $\text{Lim}(m)$ has a greatest element, which we indicate by $\llbracket m \rrbracket$.

Proof. The proof of both existence and uniqueness of maximal elements relies on Lemma 21. Let us explicitly show uniqueness. Let $\mu \in \text{Lim}(m)$ be maximal. Given any $\rho \in \text{Lim}(m)$, we prove that $\rho \leq \mu$. Let $\langle r_n \rangle_{n \in \mathbb{N}}$ be a sequence from m such that $\langle r_n \rangle_{n \in \mathbb{N}} \Downarrow \rho$. By Lemma 21, $\forall r_n$ there is a \Rightarrow -sequence from r_n which has limit μ . Therefore $\forall \mathbf{V} \in \mathcal{V}$, $\forall n$, $\rho_n(\mathbf{V}) \leq \mu(\mathbf{V})$, hence $\rho(\mathbf{V}) \leq \mu(\mathbf{V})$. If ρ is maximal, $\rho = \mu$. \square

Theorem 23 (Adequacy). If $m \Rightarrow^* s$, then $\llbracket m \rrbracket = \llbracket s \rrbracket$.

Proof. Observe first that $\llbracket s \rrbracket \in \text{Lim}(m)$, hence $\llbracket s \rrbracket \leq \llbracket m \rrbracket$. By Lemma 21, it holds that $\llbracket m \rrbracket \in \text{Lim}(s)$, hence $\llbracket m \rrbracket \leq \llbracket s \rrbracket$. \square

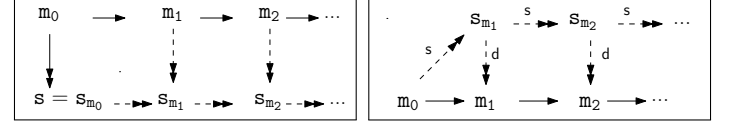


Figure 4: Proof of Main Lemma

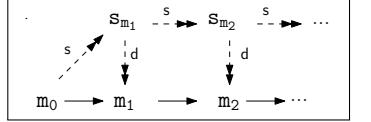


Figure 5: Surface evaluation

VII. ASYMPTOTIC STANDARDIZATION

In this section, we focus on \mathcal{V}_{\sim} as set of observations, which is the most natural choice in a CbV setting, in particular if we want to evaluate *programs*, i.e., closed terms.

We proved, in Thm. 22, that each m has a unique maximal limit distribution $\llbracket m \rrbracket$. Now we address the question: is there a reduction strategy which is guaranteed to converge to $\llbracket m \rrbracket$? We show that surface evaluation provides such a strategy; indeed, *any* limit distribution in $\text{Lim}(m)$ can be reached by surface evaluation (Thm. 26). This result of *asymptotic completeness* is the main technical contribution of the section.

Following the notation introduced in VI-A3, we denote by \mathbf{V} the set $\{W \in \mathcal{V} \mid W = \beta_v V\}$. We observe that:

Fact 24. Let $M \xrightarrow{d} m$, then m has form $[P]$ and $M = \beta_v P$; M is a value if and only if P is a value.

As a consequence of the previous fact, we have

Lemma 25. If $m \xrightarrow{d} s$ then $\mu(\mathcal{V}) = \sigma(\mathcal{V})$, and $\mu(\mathbf{V}) = \sigma(\mathbf{V})$, for each $\mathbf{V} \in \mathcal{V}_{\sim}$.

We write $m \xrightarrow{\infty} \mu$ (resp. $m \xrightarrow{\infty} \mu$) if there is a sequence $\langle m_n \rangle_{n \in \mathbb{N}}$ such that all steps $m_i \Rightarrow m_{i+1}$ are surface (resp. left) reductions and $\langle m_n \rangle_{n \in \mathbb{N}} \Downarrow \mu$. Remember that given m , we write $\llbracket m \rrbracket$ for the unique maximal element of $\text{Lim}(m)$, and $m \xrightarrow{\infty} \mu$ if there exists a \Rightarrow -sequence from m which converges to μ .

We now prove asymptotic completeness for surface evaluation. We exploit finitary standardization (Thm. 12) and *extend it to the limit*. In the proof, it is essential the fact that \xrightarrow{d} -steps preserve the distributions (Lemma 25).

Theorem 26 (Asymptotic Completeness of Surface Reduction). $m \xrightarrow{\infty} \mu$ if and only if $m \xrightarrow{\infty} \mu$.

Proof. We prove that $m \xrightarrow{\infty} \mu$ implies $m \xrightarrow{\infty} \mu$ (the other direction holds by definition). Assume $\langle m_n \rangle_{n \in \mathbb{N}} \Downarrow \mu$, with $m = m_0$. As illustrated in Fig. 5, we build a sequence $\langle s_{m_n} \rangle$ such that $m_0 = s_{m_0}$ and $\forall i$ ($s_{m_i} \xrightarrow{*} s_{m_{i+1}}$ and $s_{m_{i+1}} \xrightarrow{d} m_{i+1}$). If $i = 0$, by Thm. 12 it exists s_{m_1} such that $m_0 = s_{m_0} \xrightarrow{*} s_{m_1} \xrightarrow{d} m_1$. We then proceed by induction: for each $i > 0$, we apply

Thm. 12 to the sequence $s_{m_i} \xrightarrow{\text{d}}^* m_i \Rightarrow m_{i+1}$, and obtain the multidistribution $s_{m_{i+1}}$ such that $s_{m_i} \xrightarrow{\text{d}}^* s_{m_{i+1}}$ and $s_{m_{i+1}} \xrightarrow{\text{d}}^* m_{i+1}$, as wanted. The concatenation of all segments $s_{m_0} \xrightarrow{\text{d}}^* s_{m_1}, \dots, s_{m_i} \xrightarrow{\text{d}}^* s_{m_{i+1}}, \dots$ is a $\xrightarrow{\text{d}}$ -sequence. Let σ be its limit. By Lemma 25 and the fact that $s_{m_i} \xrightarrow{\text{d}}^* m_i$, we have $\sigma_{m_i}(\mathbf{V}) = \mu_i(\mathbf{V})$, for each $\mathbf{V} \in \mathcal{V}_\sim$. We conclude $\sigma = \mu$ because $\forall i$:

- 1) $\sigma_{m_i}(\mathbf{V}) = \mu_i(\mathbf{V}) \leq \mu(\mathbf{V})$, therefore $\sigma(\mathbf{V}) \leq \mu(\mathbf{V})$.
- 2) $\mu_i(\mathbf{V}) = \sigma_{m_i}(\mathbf{V}) \leq \sigma(\mathbf{V})$, therefore $\mu(\mathbf{V}) \leq \sigma(\mathbf{V})$.

