On Higher-Order Probabilistic Verification via the Weighted Relational Model of Linear Logic

Ugo Dal Lago
University of Bologna, Italy
INRIA Sophia Antipolis, France

Guido Fiorillo
Université Claude Bernard Lyon 1, France

Paolo Pistone
Université Claude Bernard Lyon 1, France

Abstract

The problem of determining whether a probabilistic program terminates almost surely (i.e. with probability one) is undecidable, and actually Π^0_2-complete. For this reason, a growing literature has explored classes of programs for which this and related problems can be shown (semi-)decidable. In this work we consider the termination problem for the language of Probabilistic Higher-Order Recursion Schemes (PHORS). Using the weighted relational semantics of linear logic, we translate this problem into the computation of suitable generating functions associated with the program interpreted. This way, we establish the decidability of almost sure termination for a class of programs that extends Li et al.'s affine PHORS via a type discipline with bounded exponentials. To achieve this, we show that the generating functions for such programs are always algebraic, that is, solutions of polynomial equations, yielding an effective method to answer the termination problem.

2012 ACM Subject Classification Theory of computation → Linear logic; Theory of computation → Verification by model checking

Keywords and phrases Probabilistic λ-calculus, linear logic, weighted relational model, algebraic power series

Digital Object Identifier 10.4230/LIPIcs.LICS.2026.34

Related Version *Full Version*: `https://arxiv.org/abs/2604.27986v1` [15]

Funding *Ugo Dal Lago*: funded by ANR-24-CE48-5521-01.
Guido Fiorillo: funded by ANR-23-CPJ1-0054-01.
Paolo Pistone: funded by ANR-23-CPJ1-0054-01.

1 Introduction

Probabilistic Higher-Order Verification. Model checking is one of the most successful techniques for the verification of programs and systems [3, 14], and it has found applications in a wide range of areas, leading to the development of widely used tools such as SPIN [30] and PRISM [38]. While in its original formulation model checking focused on finite-state systems modeled by Kripke structures, its scope of application has gradually expanded over time, in directions that are often very different from one another. In this paper, we are particularly interested in two such directions.

The first concerns the model checking of systems whose evolution is intrinsically *probabilistic*. Along this line of research, the literature has produced a rich body of results, both positive and negative, regarding the decidability and tractability of the underlying verification problem (see [5] for a recent survey). It is worth noting that when the property to be verified is reachability or termination (appropriately generalized to a probabilistic setting), verification may remain decidable even beyond the finite-state case, as in, for example, recursive Markov chains [21] or pushdown automata [19].

But there is also another generalization of model checking, likewise obtained by considering a broader class of systems than those traditionally studied, namely so-called *higher-order model checking*. It has been known for about twenty years that this problem is decidable when the underlying specification is an arbitrary MSO formula and the program is an *higher-order recursion scheme* (HORS, [44, 45]), a language equivalent to the simply typed λ-calculus enriched with a fixpoint operator. Problems such as termination or reachability are therefore decidable for a broad class of programs encompassing higher-order types and recursion. These positive results have nonetheless revealed a certain degree of fragility: relatively minor extensions suffice to render the problem undecidable.

What, then, happens when one considers model-checking problems for probabilistic *and* higher-order programs? Recently, Kobayashi et al. [36] have introduced *probabilistic higher-order recursion schemes* (PHORS), a probabilistic variant of HORS, which can be seen as the natural randomized variation on the theme of HORS. Unfortunately, verifying natural probabilistic counterparts of reachability and termination properties is in general a much more difficult task and is undecidable for PHORS, contrary to what happens for HORS and recursive Markov chains. For example, in a probabilistic setting, a natural property is *almost sure termination* (AST), that is, termination with probability 1. Now, while termination is the quintessential *semi-decidable* property for a Turing-complete language (and is even *decidable* for HORS), AST is Π^0_2-complete in Turing-complete probabilistic languages [32], and its decidability fails for PHORS, already at order 2 [36]. For these reasons, the recent literature has focused on capturing sub-classes of PHORS for which AST and, possibly, the related *positive almost sure termination* (PAST) – the finiteness of the *expected* number of steps to termination – could be shown decidable, and thus possibly amenable to verification and model-checking techniques. Notably, while [36] established the decidability of AST for order-1 PHORS, Li et al. [42] established the decidability of both AST and PAST for the *affine* PHORS (styled PAHORS), i.e. the recursive programs which are allowed to use each of their inputs *at most* once during their reductions.

Despite the aforementioned results, the nature of the problem remains poorly understood, and a precise assessment of its computational difficulty is still lacking. For instance, it is currently unknown whether the decidability problem for PHORS remains Π^0_2-complete, or whether it can, while being undecidable, be placed lower in the arithmetical hierarchy. Similarly, while different methods (like intersection types, game semantics or automata theory) have been applied to achieve decidability, a general approach encompassing all these results has not yet been identified. We therefore believe that it is important to investigate the nature of this problem in greater depth, which is the primary goal of this paper.

Probabilistic Termination via Generating Functions. Estimating the probability of termination of higher-order programs is hard, because it demands to *count* probabilities over a typically *infinite* set of reductions, a problem that remains difficult even when these reductions follow some well-defined recursive pattern.

In fact, similar counting problems are common in the field of combinatorics, in which they are addressed by studying the algebraic and analytic properties of the corresponding *generating functions*, i.e. of the formal power series that are naturally associated with the sequence of numbers that one wishes to compute or estimate. A famous example is provided by the well-known *Catalan numbers* $C_n = \frac{1}{n+1}\binom{2n}{n}$, where C_n is the number of labelled binary trees with n nodes. This sequence induces the generating function $c(x) = \sum_{n=0}^{\infty} C_n x^n$, which can be characterized as the solution of a simple algebraic equation, namely $xc(x)^2 - c(x) + 1 = 0$. From a polynomial equation like this it is possible to extract information about the generating

function and the corresponding sequence that it would be otherwise very hard to compute directly. In our example, it allows us to deduce the closed expression $c(x) = \frac{1-\sqrt{1-4x}}{2x}$, hence providing a way to compute the values of $c(x)$ effectively.

Starting from Chomsky and Schützenberger's seminal work [10], generating functions have been widely applied in formal language theory. For example, it is well-known that for all regular languages L, the associated generating function $L(x) = \sum_{n=0}^{\infty} L_n x^n$, where L_n counts the words of L of length n, is *rational*, that is, it can be written as a fraction $L(x) = p(x)/q(x)$ of two polynomials. When L is context-free, instead, $L(x)$ is *algebraic*, i.e. it is the solution, like $c(x)$ above, of some polynomial equation $p(x, L(x)) = 0$.

Now, suppose $\mathcal{G}$ is a PHORS, i.e. an higher-order recursive probabilistic program, and that $\mathcal{G}$ makes, at each of its reduction steps, an unbiased choice between two different alternatives. Letting $\mathcal{G}_n$ be the number of *distinct* terminating reductions of $\mathcal{G}$ with exactly n steps, the power series $a_{\mathcal{G}}(z) = \sum_{n=0}^{\infty} \mathcal{G}_n/2^n z^n$ precisely captures the probabilistic behavior of $\mathcal{G}$. Notably, $a_{\mathcal{G}}(1)$ computes its probability of termination, and, as we'll see, the derivative $a'_{\mathcal{G}}(1)$ captures the *expected* number of reduction steps to termination.

A natural question is thus: is the complexity of the termination problem for a PHORS $\mathcal{G}$ somehow related with the algebraic or analytic properties of its generating function $a_{\mathcal{G}}(z)$?

Generating Functions via Linear Logic. Generating functions have been applied in the literature to the study of both formal grammars [37] and *first-order* programming languages (also in the probabilistic case, [34, 35]). However, extending this natural and powerful approach to functional programs is challenging, as it is *prima facie* not obvious how to associate *higher-order* functions with sequences of real numbers, as required to induce a generating function.

In this work we provide a solution to this problem, by presenting the first combinatorial study of the termination problem for higher-order recursive programs like PHORS. To do this, we exploit in an essential way two ideas coming from the toolbox of linear logic. The first fundamental ingredient is the *weighted relational model* [40], which associates proofs in linear logic, as well as PCF programs, with families of *formal power series* with coefficients in a continuous semiring. We show how this model can be used to associate each PHORS with a corresponding generating function $a_{\mathcal{G}}(z)$, capturing its probability of termination, in an elegant and compositional way.

As mentioned above, linearity has been recognized as a key factor to ensure the decidability of the AST problem for PHORS [42]. The viewpoint of generating functions provides a *novel* way to look at this result: using our interpretation via the weighted relational model, we will show that for all affine PHORS, while the corresponding language might *not* be context-free, the corresponding generating function is, in fact, always algebraic. Actually, the combinatorial viewpoint allows us to go beyond the linear/affine restriction and consider even *non-linear* PHORS typable via the discipline of *graded* or *bounded exponentials* [26, 39, 8, 24]: this discipline allows one to define types of the form $!_n A \multimap B$, intuitively expressing functions from A to B using their inputs *at most* n times.

Hence, on the one hand, in this work we capture a *larger* class of programs than [42], including PHORS admitting both bounded and unbounded forms of duplication, for which the AST and PAST problems remains decidable. On the other, our results suggest the *robustness* of the class of PHORS individuated by [42]: all PHORS typable in our discipline are shown equivalent (in terms of both probabilistic termination and the corresponding language) to some PAHORS, even though reconstructing the latter effectively may lead to an exponential blow up in size.

Contributions. The weighted relational semantics associates each PHORS with a generating function defined as (part of) the minimal solution of a system of countably many recursive equations between formal power series. While such systems cannot be solved explicitly in most cases, in this work we explore a class of PHORS whose generating functions can be handled effectively. More precisely, our contributions can be summarized as follows:

- First, we introduce a class of recursive systems, called *finitary*, which, although consisting in countably many fps equations, can be reduced to *finite polynomial systems*, and can thus be treated effectively. This is in Section 5.
- Then, we introduce the *finitely bounded PHORS*, PBHORS$^{<\infty}$, extending the PAHORS of [42] with finitely bounded non-linearity, and show that the corresponding generating functions are finitary, and, in particular, algebraic. Moreover, we use these facts to prove the decidability of AST and PAST for this class. This is in Section 6.
- Finally, we introduce an even larger class PBHORS$^{\infty}$, extending the PBHORS$^{<\infty}$ with a restricted use of *unbounded* non-linearity, still preserving algebraicity and decidability. Notably, the PBHORS$^{\infty}$ can be *composed*, thus enabling a modular study of their generating functions. This is in Section 7.

Complete proofs can be found in the extended version [15].

2 From PHORS to Generating Functions, via Linear Logic

In this section we provide an overview of our approach to probabilistic termination via algebraic generating functions.

The Generating Function of a PHORS. Consider a program $\mathcal{G}$ defined by the equations:

$$\begin{aligned} Fx &= x \oplus_{\frac{1}{2}} Fx \\ S &= F\ e \end{aligned} \tag{1}$$

where $F : o \to o$. This is a very simple example of a order-1 PHORS: the upper case letters F, S are called *non-terminal* symbols, the execution of the program starts from the *source* non-terminal $S : o$ by applying instances of the equations, read from left to right, as well as probabilistic choices, terminating when the unit constant $e : o$ is, eventually, produced. If we ignore probabilities, we can consider the *branch language* $\mathcal{L}(\mathcal{G})$ computed by the PHORS $\mathcal{G}$ as follows: replace each choice $t \oplus_{\frac{1}{2}} u$ with a non-deterministic operation $ct + cu$ marked by an order-1 function symbol $c : o \to o$; this leads to consider the following (non-deterministic) HORS $\mathcal{G}'$, where $F' : (o \to o) \to \tau$:

$$\begin{aligned} F'zx &= zx + F'zx \\ S &= F'\ c\ e \end{aligned} \tag{2}$$

The non-deterministic execution of $\mathcal{G}'$ produces an infinite tree whose finite branches form the language $\mathcal{L}(\mathcal{G}) = \{c^n e \mid n \in \mathbb{N}\}$.

Our goal is to extract a generating function from programs like (1), i.e. a formal power series $a_{\mathcal{G}}(z) = \sum_{i=0}^{\infty} a_i z^i$ capturing the probabilistic termination of $\mathcal{G}$. Recall that a power series induces an *analytic function* defined over some open subset $U \subset \mathbb{C}$ of the field of complex numbers, which (provided this set is not empty) can be studied with methods coming from both algebra and complex analysis. For instance, the formal power series $a(x) = \sum_{n=0}^{\infty} \frac{1}{2^n} x^n$ is equal to the analytic function $a(x) = \frac{1}{1-\frac{x}{2}}$ over the open set of complex numbers of absolute value strictly less than 2.

As anticipated, our fundamental ingredient to extract generating functions from PHORS is the weighted relational model of linear logic. This semantics is often styled *quantitative*, as it provides a *precise count* of the number of times that a program may use each of its inputs during any of its reductions. If we consider the HORS $\mathcal{G}'$, we see that, for any $i \in \mathbb{N}$, the term $F'zx$ has a unique way of terminating using z exactly i times. Formally, each such reduction yields, in the semantics, a monomial of the form $z^i x$. Notice that the parameter z precisely counts the number of times that a choice is made. By considering all possible reductions, the corresponding monomials yield then the power series below:

$$a_{F'}(z,x) = \sum_{i>0}^{\infty} z^i x$$

To get back to the original PHORS $\mathcal{G}$ we can simply add probabilities back, by interpreting the choice operator $x \oplus_{\frac{1}{2}} y$ as $z(\frac{1}{2}x + \frac{1}{2}y)$, yielding the rational power series

$$a_F(z,x) = \sum_{i>0}^{\infty} \frac{1}{2^i} z^i x = \frac{x}{1-\frac{z}{2}} - x$$

saying that the probability that Fx terminates using x once and making $i > 0$ choices (i.e. using z i times) is $\frac{1}{2^i}$. As e trivially terminates with probability 1, we can then associate the source symbol S with the generating function $a_S(z) := a_F(z,1) = \sum_{i>0}^{\infty} \frac{1}{2^i} z^i = \frac{1}{1-z/2} - 1$, which describes the reductions with i choices, and finally deduce that $\mathcal{G}$ terminates with probability $a_S(1) = 1$, i.e. almost surely.