□

Remark 27. We observe that completeness of surface evaluation (Thm. 26) is specific to convergence w.r.t. \mathcal{V}_\sim and $\{\mathcal{V}\}$ (the most natural set of observations in CbV). It does not necessarily hold if we evaluate w.r.t. other sets of observations, such as normal forms, where deep steps may be needed. Consider, for example, the term $\lambda z. II \xrightarrow{\text{d}} \lambda z. I$. To define a complete strategy w.r.t. $\mathcal{N}_{\{\}}^{\text{d}}$ demands to refine the approach.

A. Surface and Left Evaluation

We are now equipped to tackle the goal of this section, namely the existence of a strategy to find the greatest limit distribution of a program.

Since our aim is to reach the greatest limit, it makes sense to reduce "whenever is possible", and use the full lifting \Rightarrow (Def. IV-A3). The reason is easy to see. Consider for example $m = [\frac{1}{2}\Delta\Delta, \frac{1}{2}II]$, which has greatest limit $\llbracket m \rrbracket = \{\mathbf{I}^{\frac{1}{2}}\}$. We observe that a \Rightarrow -sequence from m may very well keep reducing only the diverging term $\Delta\Delta$ and never reach $\llbracket m \rrbracket$. The reduction \Rightarrow , instead, forces the reduction of each term which is not in normal form for \rightarrow .

Lemma 28. Let ρ be maximal among the limit distribution of all \Rightarrow -sequences from m . Then $\rho = \llbracket m \rrbracket$.

Proof. Obviously, $\rho \in \text{Lim}(m)$. It is straightforward to check that if μ is the limit of a \Rightarrow -sequence, then there is a \Rightarrow -sequence, whose limit is greater or equal to μ . □

We write $\xrightarrow{\text{s}}$ (resp. $\xrightarrow{\text{l}}$) for the full lifting of $\xrightarrow{\text{s}}$ (resp. $\xrightarrow{\text{l}}$). Observe that given m , there is only one $\xrightarrow{\text{l}}$ -sequence. We use the letters $l = \langle l_n \rangle_{n \in \mathbb{N}}$, $s = \langle s_n \rangle_{n \in \mathbb{N}}$, $t = \langle t_n \rangle_{n \in \mathbb{N}}$ to indicate (infinite) reduction sequences. We say that m is *closed* if it is a multidistribution on closed terms.

Proposition 29 (Left Evaluation). Let m be closed.

- 1) Let s, t be $\xrightarrow{\text{s}}$ -sequences from m ; $s \Downarrow \mu$ if and only if $t \Downarrow \mu$.
- 2) Let s be any $\xrightarrow{\text{s}}$ -sequence from m , and l the $\xrightarrow{\text{l}}$ -sequences from m . Then $s \Downarrow \mu$ if and only if $l \Downarrow \mu$.

Proof. [15], Sec.6, studies a CbV probabilistic λ -calculus with surface reduction ($\Lambda_{\oplus}^{\text{weak}}$) and proves using a diamond property that if $m \xrightarrow{\text{s}}^k m_k$ and $m \xrightarrow{\text{s}}^k r_k$ (both sequence have k steps) then $\forall V \in \mathcal{V}$, $\mu_k(V) = \rho_k(V)$. Claim (1.) easily follows.

Claim (2.) follows from (1.) and from Lemma 11, which implies that if $M \xrightarrow{\text{s}} n$ is *closed*, we can always choose a surface step which is a $\xrightarrow{\text{l}}$ -step. □

Putting all elements together, we have proved that the limit distribution of any $\xrightarrow{\text{s}}$ -sequence from m is $\llbracket m \rrbracket$. In particular, $\llbracket m \rrbracket$ is also the limit distribution of the $\xrightarrow{\text{l}}$ -sequence from m .

Theorem 30. For m closed, the following hold.

- 1) Let s be any $\xrightarrow{\text{s}}$ -sequence from m . Then $s \Downarrow \llbracket m \rrbracket$.
- 2) Let l be the $\xrightarrow{\text{l}}$ -sequence from m . Then $l \Downarrow \llbracket m \rrbracket$.
- 3) The sets $\{\rho \mid m \xRightarrow{\infty} \rho\}$, $\{\rho \mid m \xrightarrow{\text{s}}^{\infty} \rho\}$, and $\{\rho \mid m \xrightarrow{\text{l}}^{\infty} \rho\}$ have the same greatest element, which is $\llbracket m \rrbracket$.

While left reduction is not standard for finite sequences (as Example 13 shows), still is able to reach $\llbracket m \rrbracket$, if we only evaluate programs, i.e., closed terms. Thm. 30 justifies (a posteriori!) the use of the leftmost-outermost strategy in the literature of probabilistic λ -calculus: left evaluation actually produces the best asymptotic result. However, it is not the only strategy to achieve this: any $\xrightarrow{\text{s}}$ -sequence will.