Using the ideas just sketched, as we show in Section 4, any PHORS $\mathcal{G}$ can be associated with a generating function $a_{\mathcal{G}}(z) := a_S(z) = \sum_{n=0}^{\infty} a_n z^n$ such that a_n designates the probability that $\mathcal{G}$ terminates after *exactly* n reduction steps. Observe that the probability of termination of $\mathcal{G}$ is given by $a_{\mathcal{G}}(1)$. Moreover, $a'_{\mathcal{G}}(1)$, where $a'_{\mathcal{G}}(z) = \sum_{n=0}^{\infty} n a_n z^{n+1}$ indicates the *derivative* of $a_{\mathcal{G}}(z)$, precisely captures the *expected* number of reduction steps to termination.

It is worth observing at this point that, in our setting, the generating function $a_{\mathcal{G}}(z)$, while characterizing the termination problem for the PHORS $\mathcal{G}$, *does not* characterize the corresponding branch language: for instance, two PHORS respectively inducing the non context-free language $\mathcal{L}_1 = \{0^n 1^n 0^n \mid n \in \mathbb{N}\}$ and the context-free language $\mathcal{L}_2 = \{0^n 1^{2n} \mid n \in \mathbb{N}\}$ would *both* induce a rational generating function of the form $a(z) = \sum_{n=0}^{\infty} \frac{1}{2^{3n}} z^{3n} = \frac{1}{1-(z/2)^3}$, as the latter only counts the number of reduction steps to termination.

Algebraic Generating Functions via Bounded Exponentials. Consider now a slightly more complex PHORS $\mathcal{G}$, defined by:

$$\begin{aligned} Hfx &= (H(A \circ f)x \oplus_a H(B \circ f)x) \oplus_a f(fx) \\ Ax &= x \oplus_b \Omega \\ Bx &= x \oplus_c \Omega \\ S &= HIe, \end{aligned} \tag{3}$$

where $H : (o \to o) \to (o \to o)$, $A, B : (o \to o)$ and $S : o$, Ω stands for a diverging term, and a, b, c, d stand for rational biases for the probabilistic choice operators. Observe that H is not linear, as it may use the variable f twice. Using distinct constants $a, b, c : o \multimap o$ for each choice symbol we can see that the branch language of $\mathcal{G}$, namely

$$\mathcal{L}(\mathcal{G}) = \{a^{2|w|+2} ww \mid w \in \{b, c\}^*\},$$

is not context-free, as it contains an arbitrary word repeated twice.

In the weighted relational model, the non-terminal H yields, as before, a power series $a_H(z, f, x) = \sum_i^\infty H_i(z) f^i x$, where $H_i(z) = \sum_j H_{ij} z^j$ is the probability that Hfx terminates using f exactly i times, x once, but running through an arbitrary number of choices. By translating (3) into equations between the corresponding power series, we are led to:

$$H_2(z) = \alpha H_2(z) z^2 + \beta z, \qquad H_i(z) = 0 \quad (i \neq 2), \tag{4}$$

where $\alpha = (a^2 b^2 + a(1-a)c^2)$ and $\beta = 1 - a$. The power series $a_H(z, f, x) = H_2(z) f^2 x$ can then be computed by finding the minimal non-negative solution of the *polynomial equation*

$$p(z, H_2(z)) = 0,$$

where $p(z, w) = (\alpha z^2 - 1)w + \beta z$, yielding $a_H(z, f, x) = \frac{\beta z}{1-\alpha z^2} f^2 x$. This notably yields the termination probability $a_{\mathcal{G}}(1) = a_H(1, 1, 1) = \frac{\beta}{1-\alpha}$.

Unfortunately, termination probabilities cannot always be computed algebraically, as above. For instance, it is well-known that the generating functions of order-2 PHORS (which correspond to the so called *indexed grammars*) may fail to be algebraic [1]. In our semantics, the generating function $a_{\mathcal{G}}(z)$ of a PHORS $\mathcal{G}$ will be determined by a (countably) infinite family of equations, which we might not be able to solve algebraically.

The main theme of this work is then to find conditions under which the generating function $a_{\mathcal{G}}(z)$ may be found as the minimal solution of some *finite system of polynomial equations*. This will have two applications: in the first place, thanks to the characterization in terms of algebraicity *and* minimality, we can deduce the existence of effective methods to answer both the AST and PAST problem for the corresponding PHORS $\mathcal{G}$, by relying on the decidability of the *first order theory of the reals* **ETR** (see for example [43] for an account of this classic result by Tarski). On the other side, we can obtain for suitable classes of PHORS a result in the spirit of the celebrated Chomsky-Schützenberger enumeration theorem for context-free languages: the generating power series of such PHORS will be shown to be algebraic. This paves the way to studying the asymptotic rate of convergence of such recursion schemes through the techniques of analytic combinatorics, although we leave this interesting development as future work.

Algebraic PHORS via Bounded Exponentials. To obtain such conditions we rely, once more, on the toolbox of linear logic: as the relational semantics counts input uses, bounding the number of *unknowns* that we need to solve to compute the interpretation of the program amounts at bounding the number of uses that the program may do of each of its inputs. A first obvious idea would be thus to restrict ourselves to linear, or even affine, programs, as in [42]. Yet, the PHORS in (3) shows that one can well admit bounded forms of non-linearity. A standard and well-studied way to impose such bounds is via *bounded exponentials* [26, 39, 8, 24] $!_n A$, where a program $t : !_n A \multimap B$ may use its input *at most* n times. In our example, the non-terminal H could be given the bounded type $!_2(!_1 o \multimap o) \multimap (!_1 o \multimap o)$, as H uses its input f twice (indeed, all generating functions $H_i(z)$ are equal to zero, for $i \neq 2$).

But it is possible to go even beyond that, still preserving algebraicity and decidability. Similarly to how we handled the formal variable z counting probabilistic choices above, an algebraic PHORS may well use some inputs in an *unbounded* way, provided these are handled as *formal parameters*. This idea leverages the intrinsically parametric nature of the weighted relational semantics, which allows one to pass smoothly from power series $s(x, w)$ in x *and* w, with values over some semiring $\mathcal{R}$, to "parametric" power series $s(w)(x)$ in x *only* with coefficients in the semiring $\mathcal{R}\{\!\{w\}\!\}$ of power series in w. This allows us to admit also infinitely graded types $!_\infty \sigma \multimap \tau$, although used in a restricted way.

3 PHORS

In this section we recall the Higher Order Recursion Schemes (HORS), their probabilistic counterpart, the Probabilistic Higher Order Recursion Schemes (PHORS), as well as their correspondence with the (probabilistic) λY-calculus.

3.1 HORS, a.k.a. the λY-calculus

HORS, widely used in higher order model checking, can be seen as grammars that generate infinite ranked trees (or equivalently, infinite typed lambda terms). In the theory of HORS, the simple types are generated by the grammar $T = o \mid T \to T$. We define the order of a type as $ord(o) = 0$ and $ord(T \to S) = \max(ord(T) + 1, ord(S))$. A (deterministic) HORS $\mathcal{G}$ is a 4-uple $(\mathcal{N}, \mathcal{T}, \mathcal{D}, S)$ where $\mathcal{N}$ is a set of typed *non-terminals*, $\mathcal{T}$ is a set of typed *terminals*, S is a distinguished non-terminal called the *starting symbol* of $\mathcal{G}$ and $\mathcal{D}$ is a function associating each non-terminal L with a rewriting rule of the form: $Lx_1 \dots x_n = t$, where $FV(t) \subseteq \{x_1, \dots x_n\} \cup \mathcal{T}$ and $t : o$. These rules are to be understood as a (mutually) recursive definitions of the non-terminals that can be progressively unfolded: to $\mathcal{G}$ we will associate the rewriting system induced by the rules $Lt_1 \dots t_n \to t[x_1/t_1, \dots x_n/t_n]$. The rewriting, starting from S, will generate at each step finite (simply typed) terms containing the terminals $\mathcal{T}$; the "limit" of this set of terms is the *value tree* defined by $\mathcal{G}$ (for formal definitions, see [45], [9]). The nodes of this tree are labelled by terminals and its branching factor at each node is equal to the arity of the non terminal labelling that node.

The *branch language* $\mathcal{L}_{\mathcal{G}} \subseteq \mathcal{T}^*$ of an HORS $\mathcal{G}$ is the branch language of its value tree: its words are the sequences of terminals encountered in some finite branch of the tree. It is well known that the branch languages of order-1 HORS coincide with the context-free languages, while the branch languages of order-2 HORS coincide with the *indexed languages* of [2].

▶ **Example 1.** Consider the order-2 HORS $\mathcal{G} := (\mathcal{N}, \mathcal{T}, \mathcal{D}, S)$ defined as follows: its non terminals are $S : o$, $L : (o \to o) \to o \to o$, its terminals are $e : o$ of arity 0, $r : o \to o$ of arity 1 and $h : o \to o \to o$ of arity 2. Its rewriting rules are:

$$S = Lre$$
$$Lx_1x_2 = hx_2(Lx_1(Lx_1x_2))$$

The first reduction steps are $S \to Lre \to he(Lr(Lre))$. Now we can for example unfold the leftmost L and obtain $he(h(Lre)(Lr(Lr(Lre))))$.

The language of HORS is equivalent to the λY-calculus, that is, the simply typed λ-calculus enriched with cartesian products and a fixpoint $Y : (T \to T) \to T$: in $\mathcal{G} = (\mathcal{N}, \mathcal{T}, \mathcal{D}, S)$ the function $\mathcal{D}$ can be equivalently defined as a function associating each non-terminal $L \in \mathcal{N}$ with a simply typed term $\mathcal{N} \vdash \lambda x_1 \dots . x_n . t_L : \mathcal{N}(L)$, such that $\mathcal{N}, x_1, \dots, x_n \vdash t_L : o$. To $\mathcal{G}$ can we then associate the λY-term $t_{\mathcal{G}} : \mathcal{N}$, where $t_{\mathcal{G}} := \lambda\langle L_1, \dots, L_n\rangle.\langle t_{L_1} \dots t_{L_n}\rangle : \mathcal{N} \to \mathcal{N}$, where we identify the context $\mathcal{N}$ with the product type $\mathcal{N}(L_1) \times \dots \times \mathcal{N}(L_n)$.

3.2 PHORS, a.k.a. the Probabilistic λY-calculus

PHORS are a probabilistic extension of HORS: they can be obtained by adding to the syntax of HORS a set of new constant symbols $\oplus_p, p \in \mathbb{Q} \cap [0, 1]$, of type $o \to o \to o$, $t_1 \oplus_p t_2$ representing a probabilistic choice that chooses with probability p the first argument and

with probability $1-p$ the others; moreover, as in [36], we focus on PHORS with a unique terminal symbol $e : o$. A PHORS can thus be defined simply as a triple $\mathcal{G} = (\mathcal{N}, \mathcal{D}, S)$. Each non-terminal L will be given a recursive definitions of the form

$$Lx_1 \dots x_n = t_L \oplus_p t_R,$$

with the two (one step) probabilistic rewriting rules

$$Lt_1 \dots t_n \xrightarrow{\mathsf{l},p} t_L[x_1/t_1, \dots x_n/t_n] \qquad Lt_1 \dots t_n \xrightarrow{\mathsf{r},1-p} t_R[x_1/t_1, \dots x_n/t_n].$$

Here, we annotate the rewriting to keep track of the probability and the direction of this choice. For a reduction $R : s \xrightarrow{\sigma_1,p_1} s_1 \dots \xrightarrow{\sigma_k,p_k} s_k$, we also write $R : s \xrightarrow{\sigma,p} s_k$, where $\sigma = \sigma_1 \dots \sigma_n$ and $p = p_1 \dots p_n$.Given a PHORS $\mathcal{G} = (\mathcal{N}, \mathcal{D}, S)$, we can define its probability of termination and expected number of steps to termination as

$$\mathbb{P}(\mathcal{G}\downarrow) := \sum_{R:S \xrightarrow{\sigma,p} e} p \qquad\qquad \mathbb{E}(\mathcal{G}\downarrow) := \sum_{R:S \xrightarrow{\sigma,p} e} |\sigma| p$$

We can now define the two central problems of this paper:

▶ **Definition 2.** *Given a PHORS $\mathcal{G}$, the AST problem asks to decide whether $\mathbb{P}(\mathcal{G}\downarrow) = 1$, and the PAST problem asks to decide whether $\mathbb{E}(\mathcal{G}\downarrow) < +\infty$*

▶ **Example 3.** Consider the order-1 PHORS defined by

$$Fx = F(Fx) \oplus_{\frac{1}{2}} x,$$
$$S = Fe.$$

As any choice either adds one F or deletes one, and termination comes when e is reached, it simulates a simple random walk starting from $+1$, adding ± 1 at each step and terminating at 0. We will see in the next sections that the aforementioned PHORS is AST but not PAST.

It is known that PAST implies AST. On the other hand, in [36], it is proved that AST is undecidable for PHORS of order at least 2, while a decision procedure exists for PHORS of order 1. In [42], it is shown that, if we restrict to terms typable via affine types, AST and PAST are decidable. The proof builds upon the correspondence, devised in [13], between affine PHORS and restricted tree stack automata. As a consequence of the theorems of Section 6, we will see that this can be proved without resorting to automata, but rather relying on the relational model of linear logic.

▶ Remark 4. To each PHORS we can associate an HORS whose value tree represents the probabilistic choices made during reduction: to this aim, it is enough to treat every $\oplus_p$ as a terminal symbol of arity 2. This way, we can see that the branching structure of the order-2 PHORS (3) from Section 2, yielding a non-context free branch language, cannot be simulated by any order-1 PHORS.

The language of PHORS is equivalent to the $P\lambda Y$-calculus, that is, to the extension of λY with the constants $e : o$ and $\oplus_p : o \to o \to o$, together with reduction rules $M \oplus_p N \xrightarrow{\mathsf{l},p} M$, $M \oplus_p N \xrightarrow{\mathsf{r},1-p} N$. The *call-by-name reduction* $\to$ of the $P\lambda Y$-calculus is the closure of the reductions above as well as β-reduction $(\lambda x.M)N \xrightarrow{\epsilon,1} M[N/x]$ and Y-reduction $YM \xrightarrow{\epsilon,1} M(YM)$ under the congruence $M \xrightarrow{\sigma,p} N \;\Rightarrow\; MP \xrightarrow{\sigma,p} NP$.

The translation of a PHORS $\mathcal{G}$ into a term $t_\mathcal{G}$ of the $P\lambda Y$-calculus works as in the case of HORS. In particular, reductions sequences $R : S \xrightarrow{\sigma,p} e$ to normal form are in one-to-one correspondence with probabilistic call-by-name reductions $R : M_S \xrightarrow{\sigma,p} e$ to normal form in $P\lambda Y$. For example, the PHORS from Example 3 is encoded in $P\lambda Y$ by $Yt_\mathcal{G} : (o \to o) \times o$, where $t_\mathcal{G} = \lambda\langle F, S\rangle.\langle \lambda x.F(Fx) \oplus_{\frac{1}{2}} x, Fe\rangle : ((o \to o) \times o) \to ((o \to o) \times o)$.

4 Interpreting PHORS by Formal Power Series

In this section we recall the relational semantics of linear logic with weights taken over a continuous semiring $\mathcal{R}$ [40], whose morphisms can be presented as families of formal power series with coefficients in $\mathcal{R}$. We then illustrate how this semantics associates each PHORS with a generating function capturing its probability of termination.

4.1 Formal Power Series over a Semiring

Let us first introduce formal power series (fps in short); for more details about them and their applications to combinatorics we refer to [22], [33].

In the following we let $\mathcal{R}, \mathcal{Q}$ range over semirings (which we suppose to be commutative and having units 0 and 1). A semiring $\mathcal{R}$ is *continuous* if it is ordered, has 0 as its bottom element and all directed joins admit supremum, such supremum commuting with multiplication (see e.g. [16]). In particular, continuous semiring admits infinite sums. Our fundamental example of continuous semiring is given by the non-negative possibly infinite reals $\mathbb{R}^{+\infty}_{\geq 0}$, with standard addition and multiplication.

Given a set Σ, let $!\Sigma$ be the set of finite multisets over it, i.e functions $\mu : \Sigma \to \mathbb{N}$ with finite support. Given a semiring $\mathcal{R}$, the set of *formal power series (with commuting variables) over* $\mathcal{R}$, denoted $\mathcal{R}\{\!\{\Sigma\}\!\}$, is the set of all functions $!\Sigma \to \mathcal{R}$. More concretely, if we introduce for each $s \in \Sigma$ a variable x_s (we will denote by x_Σ the set of all these variables), then a finite multiset $\mu \in !\Sigma$ can be seen as the monomial $x^\mu_\Sigma := \Pi_{s\in\Sigma} x_s^{\mu(s)}$; then any formal power series $s \in \mathcal{R}\{\!\{\Sigma\}\!\}$ can then be expressed as a formal sum:

$$s = \sum_{\mu\in!\Sigma} s_\mu x^\mu_\Sigma.$$

We will sometimes write $s(x_\Sigma)$ to highlight which variables appear in s. The *support* $\mathsf{supp}\,(s) \subseteq !\Sigma$ is the set of multisets μ such that $s_\mu \neq 0$. We let $\mathcal{R}\{\Sigma\} \subseteq \mathcal{R}\{\!\{\Sigma\}\!\}$ be the set of *polynomials*, i.e. of fps with finite support, which can written, as usual, as finite sums

▶ **Example 5.** The power series $s(x) = \sum_n (1/2)^n x^n$ belongs to $\mathbb{Q}\{\!\{x\}\!\}$. The power series $s(x) = \sum_{n=0}^\infty \sum_{i+j=n} (1/3)^n x_0^i x_1^j$, i.e.$\sum_{\mu\in!\{0,1\}} (1/3)^{\mu(0)+\mu(1)} x^\mu$, belongs to $\mathbb{Q}\{\!\{x_0, x_1\}\!\}$.

In $\mathcal{R}\{\!\{\Sigma\}\!\}$ we can define two operations: sum, performed componentwise: $\sum_{\mu\in!\Sigma} r_\mu x^\mu_\Sigma + \sum_{\mu\in!\Sigma} s_\mu x^\mu_\Sigma := \sum_{\mu\in!\Sigma}(r_\mu + s_\mu)x^\mu_\Sigma$ and the Cauchy product: $\sum_{\mu\in!\Sigma} r_\mu x^\mu_\Sigma \cdot \sum_{\mu\in!\Sigma} s_\mu x^\mu_\Sigma := \sum_{\kappa\in!\Sigma} \left(\sum_{\mu+\nu=\kappa} r_\mu s_\nu\right) x^\kappa_\Sigma$. With these operations, $\mathcal{R}\{\!\{\Sigma\}\!\}$ is a semiring and, if $\mathcal{R}$ is continuous and we take on $\mathcal{R}\{\!\{\Sigma\}\!\}$ the pointwise partial order, it is also continuous. Continuity is essential to define the *composition* of formal power series: let $r \in \mathcal{R}\{\!\{\Sigma\}\!\}$ and let $s_\Sigma \in \mathcal{R}\{\!\{\Sigma'\}\!\}^\Sigma$ be a Σ-indexed family of power series over the set Σ', $s_\sigma = \sum_{\nu\in!\Sigma'} s_{\sigma,\nu} y^\nu$. Then, we can define the power series $r(s_\Sigma) \in \mathcal{R}\{\!\{\Sigma'\}\!\}$ by the formula:

$$r(s_\Sigma) = \sum_{\kappa\in!\Sigma'} \left(\sum_{\mu=[\sigma_1,\ldots,\sigma_k]\in!\Sigma} \sum_{\substack{(\nu_1,\ldots,\nu_k)\in(!\Sigma')^k \\ \nu_1+\cdots+\nu_k=\kappa}} r_\mu \cdot \prod_{i=1}^k s_{\sigma_i,\nu_i} \right) y^\kappa_\Sigma.$$

Observe that if there is at least an s_σ such that $s_{\sigma,[]} \neq 0$, the sum over $!\Sigma$ will be infinite; still, by continuity, we can define its value to be the sup of the partial sums (yet, without continuity, this composition need not be well-defined). If the s_Σ are all constants (i.e. $s_{\sigma,\mu} = 0\,\forall \mu \neq []$), we will say that $r(s_\Sigma) \in \mathcal{R}\{\!\{\emptyset\}\!\} = \mathcal{R}$ is the *value of* r *at the point* $(s_{\sigma,[]})_{\sigma\in\Sigma}$. With respect to these operations, the polynomials $\mathcal{R}\{\Sigma\}$ form a (non-continuous) sub-semiring of $\mathcal{R}\{\!\{\Sigma\}\!\}$.

Since $\mathcal{Q} := \mathcal{R}\{\!\{\Sigma\}\!\}$ is a continuous semiring, for any set Σ' we can consider the continuous semiring $\mathcal{Q}\{\!\{\Sigma'\}\!\}$ of formal power series whose coefficients are *themselves* power series (on $\mathcal{R}$), and we have the isomorphism $\mathcal{Q}\{\!\{\Sigma'\}\!\} = (\mathcal{R}\{\!\{\Sigma\}\!\})\{\!\{\Sigma'\}\!\} \equiv \mathcal{R}\{\!\{\Sigma + \Sigma'\}\!\}$: this generalizes the remark that a fps in *two* variables $s(x,y) = \sum_{\mu,\nu} s_{\mu,\nu} x^\mu y^\nu$ can always be written as a fps $s_y(x) = \sum_\mu (s_y)_\mu x^\mu$ in x *only*, whose coefficients are fps $(s_y)_\mu = \sum_\nu s_{\mu,\nu} y^\nu$ in y.

If we do not start, as we do here, with a continuous semiring $\mathcal{R}$, but rather with a ring R, almost all constructions still work: we can similarly define the set $R\{\!\{\Sigma\}\!\}$ of formal power series over R, which is a ring, as well as its subring $R\{\Sigma\}$ of polynomials over R. Still, the composition of formal power series will not be defined in general, as it involves an infinite sum, while the composition of polynomials remains well-defined.

4.2 The Weighted Relational Semantics

Now, we recall the link between formal power series and the weighted relational model of linear logic [40]. Given a continous semiring $\mathcal{R}$, the category $\mathcal{R}\mathbf{Rel}_!$ has sets as objects, and morphisms $\mathcal{R}\mathbf{Rel}_!(X,Y)$ are $\mathcal{R}$-valued matrices indexed by $!X \times Y$. $\mathcal{R}\mathbf{Rel}_!$ is the coKleisli category, with respect to the ! comonad, of the more familiar category $\mathcal{R}\mathbf{Rel}$ of sets and $\mathcal{R}$-valued matrices.

At the same time, $\mathcal{R}\mathbf{Rel}_!$ can be seen as a category of formal power series: as a matrix $(t_{\mu,y})_{\mu \in !X, y \in Y}$ can be identified with the Y-indexed family of fps $(\sum t_{\mu,y} x_X^\mu)_{y \in Y}$, the morphisms of $\mathcal{R}\mathbf{Rel}_!$ can be identified with (families of) formal power series, i.e. $\mathcal{R}\mathbf{Rel}_!(X,Y) \equiv \mathcal{R}\{\!\{X\}\!\}^Y$. Indeed, composition in $\mathcal{R}\mathbf{Rel}_!$ is given by (pointwise) composition of the underlying power series, with identities $\mathrm{id}_X \in \mathcal{R}\{\!\{X\}\!\}^X$ given by $\mathrm{id}_i(x_X) = x_i$.

$\mathcal{R}\mathbf{Rel}_!$ is cartesian closed, with cartesian products given by $X + Y$ (with neutral $0 := \emptyset$) and exponentials given by $!X \times Y$. This allows us to interpret simple types as $[\![o]\!] = 1 := \{\star\}$, $[\![T \times U]\!] = [\![T]\!] + [\![U]\!]$ and $[\![T \to U]\!] = ![\![T]\!] \times [\![U]\!]$. Intuitively, a type T translates into a set of variables, and a term $\Gamma \vdash t : T$ into a $[\![T]\!]$-indexed family of fps $[\![t]\!]_i^{\mathcal{R}}(x_\Gamma)$ in *as many variables as* $[\![\Gamma]\!]$. Here we can observe the main challenge appearing with higher-order types, as these are interpreted by *infinitely many* variables: for instance, $[\![o \to o]\!] = ![\![o]\!] \times [\![o]\!] \equiv \mathbb{N} \times 1 \equiv \mathbb{N}$ translates into a countable sequence of variables, and $[\![(o \to o) \to (o \to o)]\!] \equiv !\mathbb{N} \times \mathbb{N}$ has one distinct variable $x_{\mu,n}$ for each $\mu \in !\mathbb{N}$ and $n \in \mathbb{N}$. Higher-order terms correspond thus to families of fps in *countably many* variables.

▶ **Example 6.** Consider the program $t = \lambda x.y(yx)$, where $y : (o \to o) \vdash t : o \to o$: recalling $[\![o \to o]\!] = !o \times o \equiv \mathbb{N}$, we have that t is interpreted by a $\mathbb{N}$-indexed family of power series $(b_i(y_{\mathbb{N}}))_{i \in \mathbb{N}} \in \mathcal{R}\mathbf{Rel}_!(\mathbb{N},\mathbb{N}) = \mathcal{R}\{\!\{\mathbb{N}\}\!\}^{\mathbb{N}}$. Now, think of the variables $y_{\mathbb{N}}$, which interpret the function $y : o \to o$, as encoding the fps $a(x) = \sum_n y_n x^n \in \mathcal{R}\{\!\{\mathbb{N}\}\!\}\mathbf{Rel}_!(1,1) \equiv (\mathcal{R}\{\!\{\mathbb{N}\}\!\})\{\!\{1\}\!\}$; the term t should translate then into the composition

$$a(a(x)) = \sum_n y_n \left(\sum_m y_m x^m \right)^n = \sum_{i=0}^{\infty} \left(\sum_{\substack{(n, m_1, \ldots, m_n), \\ m_1 + \cdots + m_n = i}} y_n y_{m_1} \cdots y_{m_n} \right) x^i,$$

from which we can deduce $b_i(y_{\mathbb{N}}) = \sum_{\substack{(n, m_1, \ldots, m_n), \\ m_1 + \cdots + m_n = i}} y_n y_{m_1} \cdots y_{m_n}$. Notice that $b_1(y_{\mathbb{N}}) = y_1$. Moreover, the evaluation tx of t over some variable $x : o$ is then precisely interpreted by $a(a(x)) \in \mathcal{R}\mathbf{Rel}_!(\mathbb{N}+1,1) = \mathcal{R}\{\!\{\mathbb{N}+1\}\!\} \equiv (\mathcal{R}\{\!\{\mathbb{N}\}\!\})\{\!\{1\}\!\}$.