VIII. SUMMING-UP AND OVERVIEW

The definition of reduction in $\Lambda_{\oplus}^{\text{cbv}}$ is based on two components: the β_v -rule and the \oplus -rule. We stress that only the \oplus -step is constrained, while β_v is inherited "as is" from the λ -calculus. The β_v -rule is allowed in *all* contexts, while the \oplus -rule is disabled in a function body. This avoids confusion between duplicating a function which performs a choice, and duplicating the choice, that is the core of confluence failure. It is then natural to expect that the fine control on duplication which is offered by linear logic could be beneficial.

In Sec. IX we apply the methods and tools which we have developed to study $\Lambda_{\oplus}^{\text{cbv}}$ to define a probabilistic linear calculus $\Lambda_{\oplus}^{\text{l}}$ which extends with a probabilistic choice Simpson's linear λ -calculus [31]. This is a result of interest in its own, but also evidence that our approach is robust, as it transfers well to other probabilistic calculi. In Sec. X we then define a call-by-name probabilistic calculus, $\Lambda_{\oplus}^{\text{cbn}}$, and we show that similar results to the ones we have established for $\Lambda_{\oplus}^{\text{cbv}}$ hold.

As we will see, the three calculi follow the same pattern: the \oplus -reduction (and *only this* reduction) is restricted to surface contexts. In Sec. XI we discuss how the three calculi relate.

IX. PROBABILISTIC LINEAR LAMBDA CALCULUS

Λ^{l} [31] is an untyped linear λ -calculus which is closely based on linear logic. Abstraction is refined into linear abstraction $\lambda x.M$ and non-linear abstraction $\lambda!x.M$, which allows duplication of the argument. The argument of $\lambda!x.M$ is required to be suspended as thunk $!N$, that corresponds to the $!$ -box of linear logic. In this section, we define a probabilistic linear λ -calculus $\Lambda_{\oplus}^{\text{l}}$ by extending Λ^{l} with an operator \oplus .

A. Syntax of $\Lambda_{\oplus}^{\text{l}}$

1) *The language:* Raw terms M, N, \dots are built up from a countable set of variables x, y, \dots according to the grammar:

$$M ::= x \mid !M \mid \lambda x.M \mid \lambda!x.M \mid MM \mid M \oplus N \quad (\text{terms } \Lambda_{\oplus}^{\text{l}})$$

We say that x is affine (resp. linear) in M if x occurs free at most (resp. exactly) once in M , and moreover, the free occurrence of x does not lie within the scope of a $!$ operator.

A term M is affine (resp. linear) if for every subterm $\lambda x.P$ of M , x is so in P . Henceforth, we consider affine terms only.

Contexts (\mathbf{C}) and surface contexts (\mathbf{S}) are generated by the grammars:

$$\begin{aligned} \mathbf{C} &::= \square \mid \mathbf{M}\mathbf{C} \mid \mathbf{C}\mathbf{M} \mid \lambda x.\mathbf{C} \mid \lambda!x.\mathbf{C} \mid \mathbf{C} \mid \mathbf{C} \oplus \mathbf{M} \mid \mathbf{M} \oplus \mathbf{C} \quad (\text{contexts}) \\ \mathbf{S} &::= \square \mid \mathbf{M}\mathbf{S} \mid \mathbf{S}\mathbf{M} \mid \lambda x.\mathbf{S} \mid \lambda!x.\mathbf{S} \quad (\text{surface c.}) \end{aligned}$$

where \square denotes the *hole* of the term context. Observe that a *surface context* is defined in a different way than in IV-A. Here it expresses the fact that a *surface redex cannot occur in the scope of a $!$ operator* (nor in the scope of a \oplus).

2) *Reductions*: We follow the same pattern as for $\Lambda_{\oplus}^{\text{cbv}}$. The beta rules \mapsto_{β} are given in Fig. 7. The probabilistic rules $\mapsto_{l\oplus}, \mapsto_{r\oplus}$ are as in Fig. 1. The reduction steps are in Fig. 6: the β -rule is closed under general context, while the \oplus -rules are closed under surface contexts. The β -rules also can be restricted to the closure under surface contexts, as shown in Fig. 6. A \rightarrow -step is *deep* (written \xrightarrow{d}) if it is *not* surface. The *lifting* of the relation $\rightarrow: \Lambda_{\oplus}^! \times \text{MDST}(\Lambda_{\oplus}^!)$ to a binary relation on $\Rightarrow \text{MDST}(\Lambda_{\oplus}^!)$ is defined as in Fig. 3.

Beta Step \rightarrow_{β}	Surface Beta Step \xrightarrow{s}_{β}
$\frac{M \mapsto_{\beta} M'}{\mathbf{C}(M) \rightarrow_{\beta} [\mathbf{C}(M')]}$	$\frac{M \mapsto_{\beta} M'}{\mathbf{S}(M) \xrightarrow{s}_{\beta} [\mathbf{S}(M')]}$
	(Surface) \oplus-Step $\rightarrow_{\oplus} := \xrightarrow{s}_{\oplus}$
	$\frac{M \oplus N \mapsto_{l\oplus} M \quad M \oplus N \mapsto_{r\oplus} N}{\mathbf{S}(M \oplus N) \xrightarrow{s}_{\oplus} [\frac{1}{2}\mathbf{S}(M), \frac{1}{2}\mathbf{S}(N)]}$
Reduction Step \rightarrow	Surface Reduction Step \xrightarrow{s}
$\rightarrow := \rightarrow_{\beta} \cup \rightarrow_{\oplus}$	$\xrightarrow{s} := \xrightarrow{s}_{\beta} \cup \xrightarrow{s}_{\oplus}$

Figure 6: Reduction Steps

$(\lambda x.M)N \mapsto_{\beta} M[N/x]$	$(\lambda!x.M)!N \mapsto_{\beta} M[N/x]$
---	--

Figure 7: β reduction rules for $\Lambda_{\oplus}^!$

Remark 31. To limit notations for reductions and contexts, we use the same as for $\Lambda_{\oplus}^{\text{cbv}}$, clearly the meaning is different.