Beyond exponentials, $\mathcal{R}\mathbf{Rel}_!$ is endowed with Conway fixpoints [29], [27]: given a morphism $f \in \mathcal{R}\mathbf{Rel}_!(X+Y,X)$, we define $\mathsf{fix}\, f \in \mathcal{R}\mathbf{Rel}_!(Y,X)$ as follows: take the sequence $f^0 := 0 \times id_Y \in \mathcal{R}\mathbf{Rel}_!(1 \times Y, X \times Y)$, $f^{n+1} := f \circ (f^n \times id_Y) \circ \langle id_o, \Delta_Y \rangle \in \mathcal{R}\mathbf{Rel}_!(1 \times Y, X)$

and finally let $\mathsf{fix}_{Y,X} f := \sup_n f^n \in \mathcal{R}\mathbf{Rel}_!(Y, X)$. It is worth to restate this construction in terms of power series: f will be represented by an X-indexed family $(s_x(x_X, x_Y))_{x\in X}$. Then we can define its iterates and the fixpoint as follows, for $x \in X$:

$$\begin{aligned} r_x^{(0)}(x_Y) &= 0 \in \mathcal{R}\{\!\{Y\}\!\}, \\ r_x^{(n+1)}(x_Y) &= s_x(r_X(x_Y), x_Y), \\ (\mathsf{fix}\, s_X)_x(x_Y) &= \sup_n r_x^{(n)}(x_Y). \end{aligned} \tag{5}$$

From this, it is clear that the power series $r_X = \mathsf{fix}\, f$ are the minimal solution of the infinite family of equations: $(r_x = s_x(r_X, x_Y))_{x\in X}$.

Using the cartesian closed structure and the fixpoints, any term $\Gamma \vdash t : T$ of λY yields in $\mathcal{R}\mathbf{Rel}_!$ a family of fps $[\![t]\!]^{\mathcal{R}} \in \mathcal{R}\{\!\{[\![\Gamma]\!]\}\!\}^{[\![T]\!]}$. To interpret $P\lambda Y$, we restrict our attention to semirings of the form $\mathcal{R} = \mathbb{R}_{\geq 0}^{+\infty}\{\!\{\{z\} + \Sigma\}\!\}$, where z denotes a distinguished variable. This allows us to interpret probabilistic choice (with bias p) as

$$[\![M \oplus_p N]\!]^{\mathcal{R}} = pz \cdot [\![M]\!]^{\mathcal{R}} + (1-p)z \cdot [\![N]\!]^{\mathcal{R}}$$

. As shown by Proposition 9 below, the variable z plays the role of a *counter* for each probabilistic choice: each reduction $t \xrightarrow{\sigma,p} e$ will produce a monomial $pz^{|\sigma|}$ in the semantics.

▶ Remark 7. [7] considers the interpretation of *parametric* choices $M \oplus_x N$, i.e. choices according to some unknown bias x, by taking the semiring $\mathcal{Q} = \mathcal{R}\{\!\{x, \overline{x}\}\!\}$ and letting $[\![M \oplus_p N]\!]^{\mathcal{Q}} = x \cdot M + \overline{x} \cdot N$.

We now see how any PHORS $\mathcal{G} = (\mathcal{N}, \mathcal{D}, S)$ produces a morphism $[\![\mathcal{G}]\!]^{\mathbb{R}_{\geq 0}^{+\infty}\{\!\{z\}\!\}}$ (i.e. $[\![t_{\mathcal{G}}]\!]^{\mathbb{R}_{\geq 0}^{+\infty}\{\!\{z\}\!\}}$) in $\mathbb{R}_{\geq 0}^{+\infty}\{\!\{z\}\!\}\mathbf{Rel}_!([\![\mathcal{N}]\!], [\![\mathcal{N}]\!]) \equiv (\mathbb{R}_{\geq 0}^{+\infty}\{\!\{z\}\!\})\{\!\{[\![\mathcal{N}]\!]\}\!\}^{[\![\mathcal{N}]\!]}$, i.e. a $[\![\mathcal{N}]\!]$-indexed family of power series in the variables $x_{[\![\mathcal{N}]\!]}$. All this leads to the following definitions:

▶ **Definition 8** (PGF of a PHORS). *For every PHORS $\mathcal{G} = (\mathcal{N}, \mathcal{R}, S)$ and non-terminal $L_i : T_1 \to \cdots \to T_n \to o \in \mathcal{N}$, letting $\Sigma = [\![T_1]\!] + \cdots + [\![T_n]\!]$, we define:*

$$a_{L_i}(z)(x_\Sigma) := \pi_i \left(\mathsf{fix}_{\mathcal{N}} [\![\mathcal{G}]\!]^{\mathbb{R}_{\geq 0}^{+\infty}\{\!\{z\}\!\}} \right) \in (\mathbb{R}_{\geq 0}^{+\infty}\{\!\{z\}\!\})\{\!\{\Sigma\}\!\}.$$

The fps $a_{\mathcal{G}}(z) := a_S(z) \in \mathbb{R}_{\geq 0}^{+\infty}\{\!\{z\}\!\}$ is called the probabilistic generating function of $\mathcal{G}$.

Observe that the fps $a_{L_i}(z)(x_\Sigma)$ are the minimal solutions of the equational system $(r_x = ([\![\mathcal{G}]\!]^{\mathcal{R}})_x(r_x, z))_{x\in[\![\mathcal{N}]\!]}$ induced by $\mathcal{G}$.

The generating function $a_{\mathcal{G}}(z)$ precisely captures the call-by-name probabilistic execution of closed terms, as stated below:

▶ **Proposition 9.** *For any PHORS $\mathcal{G}$, $a_{\mathcal{G}}(z) = \sum_{i=0}^{\infty} \mathbb{P}(\mathcal{G} \downarrow_i) z^i$ where $\mathbb{P}(\mathcal{G} \downarrow_i) = \sum_{R:t\to^i e} w(R)$ is the probability that S terminates after i probabilistic steps. In particular, $a_{\mathcal{G}}(1) = \mathbb{P}(\mathcal{G} \downarrow)$.*

Proof. From [40, 18, 7] we know that each reduction $S \xrightarrow{\sigma,p} e$ precisely adds up one monomial $pz^{|\sigma|}$, hence $a_{|\sigma|}z^{|\sigma|}$ in $a_{\mathcal{G}}(z)$ is the sum of the probabilities of all reductions of length $|\sigma|$. ◀

From Proposition 9 we also deduce that the *derivative* of $a_{\mathcal{G}}(z)$ captures the expected number of steps to termination:

▶ **Corollary 10** ([17]). $a'_{\mathcal{G}}(1) = \sum_{i=1}^{\infty} i \cdot \mathbb{P}(\mathcal{G} \downarrow_i) = \mathbb{E}(\mathcal{G} \downarrow)$.

▶ **Example 11.** Consider the PHORS from Example 3, with $t_F = F(Fx) \oplus_{\frac{1}{2}} x$ and $t_S = Fe$. The interpretation in $\mathbb{R}^{+\infty}_{\geq 0}\{\!\{z\}\!\}\mathbf{Rel}_!$ of $t_{\mathcal{G}} = \lambda\langle F, S\rangle.\langle t_F, t_S\rangle : ((o \to o) \times o) \to (o \to o) \times o$ is a pair $\langle s^F, s^S\rangle$ formed by a $\mathbb{N}$-indexed family of fps $s^F_i \in (\mathbb{R}^{+\infty}_{\geq 0}\{\!\{z\}\!\})\{\!\{\mathbb{N}+1\}\!\}$ and a single fps $s^S \in (\mathbb{R}^{+\infty}_{\geq 0}\{\!\{z\}\!\})\{\!\{\mathbb{N}+1\}\!\}$ given, for $i \in \mathbb{N}$, and recalling $b_i(y_{\mathbb{N}})$ from Example 6, by

$$s^F_i(y_{\mathbb{N}+1}, z) = \begin{cases} \frac{1}{2}zb_1(y_{\mathbb{N}}) + \frac{1}{2}z = \frac{1}{2}(zy_1^2 + z) & \text{if } i = 1, \\ \frac{1}{2}zb_i(y_{\mathbb{N}}) & \text{if } i \neq 1 \end{cases} \qquad s^S(y_{\mathbb{N}+1}, z) = \sum_{i\in\mathbb{N}} y_i.$$

Computing $\mathsf{fix}\ \langle s^F, s^S\rangle$ means finding a minimal solution $(a_i(z))_{i\in\mathbb{N}+1} \in (\mathbb{R}^{+\infty}_{\geq 0}\{\!\{z\}\!\})^{\mathbb{N}+1}$ of the fixpoint equations $a_i(z) = s^F_i(a_{\mathbb{N}+1}(z), z)$ $(i \in \mathbb{N})$ and $a_{\mathcal{G}}(z) = s^S(a_{\mathbb{N}+1}(z), z)$. One can easily see that this yields $a_i(z) = 0$, for $i \neq 1, \star$, while $a_{\mathcal{G}}(z) := a_\star(z) = a_1(z) \in \mathbb{R}^{+\infty}_{\geq 0}$ can be found as minimal solution of the polynomial equation

$$a_{\mathcal{G}}(z) = \frac{1}{2}\left(za^2_{\mathcal{G}}(z) + z\right) \tag{6}$$

We will see in the next section that this solution is given by the series $a_{\mathcal{G}}(z) = \sum_{i=0}^{\infty} \frac{C_i}{2^{2i+1}} z^{2i+1}$, where C_i is the i-th Catalan number.

For order-1 PHORS, we have the following useful lemma:

▶ **Lemma 12.** *For all order-1 PHORS $\mathcal{G} = (\mathcal{N}, \mathcal{R}, S)$ and non-terminal symbol $L \in \mathcal{N}$, the fps $a_L \in (\mathcal{R}\{\!\{z\}\!\})\{\!\{x_1, \ldots, x_n\}\!\}$ is* affine*: there exist $w_0, w_1, \ldots, w_n \in \mathbb{R}^{+\infty}_{\geq 0}\{\!\{z\}\!\}$ such that*

$$a_L(z)(x_1, \ldots, x_n) = w_0(z) + w_1(z)x_1 + \cdots + w_n(z)x_n,$$

where $w_0(1) = \mathbb{P}[L\vec{x} \to^* e]$ and $w_{i+1}(1) = \mathbb{P}[L\vec{x} \to^* x_{i+1}]$.

This result translates the fact that in a reduction of $Lx_1 \ldots x_n$ to normal form, a ground variable $x_i : o$ can occur in head position *at most once*: as soon as it does, we must have $Lx_1 \ldots x_n \to^* x$, that is, the reduction has terminated.

5 Algebraic Power Series

As we saw, power series are infinitary objects and their manipulation involves the concept of limit, making them generally uncomputable. In this section, we explore a way to provide *finitary* specifications for power series through the concept of algebraicity. Algebraic power series are widely used in algebra and combinatorics, in the context of rings or fields. Their treatment in semirings is less standard, but extensively studied in the context of formal language theory. We will show that, through the weighted relational semantics, such ideas apply naturally to probabilistic λ-calculi.

5.1 Fixpoint Algebraic Systems

In the previous section we have seen that PHORS can be associated with formal power series defined as the minimal solution of a (generally infinite) equational system. In practice, computing such power series directly can be very hard; recall, by the way, that such fps capture properties like AST or PAST, which are undecidable in general. However, when the equational system defining a family of power series can be expressed by *finitely many polynomial equations*, it becomes possible to extract information about such power series as well as the sequences of their coefficients.

As a famous example, consider again the fps $c(x) = \sum_{n\geq 0} C_n x^n \in \mathbb{R}\{\!\{x\}\!\}$. $c(x)$ is the Taylor expansion around 0 of the (analytic) function $\frac{1-\sqrt{1-4x}}{2x}$. From this, it is easy to see, taking $p(w,x) = xw^2 - w + 1$, that $p(c(x),x) = 0$. Observe that, while computing e.g. $c(1/4) = \sum_{n\geq 0} \frac{1}{4^n(n+1)}\binom{2n}{n}$ as a limit is hard, we can easily deduce that $c(1/4) = 1$ is the (only) positive root of $p(w, 1/4)$.

▶ **Example 13.** Equation (6) from Example 11 can be rewritten as $p(a_{\mathcal{G}}(z), z) = 0$, where $p(w,z) = \frac{1}{2}zw^2 - w + \frac{1}{2}z$, which gives the solution $a_{\mathcal{G}}(z) = \frac{1-\sqrt{1-z^2}}{z}$. Using the fps $c(x)$ from above, we deduce, with $x = \left(\frac{z}{2}\right)^2$, $a_{\mathcal{G}}(z) = \frac{1-\sqrt{1-4z}}{2z}z = c(z)\frac{z}{2} = \sum_{i=0}^{\infty}\frac{C_i}{2^{2i+1}}z^{2i+1}$. The interpretation of this is that, for any $i \in \mathbb{N}$, there are C_i terminating reductions that make $2i+1$ probabilistic choices. On the other hand, there is no terminating path of even length, as there is no random walk going from 1 to 0 in an even number of steps. Now, as 1 is the smallest root of $p(w,1)$, we easily obtain $\mathbb{P}[\mathcal{G}\downarrow] = a_{\mathcal{G}}(1) = 1$, that is, that $\mathcal{G}$ is AST; moreover, $\mathbb{E}[S\downarrow]$ is given by the diverging series $a'_{\mathcal{G}}(1) = \sum_{i=0}^{\infty}\frac{2i+1}{2^{2i+1}}C_i = \infty$, so PAST fails.

Before considering algebraic power series on a (continuous) semiring, let us recall the case of rings, indeed the one most customary in combinatorics. We assume our rings to be integral domains.

▶ **Definition 14.** *Let R be a ring and let $R' \leq R$ a subring. A tuple $(s_1, \dots s_n)$ of formal power series $s_i \in R\{\!\{\Sigma\}\!\}$ is an* R'-algebraic family *if there exist polynomials $p_i \in R'\{w_1, \dots w_n, \Sigma\}$ such that:*

$$\begin{cases} p_1(s_1, \dots s_n, x_\Sigma) = 0 \\ \dots \\ p_n(s_1, \dots s_n, x_\Sigma) = 0 \end{cases}$$

When $n = 1$, an R'-algebraic family (s) is simply called a R'-algebraic fps *and the polynomial p_1 is called an* annihilating polynomial *of s.*

▶ **Example 15.** Let $s \in \mathbb{R}\{\!\{x\}\!\}$ be $\sum_{n\geq 0} x^n$. Taking the polynomial $p(w,x) = w(1-x) - 1$, we get $p(s(x), x) = 0$. Hence, s is algebraic over $\mathbb{Q}$. Indeed, s is the multiplicative inverse $(1-x)^{-1}$ of $(1-x)$ in $\mathbb{R}\{\!\{x\}\!\}$. In general, every $\mathbb{Q}$-rational function (i.e every power series of the form $r_1(x_\Sigma)r_2(x_\Sigma)^{-1}$ for $r_1, r_2 \in \mathbb{Q}\{\Sigma\}$) is algebraic over $\mathbb{Q}$.

▶ Remark 16. The non-zero coefficients of an algebraic power series cannot be "too far": if $a(x) = \sum_n a_n x^{p_n}$ is algebraic, then $\lim_n p_n/n < \infty$: this is a consequence of *Fabry's gap theorem*, which asserts that, when $\lim_n p_n/n$ diverges, a cannot be analytically continued on any point of its circle of convergence. For instance, no series of the form $\sum_n a_n x^{n^2}, \sum_n a_n x^{n^3}, \sum_n a_n x^{2^n}$ is algebraic.

▶ Remark 17. If $s \in \mathbb{R}\{\!\{x\}\!\}$ is algebraic with an annihilating polynomial $p(w, x_\Sigma)$, also its derivative $s' \in \mathbb{R}\{\!\{x\}\!\}$ is algebraic. Moreover, one can find effectively ([11], [33]) a rational function $r(x,y)$ such that $s'(x)$ can be computed from s via $s'(x) = r(x, s(x))$.

▶ Remark 18. The coefficients a_n of a unary algebraic fps $a(x) = \sum_n a_n x^n$ whose annihilating polynomial $p(w,x)$ satisfies $p(0,0) = 0$, $\partial_w p(0,0) \neq 0$ and $p(0,z) \neq 0$ can be computed directly via a "multinomial formula" [6]. More generally, since all algebraic functions are *D-finite* [33], one can deduce a linear recursive equation with polynomial coefficients $p_0(n)a_n + p_1(n)a_{n+1} + \dots + p_r(n)a_{n+r} = 0$ that can be used to compute the a_n effectively.

The theory of algebraic power series should be stated in a slightly different way in the context of semirings: if $a \in \mathcal{R}\{\!\{\Sigma\}\!\}, a > 0$, then for each k $a^k > 0$, hence it is not possible that $p(a, x_\Sigma) = 0$ for some $p \in \mathcal{R}\{\Sigma\}$. Definition 14 must thus be adapted as follows:

▶ **Definition 19** ([37]). *Let $\mathcal{R}$ be a semiring and let $\mathcal{R}' \leq \mathcal{R}$ be a subsemiring. A* $\mathcal{R}'$-fixpoint algebraic system *(in short* $\mathcal{R}'$-FAS*) with parameters x_Σ is a system of n equations:*

$$\begin{cases} w_1 = p_1(w_1, \dots w_n, x_\Sigma) \\ \dots \\ w_n = p_n(w_1, \dots w_n, x_\Sigma) \end{cases} \tag{7}$$

A family $(s_1, \dots s_n)$ of formal power series $s_i \in \mathcal{R}\langle\!\langle \Sigma \rangle\!\rangle$ that is the minimal solution of a $\mathcal{R}'$-FAS is called a $\mathcal{R}'$-fixpoint algebraic family*. If the p_i are such that $p_i(0,0) = 0$ and, for each j, the coefficient of the monomial w_j in p_i is 0, we say that the system is* proper.

The theory of systems like (7) has been extensively studied ([37], [6]) in the context of combinatorics and formal language theory: in particular, it can be shown that they admit a (unique) minimal solution, that can be obtained by iterating the polynomials $p_1 \dots p_n$. In the following we will be in general be interested in $\mathbb{R}^{+\infty}_{\geq 0}$-power series algebraic over $\mathbb{Q}_{\geq 0}$: in this case, the polynomial equations $w_i = p_i$, $p_i \in \mathbb{Q}_{\geq 0}\{\Sigma\}$ can be rewritten as $p'_i := p_i - w_i = 0$, $p \in \mathbb{Q}_{\geq 0}\{\Sigma\}$. Rather than speaking of the minimal solution of (7) as a system with coefficients in $\mathbb{Q}_{\geq 0}$, we can then speak of the minimal non-negative solution $(s_1, \dots s_n)$ of the system $(p'_i = 0)_{1 \leq i \leq n}$ with coefficients over $\mathbb{Q}$. In this case, if the FAS is proper, even more is true: every s_i part of a $\mathbb{Q}^+$-algebraic family is algebraic:

▶ **Theorem 20** (Variable Elimination, [37], Thms. 16.9, 16.10)**.** *Let F be a proper $\mathbb{Q}_{\geq 0}$-FAS consisting of n equations with parameters x_Σ and let $(s_1, \dots s_n)$ be its minimal solution. From F we can compute irreducible polynomials $(q_i(w', x_\Sigma))_{1 \leq i \leq n}$ such that: $q_i(s_i(x_\Sigma), x_\Sigma) = 0$, for all $1 \leq i \leq n$, that is, every s_i is a $\mathbb{Q}$-algebraic power series.*

▶ Remark 21. The definition of FAS consists of two conditions: the purely algebraic fact of being solution of a FAS and the minimality with respect to the order relation. For *proper* FAS, the minimal solution $(s_1(z), \dots s_n(z))$ is also the unique solution in a neighborhood of $z = 0$ [37]. However, the algebraic functions $s_1(z), \dots s_n(z)$ might have singularities over the disk $|z| \leq 1$; in this case there might be multiple solutions of the FAS for $z = 1$, so we must require minimality. As an example, take the equation $w = zw$: the only solution on the open disk $|z| < 1$ is the identically zero power series $w(z) = 0$, but for $z = 1$ we find that any $w \in \mathbb{R}$ is a valid solution.

Over any semiring $\mathcal{R}$, if $a(x), b(x) \in \mathcal{R}\langle\!\langle x_\Sigma \rangle\!\rangle$ are algebraic, then so are their sum $a(x) + b(x)$, multiplication $a(x)b(x)$, and derivative $a'(x)$ [33]. If $\mathcal{R}$ is continuous, the same holds of their composition $a(b(x))$ (while, as we saw in Section 4, composition is not always well-defined over arbitrary (semi)rings).

▶ Remark 22. When $a(x) = \sum_{\mu \in !\Sigma} a_\mu x^\mu \in \mathbb{R}^{+\infty}_{\geq 0}\langle\!\langle x_\Sigma \rangle\!\rangle$ is part of a $\mathbb{Q}^+$-proper algebraic family, we can refine what we said in Remark 16: as a consequence of ([37], Thm. 16.35), the support $\mathsf{supp}\, a = \{\mu \mid a_\mu \neq 0\}$ is *semilinear*, that is, it is the union of finitely many sets of the form $\{n_1\mu_1 + \dots + n_k\mu_k \mid n_1, \dots n_k \in \mathbb{N}\}$.

5.2 From PHORS to FAS

We are interested in the case in which the fixpoint equations that define (the interpretation of) a PHORS in $\mathcal{R}\mathbf{Rel}_!$ form a FAS and thus yield, as solution, an algebraic power series. Recall that higher-order types $T \to U$ are interpreted by *infinitely many* variables and, consequently, higher-order terms translate into infinitely many equations. Our goal is to find ways to reduce such infinitary systems to more finitary (and manageable) ones.

As we saw, a PHORS $\mathcal{G}$ is interpreted via the fps $a_L(z)(x_\Sigma)$ interpreting each non-terminal $L \in \mathcal{N}$; these fps are collectively defined as the minimal solutions of the family of equations $(r_x = (\llbracket t_\mathcal{G} \rrbracket)_x(r_x, z))_{x \in X}$ induced by the morphism interpreting $t_\mathcal{G} : \mathcal{N} \to \mathcal{N}$.

When $\mathcal{G}$ is order-1, this system can *always* be reduced to a FAS:

▶ **Proposition 23** (order-1 PHORS are algebraic). *For all order-1 PHORS $\mathcal{G} = (\mathcal{N}, \mathcal{R}, S)$ and non-terminal symbol $L \in \mathcal{N}$, the fps $a_L(z)(x_1, \ldots, x_n)$ is $\mathbb{Q}$-algebraic.*

Proof. When $\mathcal{G}$ is order-1, each type $\mathcal{N}(L_i)$ is of the form $o \to \cdots \to o \to o$: hence $a_L(z)(x_1, \ldots, x_n)$ can be written in the form $\sum_{(p_1,\ldots,p_n) \in \mathbb{N}^n} w_{p_1,\ldots,p_n}(z) x_1^{p_1} \ldots x_n^{p_n}$. Thanks to Lemma 12 we have that, if for some $i = 1, \ldots, n$, $p_i \geq 2$, then $w_{p_1,\ldots,p_n}(z) = 0$. We can thus reduce ourselves to a finite polynomial system only involving the finitely many $w_{p_1,\ldots,p_n}(z)$, with the $p_i \in \{0, 1\}$. ◀

Since, as we will see with Theorem 41, AST and PAST can be decided once a PHORS is interpreted as a FAS, we see that the well-known decidability of order-1 PHORS naturally follows from the simple nature of their generating functions, which are always affine.

This argument does not work beyond order-1, since PHORS can be truly non-linear, and their interpretation thus infinitary. We introduce below a class of systems satisfying a suitable form of finiteness.

▶ **Definition 24** (variable and family restrictions). *For all fps $s \in \mathcal{R}\{\!\{\Sigma\}\!\}$ and $\Sigma' \subset \Sigma$, let $\in \mathcal{R}\{\!\{\Sigma\}\!\}$ be the fps defined by $s|_{\Sigma'}(x_{\Sigma'}, x_{\Sigma - \Sigma'}) = s(x_{\Sigma'}, 0)$.*

Intuitively, in $s|_{\Sigma'}$ we only keep those monomials $s_\mu x^\mu$ from s such that μ has support included in Σ', and we "kill" all monomials $s_\mu x^\mu$ containing some x^i, with $i > 0$ and $x \notin \Sigma'$.

▶ **Definition 25** (finitary fps). *Let $s_X \in \mathcal{R}\{\!\{X + Y\}\!\}^Z$ be a family of fps. A pair (U, V), where $U \subseteq X$ and $W \subseteq Z$, is called:*

- stable *if for all $\sigma \in Z - W$, $s_\sigma|_{U+Y} = 0$;*
- polynomial *if for all $\sigma \in W$, $s_\sigma|_{U+Y}$ has finite support (i.e. it is a polynomial).*

A family $s_X \in \mathcal{R}\{\!\{X + Y\}\!\}^X$ is called finitary *if there exists some finite $W \subset_{\text{fin}} X$ such that (W, W) is stable and polynomial. If, moreover, for each $\sigma \in W$ and for each $y \in Y$, $s_\sigma(0) = 0$ and the coefficient of y in s_σ is 0, we say that s_X is a* finitary proper family.