B. $\Lambda_{\oplus}^!$ is a conservative extension of $\Lambda^!$

As in IV-B, we denote by \rightarrow_{β} both the reduction in $\Lambda^!$ and the β reduction in $\Lambda_{\oplus}^!$; we prove that $(\Lambda_{\oplus}^!, \Rightarrow_{\beta})$ is a conservative extension of $(\Lambda^!, \rightarrow_{\beta})$.

Definition 32 (Translation). $(\cdot)_! : \Lambda_{\oplus}^! \rightarrow \Lambda^!$ is defined in the following way, where z is a fixed fresh variable

$$\begin{aligned} (x)_! &= x & (\lambda x.M)_! &= \lambda x.(M)_! \\ (M \oplus N)_! &= z \cdot ! (M)_! \cdot ! (N)_! & (\lambda!x.M)_! &= \lambda!x.(M)_! \\ (MN)_! &= (M)_! (N)_! & (!M)_! &= ! (M)_! \end{aligned}$$

Note that the translation of terms of the form $M \oplus N$ is designed so to preserves surface reduction.

Proposition 33 (Simulation). *The translation $(\cdot)_!$ is sound and complete, and preserve surface reduction. Let $\rightarrow_b \in \{\rightarrow_{\beta}, \xrightarrow{s}_{\beta}\}$. For $M \in \Lambda_{\oplus}^!$, the following hold*

- 1) $M \rightarrow_b [N]$ implies $(M)_! \rightarrow_b (N)_!$.
- 2) $(M)_! \rightarrow_b P$ implies that exists (unique) $N \in \Lambda_{\oplus}^!$, with $N = (P)_!$ and $M \rightarrow_b [N]$.

The translation tells us that the reduction $[M] \Rightarrow_{\beta} [N]$ on $\Lambda_{\oplus}^!$ behaves as the reduction $(M)_! \rightarrow_{\beta} (N)_!$ on $\Lambda^!$.

C. *Confluence and Finitary Standardization for $\Lambda_{\oplus}^!$*

The following properties hold for $\Lambda^!$ [31].

Theorem (Simpson 05). The following hold in $\Lambda^!$.

- 1) **Confluence.** \rightarrow_{β} is confluent.
- 2) **Surface Standardization.** If $M \rightarrow_{\beta}^* N$ then exists R such that $M \xrightarrow{s}_{\beta}^* R$ and $R \xrightarrow{d}^* N$.

We show, using the methods developed for $\Lambda_{\oplus}^{\text{cbv}}$ and the translation in Def. 32, that the same properties hold for $\Lambda_{\oplus}^!$.

1) *Confluence*: We follow the same approach as in Sec. V-A. In fact, we already have most of the building blocks for the proof. Observe that Lemma 5 is general enough to apply also to binary relations on $\text{MDST}(\Lambda_{\oplus}^!)$.

Lemma 34. 1) *The reduction \Rightarrow_{\oplus} is diamond.*

2) *The reduction \Rightarrow_{β} is confluent.*

3) *The reductions \Rightarrow_{β} and \Rightarrow_{\oplus} commute.*

Proof. The proof of 1) and 2) is as for Lemmas 7 and 6; 3) is proved using Lemma 5, by induction on the term. \square

By Hindley-Rosen Lemma, we obtain

Theorem 35. *The reduction \Rightarrow of $\Lambda_{\oplus}^!$ is confluent.*

2) *Surface standardization*:

Proposition 36 (Finitary Surface Standardization). *In $\Lambda_{\oplus}^!$, if $m \Rightarrow^* n$ then exists r such that $m \xrightarrow{s}^* r$ and $r \xrightarrow{d}^* n$.*

The proof (see [16]) builds on the analogous result for $\Lambda^!$ and is straightforward.

D. *Asymptotic behaviour*

Normal forms are defined as in IV-A2e; we denote by $\mathcal{N}^!$ the set of \rightarrow -normal forms, and by $\mathcal{S}^!$ the set of the *surface normal forms* (i.e. the \xrightarrow{s} -normal forms). Clearly $\mathcal{N}^! \subsetneq \mathcal{S}^!$. We define $\mathcal{N}_{\{\}}^! := \{\{M\}, M \in \mathcal{N}^!\}$, and $\mathcal{S}_{\sim}^!$ as the set of all events $\mathbf{R} := \{S \in \mathcal{S}^! \mid S =_{\beta} R\}$.

a) *Observations*: A set of *observations* for $(\Lambda_{\oplus}^!, \Rightarrow)$ is defined in the same way as that for $(\Lambda_{\oplus}, \Rightarrow)$ (Def. 17).

Proposition 37. *Each of the following sets $\{\mathcal{N}^!\}$, $\{\mathcal{S}^!\}$, $\mathcal{N}_{\{\}}^!$, $\mathcal{S}_{\sim}^!$, is a set of observations for $(\Lambda_{\oplus}^!, \Rightarrow)$.*

b) *Limit distributions and evaluation*: Once we fix a set of observations Obs for $(\Lambda_{\oplus}^!, \Rightarrow)$, the definition of evaluation and limit distribution, and the notations $\langle m_n \rangle_{n \in \mathbb{N}} \Downarrow \rho$, $m \Rightarrow^{\infty} \rho$ and $\text{Lim}(m)$ are as in Def. 18. We already observed that Thm. 22 and 23 only depends on confluence, and on the definition of observations; therefore both hold.

Theorem 38. *For any choice of Obs, $\Lambda_{\oplus}^!$ has the properties:*

- $\text{Lim}(m)$ has a greatest element, which we indicate as $\llbracket m \rrbracket$.
- If $m \Rightarrow^* s$, then $\llbracket m \rrbracket = \llbracket s \rrbracket$.

c) *Asymptotic Standardization*: For the rest of the section we focus on $\text{Obs} := S_{\sim}^!$. We have established that for each $m \in \Lambda_{\oplus}^!$, $\text{Lim}(m)$ has a unique maximal element $\llbracket m \rrbracket$. We now want to have a strategy to find $\llbracket m \rrbracket$. Surface reduction plays that role. We use the following fact, which is easy to verify.