Take a family $s_X \in \mathcal{R}\{\!\{X + Y\}\!\}^X$ and let $A \subseteq X$ be such that (A, A) is a stable pair. By stability, starting from 0 and repeatedly applying s_X we can never obtain a term of the form $s_\sigma(t)$, with $\sigma \notin A$. Moreover, if A is also polynomial, then all such iterates are obtained by composing polynomials $s_\sigma|_{A+Y}$, with $\sigma \in A$; finally, if A is finite, these polynomials are only finitely many. All this leads to the following:

▶ **Proposition 26.** *Let $\mathcal{R}' \leq \mathcal{R}$. If $s_X \in \mathcal{R}'\{\!\{X + Y\}\!\}^X$ is a finitary family, then* fix s_X *has finite support and it is the minimal solution of a fixpoint algebraic system over $\mathcal{R}'$.*

Proof. Let $A \subseteq X$ be finite, stable and polynomial as above. Take the (finite) system of equations: $(r_a = s_a^*(r_a, x_Y))_{a \in A}$, where $s_a^* = (s_a)|_{A+Y}$, and let $\bar{r}_A$ be one of its solutions. Then, define $(t_x)_{x \in X}$ to be $\bar{r}_x$ if $x \in A$ and 0 otherwise. Then for all $x \in X$, $s_x(t_x, x_Y) = 0$: if $x \in A$, since $t_X = \bar{r}_A \sqcup 0_{X-A}$, we have $s_x(t_x, x_Y) = s_x^*(\bar{r}_x, x_Y) = 0$; if $x \notin A$, then $(s_x)|_{A+Y} = 0$, which again implies $s_x(t_X, x_Y) = 0$. ◀

Combining Proposition 26 and Theorem 20, we also get:

▶ **Corollary 27.** *If $s_X \in \mathcal{R}'\{\!\{X + Y\}\!\}^X$ is a proper finitary family, then for each $x \in X$ $(\text{fix}\, s_X)_x$ is an algebraic power series. Moreover, for all but finitely many x, $(\text{fix}\, s_X)_x = 0$.*

▶ **Example 28.** Let $X = \mathbb{N} + 1$, $Y = \{z\}$ and $s_X \in \mathbb{R}^{+\infty}_{\geq 0}\{\!\{X + Y\}\!\}^X \equiv (\mathbb{R}^{+\infty}_{\geq 0}\{\!\{z\}\!\})\{\!\{X\}\!\}^X$ be the fps from Example 11. Then one can easily check that $W = \{1, \star\} \subset X$ is stable and polynomial, so the family is finitary. In other words, the fixpoint $a = \mathsf{fix}\, s_X$ can be computed by considering only the equations $a_1(z) = s_1^F(a_1(z), z)$ (i.e. (6)) and $a_{\mathcal{G}} = a_\star = a_1$. Hence $a_{\mathcal{G}}(z) \in \mathbb{R}^{+\infty}_{\geq 0}\{\!\{z\}\!\}$ is algebraic over $\mathbb{Q}^+$ (with parameter z).

6 Finitely Bounded PHORS

In this section we introduce a first class of PHORS whose relational interpretation yields a finitary family of fps. As a consequence, we establish the algebraicity of the corresponding probabilistic generating functions and the decidability of the (P)AST problems.

6.1 Syntax of PBHORS$^{<\infty}$

We use types defined by the grammar below:

$$\varphi, \psi := o^n \mid !_k\varphi \multimap \varphi \qquad (1 \leq n \in \mathbb{N},\ k \in \mathbb{N}),$$

where o^n indicates the n-fold product $o \times \cdots \times o$. Let τ be an abbreviation for the affine function type $!_1 o \multimap o$. the order relation $\varphi \sqsubseteq \psi$ between types is defined by induction by

$$\frac{}{o^n \sqsubseteq o^n} \qquad \frac{\xi \sqsubseteq \xi', \varphi \sqsubseteq \varphi'}{\xi \multimap \varphi \sqsubseteq \xi' \multimap \varphi'} \qquad \frac{k \leq h \quad \varphi \sqsubseteq \psi}{!_k\varphi \sqsubseteq !_h\psi}$$

Type judgements are expressions of the form $\mathcal{N} \mid \Delta \vdash t : \varphi$, involving two kinds of contexts (as in [42]):

- non-linear contexts, noted $\mathcal{N}, \mathcal{N}'$ will be used for non-terminal symbols; these are finite sets of type bindings of the form $L : \varphi$; intuitively, we put no restrictions on the number of times a non-terminal is used;
- graded contexts, noted Δ, Δ' will be used for the arguments to be passed to the non-terminals; they are finite sets of type bindings of the form $x :_k \varphi$. Intuitively, the binding $x :_k \varphi$ means that we can use x with type φ *at most* k times.

We define by induction the operations $\Delta + \Delta', k\Delta$ on graded contexts:

$$\Delta + \emptyset = \Delta, \qquad k\emptyset = \emptyset,$$
$$(\Delta, x :_k \varphi) + (\Delta', x :_h \varphi) = (\Delta + \Delta'), x :_{k+h} \varphi, \qquad k(\Delta, x :_h \varphi) = k\Delta, x :_{kh} \varphi,$$

where, as in e.g. [4], we are assuming that Δ and Δ' agree on each variable (i.e. they contain the same type bindings but may disagree on the corresponding coefficients).

The typing rules are in Fig. 1.

▶ **Definition 29** (PBHORS$^{<\infty}$)**.** *A* finitely bounded PHORS *(noted PBHORS$^{<\infty}$) is a triple $\mathcal{G} = (\mathcal{N}, \mathcal{D}, S)$, where $\mathcal{N}$ is a finite set of typed non-terminals, $S \in \mathcal{N}$ is the start symbol such that $\mathcal{N}(S) = o$, and $\mathcal{D}$ associates each $L \in \mathcal{N}$ with a derivation $\mathcal{N} \mid \emptyset \vdash \lambda x_1 \ldots . x_n . t : \varphi$ such that $\varphi \sqsubseteq \mathcal{N}(L)$ and $\mathcal{N} \mid x_1, \ldots, x_k \vdash t : o$.*

Since affine implication is expressed by $!_1\varphi \multimap \psi$, it is immediate that the affine PHORS of [42] are PBHORS$^{<\infty}$, so all our results translate automatically to them. In particular, as the PHORS from Example 3 is affine, it is also a PBHORS$^{<\infty}$.

▶ **Example 30.** The $\mathcal{G}$ from (3) in Section 2 is not affine, but is a PBHORS$^{<\infty}$: letting $\mathcal{N}(A) = \mathcal{N}(B) = \tau$ and $\mathcal{N}(H) = !_2\tau \multimap \tau$, we can correctly type $\mathcal{N} \mid \emptyset \vdash \lambda f.\lambda x.t_H : \mathcal{N}(H)$, where $t_H = (H(A \circ f)x \oplus_a H(B \circ f)x) \oplus_a f(fx)$.

$$\frac{x :_p \varphi \in \Delta, \quad 1 \leq p}{\mathcal{N} \mid \Delta \vdash x : \varphi}\ ax1 \qquad \frac{L : \varphi \in \mathcal{N}}{\mathcal{N} \mid \Delta \vdash L : \varphi}\ ax2 \qquad \frac{\mathcal{N} \mid \Delta \vdash t : o \quad \mathcal{N} \mid \Delta \vdash t' : o}{\mathcal{N} \mid \Delta \vdash t \oplus t' : o}\ \oplus$$

$$\frac{\mathcal{N} \mid \Delta, x :_k \varphi \vdash t : \psi}{\mathcal{N} \mid \Delta \vdash \lambda x.t :!_k\varphi \multimap \psi}\ \lambda \qquad \frac{\mathcal{N} \mid \Delta \vdash t :!_k\varphi \multimap \tau \qquad \mathcal{N} \mid \Delta' \vdash u : \varphi}{\mathcal{N} \mid \Delta + k\Delta' \vdash tu : \psi}\ @$$

$$\frac{\mathcal{N} \mid \Delta \vdash t_i : o \quad i = 1, \ldots, n}{\mathcal{N} \mid \Delta \vdash \langle t_1, \ldots, t_n \rangle : o^n}\ \langle\rangle \qquad \frac{\mathcal{N} \mid \Delta \vdash t : o^n \quad i = 1, \ldots, n}{\mathcal{N} \mid \Delta \vdash \pi_i t : o}\ \pi_i$$

Figure 1 Typing rules of the PBHORS$^{<\infty}$.

▶ **Example 31** (a non-algebraic PHORS)**.** The PHORS $\mathcal{G}$ below:

$$\begin{aligned} Lfx &= L(f \circ f)x \oplus_p fx \\ S &= L\ \mathrm{id}\ e \end{aligned}$$

is *not* a PBHORS$^{<\infty}$: if we try to type $t_L = L(y \circ y)x \oplus_p fx$ with $\mathcal{N}(L) =!_k\tau \multimap \tau$, we obtain $\mathcal{N} \mid \emptyset \vdash \lambda y.\lambda x.t_L : \psi$, where $\psi = !_{2k}\tau \multimap \tau \not\sqsubseteq \mathcal{N}(L)$. Indeed, the generating function $s^L(y_{\mathbb{N}}, x) \in \mathcal{R}\{\!\{\mathbb{N}+1\}\!\}$ of L satisfies $s^L(y_{\mathbb{N}}, x) = \frac{1}{2}(y_1 x + s^L(y^2_{\mathbb{N}}, x))$, which yields the solution $s^L(y_{\mathbb{N}}, x) = \sum_i \frac{1}{2^{i+1}} y^{2^i} x$, that is not algebraic (cf. Remark 16).

While the PBHORS$^{<\infty}$ need not be linear, they can be *linearized*: we can turn them into PAHORS having the same semantics and branch language. Notice, however, that the resulting affine PHORS might have size exponential in the original one. More precisely, define the *size* of a $\mathcal{G}$ as the number of its non-terminals times the maximum size of any of its terms, i.e. $\|\mathcal{G}\| = |\mathcal{N}| \times \max\{|\mathcal{D}(L)| \mid L \in \mathcal{N}\}$, and let $\partial\mathcal{G}$ be the maximum grade occurring in $\mathcal{G}$. Then we have:

▶ **Theorem 32** (Linearization)**.** *For every PBHORS$^{<\infty}$ $\mathcal{G} = (\mathcal{N}, \mathcal{D}, S)$ there exists a PAHORS $\mathcal{G}_{\mathrm{aff}} = (\mathcal{N}, \mathcal{D}', S)$ with size $\|\mathcal{G}_{\mathrm{aff}}\| \leq \partial\mathcal{G}\|\mathcal{G}\|$ such that $a_{\mathcal{G}} = a_{\mathcal{G}_{\mathrm{aff}}}$ and $\mathcal{L}(\mathcal{G}) = \mathcal{L}(\mathcal{G}_{\mathrm{aff}})$.*

Proof sketch. Define φ_{aff} as $(o^n)_{\mathrm{aff}} = o^n$ and $(!_k\varphi \multimap \psi)_{\mathrm{aff}} = \varphi_{\mathrm{aff}} \multimap \cdots \multimap \varphi_{\mathrm{aff}} \multimap \psi_{\mathrm{aff}}$, with φ_{aff} repeated k times. Then, by induction, any derivation of $\mathcal{N} \mid \Delta \vdash t : \varphi$ yields a derivation of $\mathcal{N}_{\mathrm{aff}} \mid \Delta' \vdash t : \varphi_{\mathrm{aff}}$ in the affine system of [42], where $|\Delta'| \leq \partial\mathcal{G}|\Delta|$: an order-$n$ PBHORS$^{<\infty}$ yields then an order-n PAHORS in which every equation $Lx_1 \ldots x_n$ translates into a new equation $Lz^1_1 \ldots z^{k_n}_n$, where, if $\mathcal{N}(L) =!_{k_1}\varphi_1 \multimap \cdots \multimap !_{k_i}\varphi_i \multimap \cdots \multimap o^n$, each variable x_i is replaced by k_i variables $z^1_i, \ldots, z^{k_i}_i$. ◀

▶ **Example 33.** The linearization of the PBHORS$^{<\infty}$ from Example 30 yields the PAHORS below:

$$\begin{aligned} Lf_1f_2x &= (L(A \circ f_1)(A \circ f_2)x \oplus_a L(B \circ f_1)(B \circ f_2)x) \oplus_a f_1(f_2x) \\ Ax &= x \oplus_b \Omega \\ Bx &= x \oplus_c \Omega \\ S &= HIIe \end{aligned}$$

Notice that the unique functional variable f, that was used twice, is now replaced by *two* functional variables f_1, f_2, used once.

▶ **Example 34.** $\|\mathcal{G}_{\mathrm{aff}}\|$ may be exponential in $\|\mathcal{G}\|$: take the order-2 PBHORS$^{<\infty}$ $\mathcal{G}$ given by

$$\begin{aligned} F_1 f x &= f(fx) \\ F_2 f x &= F_1(f \circ f)x \oplus_p x \\ &\vdots \\ F_n f x &= F_{n-1}(f \circ f)x \oplus_p x \\ S &= F_n \ \mathrm{id}\ e \end{aligned}$$

As $\mathcal{N}(F_n) = !_{2^n}\tau \multimap \tau$, $\|\mathcal{G}_{\mathrm{aff}}\| \leq \partial\mathcal{G}\|\mathcal{G}\| = 2^{\mathcal{O}(\|\mathcal{G}\|)}\|\mathcal{G}\|$.

6.2 From PBHORS$^{<\infty}$ to FAS

We will now show how, for any PBHORS$^{<\infty}$ $\mathcal{G}$, the interpretation of the underlying PHORS yields a finitary FAS, whose minimal solution is thus an algebraic family.

First, from any PBHORS$^{<\infty}$ we can canonically extract the underlying (simply typed) PHORS. Observe first that the rules (and indeed the derivations) of PBHORS$^{<\infty}$ can be seen as *quantitative refinements* of the simply typed ones. Indeed, every type φ is (uniquely) a *refinement* of some simple type φ_{st}: the simple types φ_{st} are defined by $(o^n)_{\mathrm{st}} = o^n$ and $(!_k\varphi \multimap \psi)_{\mathrm{st}} = \varphi_{\mathrm{st}} \to \psi_{\mathrm{st}}$. Notice that $\varphi \sqsubseteq \psi$ implies $\varphi_{\mathrm{st}} = \psi_{\mathrm{st}}$. Moreover, for every derivation $\pi : \mathcal{N} \mid \Delta \vdash t : \varphi$, by replacing each ψ by ψ_{st}, we obtain, by induction, a simply typed derivation $\pi_{\mathrm{st}} : \mathcal{N}_{\mathrm{st}}, \Delta_{\mathrm{st}} \vdash t : \varphi_{\mathrm{st}}$.