Fact 39. *Let $M \xrightarrow{d} n$. Then 1) n is of the form $[N]$, and $M =_{\beta} N$; 2) $M \in S^!$ if and only if $N \in S^!$.*

Theorem 40 (Asymptotic Completeness). *In $\Lambda_{\oplus}^!$ it holds that $m \Rightarrow^{\infty} \mu$ if and only if $m \xrightarrow{s}^{\infty} \mu$.*

Proof. As for Thm. 26, now using Fact 39 and Prop. 36. \square

Similarly to Sec. VII, we can establish that any (infinitary) \xrightarrow{s} -sequences from m converges precisely to $\llbracket m \rrbracket$, where \xrightarrow{s} indicate the full lifting of the relation $\xrightarrow{s} \subseteq \Lambda_{\oplus}^! \times \text{MDST}(\Lambda_{\oplus}^!)$.

Theorem 41 (Surface Evaluation). *Let $s = \langle s_n \rangle_{n \in \mathbb{N}}$ be any \xrightarrow{s} -sequences from m . It holds that $s \Downarrow \llbracket m \rrbracket$.*

X. CALL-BY-NAME CALCULUS $\Lambda_{\oplus}^{\text{cbn}}$

We show that results similar to those for $\Lambda_{\oplus}^{\text{cbv}}$ hold for a CbN calculus, denoted $\Lambda_{\oplus}^{\text{cbn}}$. We could adapt all the proofs, but now we prefer to follow a different way. Once we take the point of view of linear logic, we have a roadmap to CbN via Girard's translation of intuitionistic into linear logic. More precisely, we rely on recent work [12], [18] which expresses those translations in *untyped* λ -calculus. We exploit the faithful nature of the translation to transfer both confluence and standardization from $\Lambda_{\oplus}^!$ to $\Lambda_{\oplus}^{\text{cbn}}$, essentially for free.

A. Syntax of $\Lambda_{\oplus}^{\text{cbn}}$

We write $\Lambda_{\oplus}^{\text{cbn}}$ for the set of terms Λ_{\oplus} equipped with the reduction relation \Rightarrow defined below.

1) *The language: Terms and contexts (C)* are the same as in $\Lambda_{\oplus}^{\text{cbv}}$. *Surface contexts (S)* are generated by the grammar:

$$S ::= \square \mid \lambda x. S \mid SM \quad (\text{cbn surface contexts})$$

2) *Reductions*: The β -rule \mapsto_{β} is as in the CbN λ -calculus (Fig. 8). The probabilistic rules $\mapsto_{l\oplus}, \mapsto_{r\oplus}$ are as in Fig. 1.

$$(\lambda x. M)N \mapsto_{\beta} M[N/x]$$

Figure 8: Beta Reduction Rule for $\Lambda_{\oplus}^{\text{cbn}}$

Reduction steps $\rightarrow, \rightarrow_{\beta}, \rightarrow_{\oplus} \subseteq \Lambda_{\oplus} \times \text{MDST}(\Lambda_{\oplus})$ and *surface reduction steps* $\xrightarrow{s}, \xrightarrow{s}_{\beta}, \xrightarrow{s}_{\oplus} \subseteq \Lambda_{\oplus} \times \text{MDST}(\Lambda_{\oplus})$ are defined in Fig. 6, following the usual pattern. By definition of surface context, a reduction step is *surface* if it *does not occur in argument position* (nor in the scope of \oplus).

The *lifting* of $\rightarrow \subseteq \Lambda_{\oplus} \times \text{MDST}(\Lambda_{\oplus})$ to a binary relation \Rightarrow on $\text{MDST}(\Lambda_{\oplus}^{\text{cbn}})$ is defined as in Fig. 3. The full lifting \Rightarrow is defined as in IV-A3.

3) *Normal Forms*: We denote by \mathcal{N}^{cbn} the set of \rightarrow -normal forms, and by \mathcal{S}^{cbn} the set of the *surface normal forms* (i.e. the \xrightarrow{s} -normal forms). Clearly $\mathcal{N}^{\text{cbn}} \subsetneq \mathcal{S}^{\text{cbn}}$.

Let us extend to $\Lambda_{\oplus}^{\text{cbn}}$ the notion of *head normal form*. Head reduction \xrightarrow{h} is the closure of both the β and the probabilistic

rules under head context **H**, which is defined by the following grammar

$$\mathbf{H} ::= \lambda x. \mathbf{H} \mid \mathbf{K} \quad \mathbf{K} ::= \square \mid \mathbf{K}M \quad (\text{head contexts})$$

Observe that $\xrightarrow{h} \subsetneq \xrightarrow{s}$ (for example, the reduction $(\lambda x. (\lambda y. y)P)Q \xrightarrow{s} (\lambda x. P)Q$ is not a head reduction). However, the two relations have the same normal forms. Let us write \mathcal{H} for the set of *head normal forms*. It is easy to verify that a head normal form has no \xrightarrow{s} -redex, and conclude:

$$\mathcal{S}^{\text{cbn}} = \mathcal{H}$$

4) $\Lambda_{\oplus}^{\text{cbn}}$ to $\Lambda_{\oplus}^!$: In [18], the translation from Λ^{cbn} into a linear λ -calculus is proved sound and complete. We follow their work to define a similar translation $(\cdot)_n : \Lambda_{\oplus}^{\text{cbn}} \rightarrow \Lambda_{\oplus}^!$:

$$\begin{aligned} (x)_n &= x & (\lambda x. M)_n &= \lambda!x. (M)_n \\ (MN)_n &= (M)_n!(N)_n & (M \oplus N)_n &= (M)_n \oplus (N)_n \\ ([p_i M_i \mid i \in I])_n &= [p_i (M_i)_n \mid i \in I] \end{aligned}$$

The following extends to the probabilistic setting an analogous result proved in [18]. Observe that, with a slight abuse of notation, reductions in the two calculi are denoted in the same way, the meaning being clear from the context.