For any set Σ and $k \in \mathbb{N}$, let $!_k\Sigma \subseteq !\Sigma$ be the set of multisets μ of maximal multeplicity k (i.e $\max_{x\in\Sigma} \mu(x) \leq k$). $!_k\Sigma$ corresponds to the monomials over x_Σ of degree at most k in each variable. Define then, for every type φ, $[\![\varphi]\!]$ as $[\![o^n]\!] = 1$ and $[\![!_k\varphi \multimap \psi]\!] = !_k[\![\varphi]\!] \times [\![\psi]\!]$. We extend this to contexts by $[\![\emptyset]\!] = 0$ and $[\![\mathcal{N}, x : \varphi]\!] = [\![\mathcal{N}]\!] + [\![\varphi]\!]$ (irrespectively of grades). Notice that $[\![\varphi]\!] \subseteq [\![\varphi_{\mathrm{st}}]\!]$, but the variable sets $[\![\varphi]\!]$ are always finite.

Our general idea is the following: given a PBHORS$^{<\infty}$ $\mathcal{G} = (\mathcal{N}, \mathcal{D}, S)$, from the derivations $\mathcal{D}(L)$ of $\mathcal{N} \mid \emptyset \vdash t_L : \mathcal{N}(L)$, we deduce a corresponding simply typed derivation of $t_{\mathcal{G}} : \mathcal{N}_{\mathrm{st}} \to \mathcal{N}_{\mathrm{st}}$ yielding a family $[\![t]\!]^{\mathcal{R}} \in \mathcal{R}\{\!\{[\![\mathcal{N}_{\mathrm{st}}]\!]\}\!\}^{[\![\mathcal{N}_{\mathrm{st}}]\!]}$. At the same time, we will show that the finite pair $([\![\mathcal{N}]\!], [\![\mathcal{N}]\!])$ is stable and polynomial, and we will conclude that $[\![t_{\mathcal{G}}]\!]$ is finitary, whence, by Corollary 27, algebraic.

The fundamental ingredient is the following lemma:

▶ **Lemma 35** (stability lemma)**.** *Let $\mathcal{R} = \mathbb{R}^{+\infty}_{\geq 0}\{\!\{\Sigma\}\!\}$. For every derivation $\mathcal{N} \mid \Delta \vdash t : \varphi$, the pair $([\![\mathcal{N}]\!] + [\![\Delta]\!], [\![\varphi]\!])$ is stable and polynomial for $[\![t]\!]^{\mathcal{R}} \in \mathcal{R}\{\!\{[\![\mathcal{N}_{\mathrm{st}}]\!] + [\![\Delta_{\mathrm{st}}]\!]\}\!\}^{[\![\varphi_{\mathrm{st}}]\!]}$.*

Proof sketch. Let a *finitary* variable be any $\sigma \in [\![\mathcal{N}]\!], [\![\Delta]\!], [\![\varphi]\!]$, and a *non-finitary* one be any $\sigma \in [\![\mathcal{N}_{\mathrm{st}}]\!] - [\![\mathcal{N}]\!], [\![\Delta_{\mathrm{st}}]\!] - [\![\Delta]\!], [\![\varphi_{\mathrm{st}}]\!] - [\![\varphi]\!]$. By induction, one shows the following:

1. for all non-finitary σ, $([\![t]\!]^{\mathcal{R}})_\sigma\big|_{[\![\mathcal{N}]\!]+[\![\Delta]\!]} = 0$;
2. for all finitary σ, $([\![t]\!]^{\mathcal{R}})_\sigma\big|_{[\![\mathcal{N}]\!]+[\![\Delta]\!]}$ has finite degree in each variable $x_{L:\mathcal{N}(L)}$ and degree $\leq k$ in each variable $x_{f:_k\psi}$. ◀

An immediate consequence of Lemma 35 is the following:

▶ **Proposition 36.** *For every PBHORS$^{<\infty}$ $\mathcal{G} = (\mathcal{N}, \mathcal{D}, S)$, the fps $[\![t_{\mathcal{G}}]\!]^{\mathcal{R}} \in \mathcal{R}\{\!\{[\![\mathcal{N}_{\mathrm{st}}]\!]\}\!\}^{[\![\mathcal{N}_{\mathrm{st}}]\!]}$ is finitary (via the pair $([\![\mathcal{N}]\!], [\![\mathcal{N}]\!])$).*

Thanks to Proposition 26 and the observation that all coefficients of $[\![t_{\mathcal{G}}]\!]^{\mathbb{R}^{+\infty}_{\geq 0}\{z\}}$ belong indeed to $\mathbb{Q}\{z\}$, we obtain then:

▶ **Theorem 37.** *For all PBHORS$^{<\infty}$ $\mathcal{G} = (\mathcal{N}, \mathcal{D}, S)$, $(a_L)_{L\in\mathcal{N}}$ is a proper $\mathbb{Q}^+$-fixpoint algebraic family with parameter z.*

▶ **Corollary 38.** *For all PBHORS$^{<\infty}$ $\mathcal{G} = (\mathcal{N}, \mathcal{D}, S)$,*

1. *for each $L \in \mathcal{N}$, $a_L(z)$ is an algebraic power series.*
2. *$\mathbb{P}(\mathcal{G}\downarrow)$ is a $\mathbb{Q}^+$-algebraic real and $\sum_i \mathbb{P}(\mathcal{G}\downarrow_i)z^i$ is a $\mathbb{Q}^+$-algebraic power series.*

As a consequence of Remark 22 we also obtain:

▶ **Corollary 39.** *For all PBHORS$^{<\infty}$ $\mathcal{G} = (\mathcal{N}, \mathcal{D}, S)$, the set $\{n \mid S$ terminates in n steps$\}$ is semilinear*

▶ Remark 40. By interpreting $t_L \oplus t_R$ as $x[\![t_L]\!] + \overline{x}[\![t_L]\!]$ (cf. Remark 7), the previous corollary could be strengthened to say that the set $\{(n, m) \mid S$ terminates with n left steps and m rigth steps$\}$ is also semilinear. This could also been deduced by the fact that the branch language of a linear HORS is *multiple context-free* [13], and languages of this class are known to be semilinear [46], but it is still worth noticing that we reproved this fact via generating functions.

We conclude by deducing the decidability of AST and PAST:

▶ **Theorem 41.** *AST and PAST are decidable for any PBHORS$^{<\infty}$.*

Proof sketch. The proof of Theorem 37 is constructive: it gives a way to build effectively a FAS $(\vec{x} = p_i(\vec{x}))_{1\leq i\leq n}$ that has $(a_L(z))_{L\in\mathcal{N}}$ as its (minimal) solution. We can then check if $\mathbb{P}(\mathcal{G}\downarrow) = a_{\mathcal{G}}(1) = 1$ via the first order Existential Theory of the Reals **ETR** [20]: first, we create a first order formula $\phi(\vec{x})$ expressing the fact that $\vec{x}$ satisfies $\vec{x} = p_i(\vec{x})$, then we can express the fact that $x_1 = 1$ by the formula $\exists\vec{x}(x_1 = 1 \wedge \phi(\vec{x}) \wedge \forall\vec{y}((\vec{y} < \vec{x}) \implies \neg\phi(\vec{y})))$.

As for PAST, remember that PAST implies AST. Then, given a $\mathcal{G} = (\mathcal{N}, \mathcal{D}, S) \in$ PBHORS$^{<\infty}$, we can first test if it is AST: if it is not, then it is not PAST. If it is AST, notice that, by Corollary 38, $a_{\mathcal{G}}(z)$ is an algebraic power series. Then, we can, as pointed out in Remark 17, effectively find a rational function $r(z, y)$ such that $a'_{\mathcal{G}}(z) = r(z, a_{\mathcal{G}}(z))$. But then $\mathbb{E}(\mathcal{G}\downarrow) = a'_{\mathcal{G}}(1) = r(1, 1)$. ◀

▶ Remark 42. The complexity of our decision procedure for AST and PAST can be reconstructed as follows: for an order n PHORS with non-terminals of maximal grade k and maximal arity r, the set $[\![\mathcal{N}]\!]$ has at most $\exp_k(n) \cdot r^n$ elements, with $\exp_a(b)$ the tower of exponential of base a and height b. As $r \leq \|\mathcal{G}\|$, for each non-terminal L, the polynomials $[\![t_L]\!]^{\mathbb{R}^{+\infty}_{\geq 0}\{\!\{z\}\!\}}$ involves at most $\exp_k(n)\|\mathcal{G}\|^n$ variables and they can be computed by at most $|t_L|$ products or sum of polynomials: the total cost of these operations will be $O(\exp_k(n)^2\|\mathcal{G}\|^{2n+1})$ and we have to perform them for each non terminal: the total time complexity of computing the FAS system is then $O(\exp_k(n)^2\|\mathcal{G}\|^{2n+2})$. Using a polynomial space algorithm to decide **ETR** [20], we can then see that the space complexity of deciding AST with our method *for fixed order PHORS* is polynomial in their size $\|\mathcal{G}\|$, but *grows superexponentially* when the order is not fixed. Since the algorithm to compute the rational function of Remark 17 is polynomial [11], we can conclude the same for PAST.

▶ Remark 43. Beyond AST/PAST, the fact that $a_{\mathcal{G}}(z) = \sum_{i=0}^{\infty} \mathbb{P}(\mathcal{G}\downarrow_i)z^i$ is algebraic can be used to make asymptotic estimates on the coefficients $\mathbb{P}(\mathcal{G}\downarrow_i)$ for $i \to +\infty$ (cf. Remark 18).

▶ Remark 44. Let X be a random variable representing the number of steps to termination of a PBHORS$^{<\infty}$ $\mathcal{G}$. When $\mathcal{G}$ is AST, the method in the proof of Theorem 41 can be used to compute, beyond the first moment $\mathbb{E}[X] = \mathbb{E}[\mathcal{G}\downarrow]$, also the second moment $\mathbb{E}[X^2]$ (via the second derivative $a''(1)$). This can be used to estimate several probabilistic properties of the reduction of $\mathcal{G}$, e.g. how likely is the running time to be far from its mean value.

7 Bounded PHORS

We now introduce a second class of PHORS that extends the PBHORS$^{<\infty}$ by admitting infinite grades $!_\infty$, corresponding to parameters that may be used an *unbounded* number of times. In this way, thanks to the composability of the underlying algebraic power series, the PBHORS$^{\infty}$ are proved to be closed under arbitrary compositions.

7.1 Syntax of PBHORS$^{\infty}$

While the type discipline of PBHORS$^{<\infty}$ ensures that all variables are used in a bounded way, the variable z in $a_{\mathcal{G}}(z)$, which counts the number of probabilistic choices (e.g. think again of the HORS (2)), can be thought of as an input that is used *unboundedly* during reductions, yet without compromising algebraicity.

The key insight here is that we are describing PHORS via families of power series with coefficients taken in $\mathbb{R}^{+\infty}_{\geq 0}\{\!\{z\}\!\}$: in other words, the variable z is treated as a *formal parameter* all along the way. Our goal is then to capture those situations in which a variable is used as a parameter and can, thus, be duplicated unrestrictedly.

▶ **Example 45.** Consider the following PHORS $\mathcal{G}$:

$$\begin{aligned} Lfgx &= f(Lfg(Lfgx)) \oplus_a gx \\ Ax &= x \oplus_b \Omega \\ Bx &= x \oplus_c \Omega \\ S &= LABe \end{aligned}$$

Define the parenthesis symbols "(" and ")" as, respectively, ab and ac; then the branch language $\mathcal{L}(\mathcal{G})$ is formed by all words of the form $w) \in \{(,)\}^*$, where w is "well-parenthesized", like e.g. $w = (()(()))$, $w = (()(()))())$, i.e. it essentially corresponds to the *Dyck language*. Notice that $\mathcal{G}$ is not a PBHORS$^{<\infty}$: if we try to type the term $t_L = \lambda fgx.f(Lfg(Lfgx)) \oplus_a gx$ with finite grades, setting $\mathcal{N}(L) = !_m\tau \multimap !_n\tau \multimap \tau$, we obtain $\mathcal{N} \mid \emptyset \vdash t_L : \varphi$, where $\varphi = !_{2m+1}\tau \multimap !_{2n}\tau \multimap \tau \not\sqsubseteq \mathcal{N}(L)$. The only way out is to assign infinite grades $n, m = \infty$, i.e. to let $\mathcal{N}(L) = !_\infty\tau \multimap !_\infty\tau \multimap \tau$.

The interpretation of L will be some power series $a_L(z, f, g, x) = \sum_{ijk} L_{ijk} f^i g^j x z^k$. Now, observe that $a_L(z, f, g, x)$ can be rewritten as a power series $a_L(z, f, g)(x)$ with coefficients in $\mathbb{R}^{+\infty}_{\geq 0}\{\!\{z, f, g\}\!\}$, that is a solution (letting e.g. $a = \frac{1}{2}$) of the algebraic equation

$$a_L(z, f, g) = \frac{z f a_L(z, f, g)^2 + zg}{2}. \tag{8}$$

This equation is reminiscent of (6) and can indeed be solved in a similar way, yielding $a_L(z, f, g, x) = \sum_{i=0}^{\infty} \frac{C_i}{2^{2i+1}} f^i g^{i+1} x z^{2i+1}$: there are C_i well-parenthesized words of length $2i + 1$, each using (i times and) $i + 1$ times, each of them produced by a reduction making $2i + 1$ choices, and thus having probability $\frac{1}{2^{2i+1}}$.