Proposition 42 (Simulation). *The translation $(\cdot)_n$ is sound and complete; it preserves surface reduction and surface normal forms. Let $\rightarrow_b \in \{\rightarrow_{\beta}, \xrightarrow{s}_{\beta}\}$. For $M \in \Lambda_{\oplus}^{\text{cbn}}$, it holds that:*

- 1) if $M \rightarrow_b n$ then $(M)_n \rightarrow_b (n)_n$;
- 2) if $(M)_n \rightarrow_b s$ then $\exists! n$ such that $s = (n)_n$ and $M \rightarrow_b n$;
- 3) $M \in \mathcal{H}$ if and only if $(M)_n \in S^!$.

Proof. The proof is in [16]. \square

B. Confluence and Finitary Standardization for $\Lambda_{\oplus}^{\text{cbn}}$

The fact that surface reduction is preserved by $(\cdot)_n$ is crucial to transfer the standardization result from $\Lambda_{\oplus}^!$ to $\Lambda_{\oplus}^{\text{cbn}}$. We show that via translation, $\Lambda_{\oplus}^{\text{cbn}}$ inherits both the confluence and the surface standardization property from $\Lambda_{\oplus}^!$.

Theorem 43 (Confluence). *The relation \Rightarrow_{cbn} is confluent.*

Proof. From Thm. 35, using back-and-forth Thm 42. \square

Theorem 44 (Finitary Surface standardization). *If $m \Rightarrow^* n$ then exists r such that $m \xrightarrow{s}^* r$ and $r \xrightarrow{d}^* n$.*

Proof. From Thm. 36, by using back-and-forth Thm 42, and the fact that the translation preserves surface reduction. \square

In the classical λ -calculus, the standardization property (Barendregt, Th. 11.4.7) says that every reduction sequence can be ordered in such a way to perform first only left β -redexes, reading the term from left to right, and then internal ones (a redex is internal if it is not the leftmost one).

In $\Lambda_{\oplus}^{\text{cbn}}$ this notion of standardization fails, as the following example (which we take from [20]) shows.

Example 45. *In each step, we underline the redex. Consider $[(\lambda x. I(y \oplus z))I] \Rightarrow [(\lambda x. y \oplus z)I] \Rightarrow [\frac{1}{2}(\lambda x. y)I, \frac{1}{2}(\lambda x. z)I] \Rightarrow [\frac{1}{2}y, \frac{1}{2}(\lambda x. z)I]$, where only the last step reduces a left redex.*

If we perform the left redex first, we have $[(\lambda x. I(y \oplus z))I] \Rightarrow [I(y \oplus z)]$, from which $[\frac{1}{2}y, \frac{1}{2}(\lambda x. z)I]$ cannot be reached.

A consequence of standardization is that M has a head normal form iff the \xrightarrow{h} -sequence from M terminates. In the following section we retrieve an analogue of this result.

C. Asymptotic behaviour

We denote by \mathcal{H}_\sim the set of head normal forms up to the equivalence $=_\beta$, and we define $\mathcal{N}_\sim^{\text{cbn}} = \{\{M\}, M \in \mathcal{N}^{\text{cbn}}\}$.

a) *Observations*: Observations are defined as in Def. 17.

Proposition 46. *Each of the following is a set of observations for $\Lambda_\oplus^{\text{cbn}}$: $\{\mathcal{H}\}$, $\{\mathcal{N}^{\text{cbn}}\}$, \mathcal{H}_\sim , $\mathcal{N}_\sim^{\text{cbn}}$.*

b) *Convergence and Limit distributions*: Once we fix a set of observations Obs for $\Lambda_\oplus^{\text{cbn}}$, the definition of convergence and limit distribution are as in Def. 18. We observe that Theorems 22 and 23 both hold. Hence in particular

Theorem 47. *For any choice of Obs, the following holds in $\Lambda_\oplus^{\text{cbn}}$: given \mathfrak{m} , $\text{Lim}(\mathfrak{m})$ has a greatest element $\llbracket \mathfrak{m} \rrbracket$.*

We now study the notion of convergence induced by choosing head normal forms as outcome, i.e. $\text{Obs} := \mathcal{H}_\sim$. Therefore, if $\rho \in \text{Lim}(\mathfrak{m})$, it holds that $\rho \in \text{MDST}(\mathcal{H}_\sim)$. The following results match the analogous results in $\Lambda_\oplus^!$ (Thm. 40 and 41).

Theorem 48. *Let $\text{Obs} := \mathcal{H}_\sim$. For every multidistribution \mathfrak{m} :*

- $\mathfrak{m} \xRightarrow{s} \mu$ if and only if $\mathfrak{m} \xRightarrow{s} \mu$.
- If $\langle \mathfrak{s}_n \rangle_{n \in \mathbb{N}}$ is a \xRightarrow{s} -sequences of full surface reductions from \mathfrak{m} , then $\langle \mathfrak{s}_n \rangle_{n \in \mathbb{N}} \Downarrow \llbracket \mathfrak{m} \rrbracket$.

Similarly to Prop. 29, it is not hard to prove that in $\Lambda_\oplus^{\text{cbn}}$, \xRightarrow{s} satisfies a diamond property in the sense of [15], and hence all \xRightarrow{s} -sequences from \mathfrak{m} converge to the same limit distribution.

Let $\langle \mathfrak{s}_n \rangle_{n \in \mathbb{N}}$ be the \xrightarrow{h} -sequences of full head reductions from \mathfrak{m} . It holds that $\langle \mathfrak{s}_n \rangle_{n \in \mathbb{N}} \Downarrow \llbracket \mathfrak{m} \rrbracket$.

Once again, this justifies a posteriori the choice of head reduction in probabilistic CbN (such as [13]). Observe that we follow the same reasoning as in the case of $\Lambda_\oplus^{\text{cbv}}$ (with \mathcal{V}_\sim as set of observations). First we proved that surface reduction is sufficient to reach the greatest limit distribution, then we observed that in particular left reduction can be chosen. There is a close parallelism between $\Lambda_\oplus^{\text{cbv}}$ and $\Lambda_\oplus^{\text{cbn}}$: similar results hold if we consider as set of observations \mathcal{V}_\sim and \mathcal{H}_\sim respectively.

XI. CONCLUSION AND DISCUSSION

A. Summary

In this paper we design two probabilistic extensions of respectively the CbV and CbN λ -calculus, $\Lambda_\oplus^{\text{cbv}}$ and $\Lambda_\oplus^{\text{cbn}}$, which we propose as foundational calculi for probabilistic computation. Both calculi enjoy *confluence* and *standardization*, in an extended way. Namely, first we prove both properties for the finite sequences, exploiting classical methods, then we extend these properties to the limit, developing new sophisticated proof methods. In particular, we prove the

uniqueness of the (maximal) result, parametrized by the notion of *set of observations*, and that the asymptotic extension of surface standardization supplies a family of *complete reduction strategies* which are guaranteed to reach the best result. The two calculi have a common root in the linear λ -calculus $\Lambda_\oplus^!$, which is both a technical tool and a calculus of interest in its own, in which a fine control of the interaction between copying and choice is possible.

In all three calculi, β -reduction is unconstrained; hence for each calculus, its restriction to only β -reduction exactly gives the usual corresponding (CbN, CbV, or linear) λ -calculus; this is not the case for extensions in which a strategy is fixed.

New proof methods include the *asymptotic extension of surface standardization* (Thm. 26), and the use of a *translation to transfer standardization* properties, namely from $\Lambda_\oplus^!$ to $\Lambda_\oplus^{\text{cbn}}$. It is worth stressing a crucial element: the fact that the translation is sound, complete and preserves surface contexts is what allows us to transfer the results.

B. Discussion

a) *Relating the calculi (Girard's Translations)*: The key to understand how $\Lambda_\oplus^{\text{cbv}}$, $\Lambda_\oplus^{\text{cbn}}$, and $\Lambda_\oplus^!$ relate are the two Girard's translations which embed intuitionistic logic into linear logic, and which are well known to respectively correspond to CbN and CbV computations. Let us clarify this. Let us start from $\Lambda_\oplus^!$: the natural constraint to avoid copying the result of a choice is "no \oplus -reduction in the scope of $!$ " (i.e., inside a $!$ -box). Using the intuition provided by Girard's translations as a guide, the constraint above becomes respectively "no \oplus -reduction in the scope of a λ -abstraction" (in CbV) and "no \oplus -reduction in argument position" (in CbN). Our three notions of surface context express these three constraints.

The intuitive reasoning above can be formalized thanks to a recent line of work [12], [18], which internalizes the insights coming from linear logic and proof nets into a λ -syntax. The resulting calculus subsumes both CbN and CbV λ -calculi via Girard's translation. The idea of a system which subsumes both CbV and CbN had been already advocated and developed by Levy, via the Call-By-Push-Value paradigm [21]. And indeed, [12] can be seen as an untyped version of Levy's calculus. We leave to the future a comprehensive approach, where a probabilistic linear calculus is the metalanguage in which all the results are developed.

b) *On non-deterministic λ -calculi*: The finitary results we presented (namely, confluence and finitary surface standardization) also hold if the probabilistic choice is replaced by *non-deterministic* choice (just forget the coefficients). Asymptotic results, instead, are specific to probabilistic computation.

c) *$\Lambda_\oplus^!$ and quantum λ -calculi*: The fine control of duplication which $\Lambda_\oplus^!$ inherits from linear logic has made it an ideal base for quantum λ -calculi (such as [7], [6]). In those calculi, *surface reduction* is the key ingredient to allow for the coexistence of quantum bits with duplication and erasing. *No reduction* (not even β) is allowed in the scope of a $!$ operator. Our results show that β -reduction can be unrestricted, only measurement (the quantum analogue of \oplus) needs to be surface.

REFERENCES

- [1] P. Arrighi and G. Dowek. Lineal: A linear-algebraic lambda-calculus. *Logical Methods in Computer Science*, 13(1), 2017.
- [2] M. Avanzini, U. Dal Lago, and A. Yamada. On probabilistic term rewriting. In J. P. Gallagher and M. Sulzmann, editors, *Functional and Logic Programming - 14th International Symposium, FLOPS 2018, Nagoya, Japan, May 9-11, 2018, Proceedings*, volume 10818 of *Lecture Notes in Computer Science*, pages 132–148. Springer, 2018.
- [3] G. Bacci, R. Furber, D. Kozen, R. Mardare, P. Panangaden, and D. Scott. Boolean-valued semantics for the stochastic λ -calculus. In A. Dawar and E. Grädel, editors, *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018*, pages 669–678. ACM, 2018.
- [4] H. P. Barendregt. *The Lambda Calculus: Its Syntax and Semantics*, volume 103. North Holland, 1984.
- [5] J. Borgström, U. Dal Lago, A. D. Gordon, and M. Szymczak. A lambda-calculus foundation for universal probabilistic programming. In *Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming, ICFP 2016, Nara, Japan, September 18-22, 2016*, pages 33–46, 2016.
- [6] U. Dal Lago, C. Faggian, B. Valiron, and A. Yoshimizu. The geometry of parallelism: classical, probabilistic, and quantum effects. In G. Castagna and A. D. Gordon, editors, *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages, POPL 2017, Paris, France, January 18-20, 2017*, pages 833–845. ACM, 2017.
- [7] U. Dal Lago, A. Masini, and M. Zorzi. Confluence results for a quantum lambda calculus with measurements. *Electr. Notes Theor. Comput. Sci.*, 270(2):251–261, 2011.
- [8] U. Dal Lago and M. Zorzi. Probabilistic operational semantics for the lambda calculus. *CoRR*, abs/1104.0195, 2011.
- [9] U. de'Liguoro and A. Piperno. Non deterministic extensions of untyped lambda-calculus. *Inf. Comput.*, 122(2):149–177, 1995.
- [10] A. Di Pierro, C. Hankin, and H. Wiklicky. Probabilistic lambda-calculus and quantitative program analysis. *J. Log. Comput.*, 15(2):159–179, 2005.
- [11] A. Díaz-Caro and G. Martínez. Confluence in probabilistic rewriting. *Electr. Notes Theor. Comput. Sci.*, 338:115–131, 2018.
- [12] T. Ehrhard and G. Guerrieri. The bang calculus: an untyped lambda-calculus generalizing call-by-name and call-by-value. In J. Cheney and G. Vidal, editors, *Proceedings of the 18th International Symposium on Principles and Practice of Declarative Programming, Edinburgh, United Kingdom, September 5-7, 2016*, pages 174–187. ACM, 2016.
- [13] T. Ehrhard, M. Pagani, and C. Tasson. The computational meaning of probabilistic coherence spaces. In *Proceedings of the 26th Annual IEEE Symposium on Logic in Computer Science, LICS 2011, June 21-24, 2011, Toronto, Ontario, Canada*, pages 87–96. IEEE Computer Society, 2011.
- [14] T. Ehrhard and L. Regnier. The differential lambda-calculus. *Theor. Comput. Sci.*, 309(1-3):1–41, 2003.
- [15] C. Faggian. Probabilistic rewriting: On normalization, termination, and unique normal forms. In *International Conference on Formal Structures for Computation and Deduction, FSCD, 2019*. Available at <http://arxiv.org/abs/1804.05578>.
- [16] C. Faggian and S. Ronchi Della Rocca. Lambda calculus and probabilistic computation (extended version). Available at <http://arxiv.org/abs/1901.02853>.
- [17] N. D. Goodman, V. K. Mansinghka, D. M. Roy, K. Bonawitz, and J. B. Tenenbaum. Church: a language for generative models. In D. A. McAllester and P. Myllymäki, editors, *UAI 2008, Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence, Helsinki, Finland, July 9-12, 2008*, pages 220–229. AUA Press, 2008.
- [18] G. Guerrieri and G. Manzonetto. The bang calculus and the two Girard's translations. In *to appear in EPTCS, International Workshop on Linearity and Trends in Linear Logic and Applications*, 2018.
- [19] D. Koller, D. A. McAllester, and A. Pfeffer. Effective bayesian inference for stochastic programs. In B. Kuipers and B. L. Webber, editors, *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Innovative Applications of Artificial Intelligence Conference, AAAI 97, IAAI 97, July 27-31, 1997, Providence, Rhode Island, USA.*, pages 740–747. AAAI Press / The MIT Press, 1997.
- [20] T. Leventis. *Probabilistic lambda-theories*. Phd Thesis, Aix-Marseille Université, 2016. Available at <https://tel.archives-ouvertes.fr/tel-01427279v2/document>.
- [21] P. B. Levy. Call-by-push-value: A subsuming paradigm. In J. Girard, editor, *Typed Lambda Calculi and Applications, 4th International Conference, TLCA'99, L'Aquila, Italy, April 7-9, 1999, Proceedings*, volume 1581 of *Lecture Notes in Computer Science*, pages 228–242. Springer, 1999.
- [22] S. Marlow. *Parallel and Concurrent Programming in Haskell*. O'Reilly Media, 2013.
- [23] S. Park. A calculus for probabilistic languages. In *Proceedings of TLDI'03: 2003 ACM SIGPLAN International Workshop on Types in Languages Design and Implementation, New Orleans, Louisiana, USA, January 18, 2003*, pages 38–49, 2003.
- [24] S. Park, F. Pfenning, and S. Thrun. A probabilistic language based upon sampling functions. In M. Abadi, editor, *Conference Record of the 32nd Symposium on Principles of Programming Languages (POPL'05)*, pages 171–182. Long Beach, California, Jan. 2005. ACM Press.
- [25] D. Pless and G. F. Luger. Toward general analysis of recursive probability models. In *UAI '01: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence, University of Washington, Seattle, Washington, USA, August 2-5, 2001*, pages 429–436, 2001.
- [26] G. D. Plotkin. Call-by-name, call-by-value and the lambda-calculus. *Theor. Comput. Sci.*, 1(2):125–159, 1975.
- [27] N. Ramsey and A. Pfeffer. Stochastic lambda calculus and monads of probability distributions. In J. Launchbury and J. C. Mitchell, editors, *Conference Record of POPL 2002: The 29th SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Portland, OR, USA, January 16-18, 2002*, pages 154–165. ACM, 2002.
- [28] S. Ronchi Della Rocca and L. Paolini. *The Parametric Lambda Calculus - A Metamodel for Computation*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2004.
- [29] N. Saheb-Djahromi. Probabilistic LCF. In J. Winkowski, editor, *Mathematical Foundations of Computer Science 1978, Proceedings, 7th Symposium, Zakopane, Poland, September 4-8, 1978*, volume 64 of *Lecture Notes in Computer Science*, pages 442–451. Springer, 1978.
- [30] D. S. Scott. Stochastic λ -calculi. an extended abstract. *Journal of Applied Logic*, 2014.
- [31] A. K. Simpson. Reduction in a linear lambda-calculus with applications to operational semantics. In J. Giesl, editor, *Term Rewriting and Applications, 16th International Conference, RTA 2005, Nara, Japan, April 19-21, 2005, Proceedings*, volume 3467 of *Lecture Notes in Computer Science*, pages 219–234. Springer, 2005.
- [32] L. Vaux. The algebraic lambda calculus. *Mathematical Structures in Computer Science*, 19(5):1029–1059, 2009.