We now introduce a type discipline that captures situations like the one just sketched. We use two different kinds of types:

$$\varphi, \psi := o^n \mid !_k\varphi \multimap \varphi \qquad (1 \leq n \in \mathbb{N}, k \in \mathbb{N}) \qquad \Phi, \Psi := \varphi \mid !_k\varphi \multimap \Phi \qquad (k \in \mathbb{N} \cup \{\infty\})$$

The types φ, ψ are called *finitary* and are precisely those of PBHORS$^{<\infty}$; the types Φ, Ψ are called *infinitary*. The order relation on infinitary type $\Phi \sqsubseteq \Psi$ is inherited from the one for infinite types together with the new rule $\varphi \sqsubseteq \varphi', \Psi \sqsubseteq \Psi', k \leq h \Rightarrow !_k\varphi \multimap \Psi \sqsubseteq !_h\varphi' \multimap \Psi'$

$$\frac{x :_p \varphi \in \Delta, \quad 1 \leq p}{\mathcal{N} \mid \mathcal{P} \mid \Delta \vdash x : \varphi}\ ax_\Delta \qquad \frac{w :_\infty \varphi \in \mathcal{P}}{\mathcal{N} \mid \mathcal{P} \mid \Delta \vdash w : \varphi}\ ax_\mathcal{P} \qquad \frac{L : \Phi \in \mathcal{N}}{\mathcal{N} \mid \mathcal{P} \mid \Delta \vdash L : \Phi}\ ax_\mathcal{N}$$

$$\frac{\mathcal{N} \mid \mathcal{P} \mid \Delta, x :_k \varphi \vdash t : \Psi}{\mathcal{N} \mid \mathcal{P} \mid \Delta \vdash \lambda x.t :_k \varphi \multimap \Psi}\ \lambda \qquad \frac{\mathcal{N} \mid \mathcal{P} \mid \Delta \vdash t :_k \varphi \multimap \Psi \qquad \mathcal{N} \mid \mathcal{P} \mid \Delta' \vdash u : \varphi}{\mathcal{N} \mid \mathcal{P} \mid \Delta + k\Delta' \vdash tu : \Psi}\ @$$

$$\frac{\mathcal{N} \mid \mathcal{P}, w :_\infty \varphi \mid \emptyset \vdash t : \Psi}{\mathcal{N} \mid \mathcal{P} \mid \emptyset \vdash \lambda w.t :_\infty \varphi \multimap \Psi}\ \lambda_\infty \qquad \frac{\mathcal{N} \mid \mathcal{P} \mid \Delta \vdash t :_\infty \varphi \multimap \Psi \qquad u : \varphi \in \mathcal{N}' \cup \mathcal{P}}{\mathcal{N} \cup \mathcal{N}' \mid \mathcal{P} \mid \Delta \vdash tu : \Psi}\ @_\infty$$

$$\frac{\mathcal{N} \mid \mathcal{P} \mid \Delta \vdash t_i : o \quad i = 1, \ldots, n}{\mathcal{N} \mid \mathcal{P} \mid \Delta \vdash \langle t_1, \ldots, t_n \rangle : o^n}\ \langle\rangle \qquad \frac{\mathcal{N} \mid \mathcal{P} \mid \Delta \vdash t : o^n \quad i = 1, \ldots, n}{\mathcal{N} \mid \mathcal{P} \mid \Delta \vdash \pi_i t : o}\ \pi_i$$

$$\frac{\mathcal{N} \mid \mathcal{P} \mid \Delta \vdash t : o \quad \mathcal{N} \mid \mathcal{P} \mid \Delta \vdash t' : o}{\mathcal{N} \mid \mathcal{P} \mid \Delta \vdash t \oplus t' : o}\ \oplus$$

Figure 2 Typing rules of PBHORS$^{<\infty}$.

The typing rules will use three kinds of contexts:

- the non-linear context $\mathcal{N}$, made of infinitary bindings of the form $L : \Phi$;
- a new parameters context $\mathcal{P}$, made of finitary bindings of the form $w :_\infty \varphi$;
- a graded context Δ, made of finitary bindings of the form $x :_k \varphi$, with $k < \infty$.

The rules are illustrated in Fig. 2. The rules for finitary types are as in the previous section; the new rules are $\text{ax}_\mathcal{P}$ and, notably, the infinitary abstraction and application rules λ_∞ and $@_\infty$, which ensure that an infinitary function $t :!_\infty \varphi \multimap \Psi$ can only be applied to either a non-terminal or a parameter. Recall that in a PHORS each non-terminal L is equated with a corresponding term t_L: so, when we apply a function $t :!_\infty \varphi \to \Psi$ to the symbol L, we are actually applying t to the corresponding term t_L.

▶ **Definition 46** (PBHORS$^\infty$). *A* (parametric) bounded PHORS *(noted PBHORS$^\infty$) is a tuple $\mathcal{G} = (\mathcal{N}, \mathcal{P}, \mathcal{D}, S)$, where $\mathcal{N}$ is a finite set of typed non-terminals $L : \Phi$, $\mathcal{P}$ is a finite set of (finitarily) typed parameters $w :_\infty \varphi$, $S \in \mathcal{N}$ is the start symbol such that $\mathcal{N}(S) = o$, and $\mathcal{D}$ is a function that associates each $L \in \mathcal{N}$ with a derivation $\mathcal{N} \mid \mathcal{P} \mid \emptyset \vdash \lambda x_1 \ldots . x_n.t : \Phi$ such that $\Phi \sqsubseteq \mathcal{N}(L)$ and $\mathcal{N} \mid x_1, \ldots, x_k \vdash t : o$.*

A PBHORS$^\infty$ $\mathcal{G}$ may contain *open* parameters $w :_\infty \varphi \in \mathcal{P}$. We call $\mathcal{G}$ *closed* if $\mathcal{P} = \emptyset$.

The PHORS from Example (45) is a closed PBHORS$^\infty$, with $\mathcal{N}(L) = !_\infty \tau \multimap !_\infty \tau \multimap \tau$. Instead, the non-algebraic PHORS from Example (31) is not a PBHORS$^\infty$: while we can give type $\mathcal{N}(L) = !_\infty \tau \to \tau$ to L, we cannot be apply it to the function $f \circ f : \tau$, as the latter is neither a parameter nor a non-terminal.

▶ **Example 47.** One can check that all PHORS in ([36], Examples 2.9-2.11) are PBHORS$^\infty$, but not PBHORS$^{<\infty}$, e.g. the (non-AST) tree-generation program below:

$$\begin{aligned} S &= \text{Treegen Boolgen } e, \\ \text{Boolgen } x &= x, \\ \text{Treegen } f\ x &= x \oplus_{\frac{1}{2}} f(\text{Treegen } f\ (\text{Treegen } f\ (\text{Treegen } f\ x))). \end{aligned}$$

The non-terminal Treegen uses f unboundedly, but the same parameter f occurs as first argument in all instances of Treegen, which can thus be given type $!_\infty \tau \multimap \tau$.

As for the PBHORS$^{<\infty}$, we interpret each PBHORS$^\infty$ $\mathcal{G} = (\mathcal{N}, \mathcal{P}, \mathcal{D}, S)$ via its induced simply typed PHORS $(\mathcal{N}_{\text{st}}, \mathcal{P}_{\text{st}}, \mathcal{D}_{\text{st}}, S)$. This yields then a family of fps $[\![\mathcal{G}]\!]^\mathcal{R} \in \mathcal{R}\{\!\{[\![\mathcal{N}_{\text{st}}]\!] + [\![\mathcal{P}_{\text{st}}]\!]\}\!\}^{[\![\mathcal{N}_{\text{st}}]\!]}$, that we will look at as parametric on $\mathcal{P}$, that is, as in $(\mathcal{R}\{\!\{[\![\mathcal{P}_{\text{st}}]\!]\}\!\})\{\!\{[\![\mathcal{N}_{\text{st}}]\!]\}\!\}^{[\![\mathcal{N}_{\text{st}}]\!]}$.

Two PHORS can be composed through their open parameters:

▶ **Definition 48** (composition of PBHORS$^{<\infty}$). *Given two PBHORS$^{\infty}$* $\mathcal{G}_1 = (\mathcal{N}_1, \mathcal{P} \cup \{w :_\infty \mathcal{N}(L)), \mathcal{D}_1, S_1)$ *and* $\mathcal{G}_2 = (\mathcal{N}_2, \mathcal{P}, \mathcal{D}_2, S_2)$*, where* $\mathcal{N}_1 \cap \mathcal{N}_2 = \emptyset$ *and* $L \in \mathcal{N}_2$*. Their* composition through L *is the PBHORS$^{\infty}$* $\mathcal{G}_1 \circ_L \mathcal{G}_2 = (\mathcal{N}_1 \cup \mathcal{N}_2, \mathcal{P}, \mathcal{D}_1[L/w] \cup \mathcal{D}_2, S_1)$.

The composition of two PHORS is interpreted via the composition of the associated formal power series, as shown below:

▶ **Lemma 49.** $[\![\mathcal{G}_1 \circ_L \mathcal{G}_2]\!]^{\mathcal{R}}\left(w_{[\![\mathcal{P}]\!]}\right) = [\![\mathcal{G}_1]\!]^{\mathcal{R}}\left([\![\mathcal{G}_2]\!]^{\mathcal{R}}\left(w_{[\![\mathcal{P}]\!]}\right), w_{[\![\mathcal{P}]\!]}\right)$.

Composability enables a *modular* analysis of the generating functions of a PHORS: we may for instance first focus on certain subsets of non-terminals, replacing all others with parameters, and only later put the pieces together via composition.

7.2 From PBHORS$^{\infty}$ to FAS

To show that the PBHORS$^{\infty}$ have algebraic generating functions, we will proceed in two steps. First, we will consider finitary PHORS with parameters, which can be shown algebraic with the same approach as in Section 6. Then, we show that any PBHORS$^{\infty}$ can be transformed into an equivalent parametric PBHORS$^{<\infty}$, which will result in the algebraicity of its generating function: the fundamental intuition behind this procedure is that the variables $\vec{w}$ of infinite grade can be in the end "discharged" onto the coefficient semiring, which is turned from $\mathbb{R}^{+\infty}_{\geq 0}\{\!\{z\}\!\}$ to $\mathbb{R}^{+\infty}_{\geq 0}\{\!\{z, \vec{w}\}\!\}$.

Finitary Parametric PHORS. Consider a $\mathcal{G} = (\mathcal{N}, \mathcal{P}, \mathcal{D}, S)$ that is parametric but *finitary*, i.e. all its non-terminal symbols have a finitary type. In other words, the derivations $\mathcal{D}(L_i)$ are of the form $\mathcal{N} \mid \mathcal{P} \mid \emptyset \vdash t_{L_i} : \varphi$, where $\varphi \sqsubseteq \mathcal{N}(L_i)$ and all types in $\mathcal{N}$ and $\mathcal{P}$ are finitary, but recall that the parameters w in $\mathcal{P}$ have infinite grade.

By arguing as in the previous section, but reasoning over the semiring $\mathcal{R}\{\!\{[\![P]\!]\}\!\}$, we can conclude that the pair of finite variable sets $([\![\mathcal{N}]\!], [\![\mathcal{N}]\!])$ is stable and polynomial, yielding:

▶ **Proposition 50.** *For all finitary parametric PBHORS$^{<\infty}$* $\mathcal{G} = (\mathcal{N}, \mathcal{P}, \mathcal{D}, S)$*, the fps* $a_L(z, w_{[\![\mathcal{P}]\!]}) \in \mathbb{R}^{+\infty}_{\geq 0}\{\!\{z, w_{[\![\mathcal{P}]\!]}\}\!\}$ *are all algebraic.*

▶ **Example 51.** Consider the PBHORS$^{<\infty}$ with parameters $f, g :_\infty \tau$ obtained from the one in Example 45 by adding two new non-terminals $L_{[f,g]} : \tau$ and $S_{[f,g]}$ with equations

$$L_{[f,g]}x = f(L_{[f,g]}(L_{[f,g]}x)) \oplus_a gx$$
$$S_{[f,g]} = L_{[f,g]}e$$

The interpretation of $L_{[f,g]} : \tau$ is the fps $a_F(z, f, g)(x) \in (\mathbb{R}^{+\infty}_{\geq 0}\{\!\{z, f, g\}\!\})\{\!\{x\}\!\}$ given by (8).

Reducing PBHORS$^{\infty}$ to Parametric PBHORS$^{<\infty}$. Following the idea of Example 51, we show how any closed PBHORS$^{\infty}$ can be turned into a parametric PBHORS$^{<\infty}$. The idea is to replace each non-terminal L with possibly infinitely graded inputs $w_1, \ldots, w_n$ with as many non-terminals $L_{[A_1,\ldots,A_n]}$ as all possible ways of *filling in* such inputs with other non-terminals from PBHORS$^{\infty}$.

Let then $\mathcal{G} = (\mathcal{N}, \emptyset, \mathcal{D}, S)$ be a closed PBHORS$^{\infty}$. For any non-terminal L whose type can be written as $\mathcal{N}(L) = !_\infty\phi_1 \multimap \cdots \multimap !_\infty\phi_k \multimap \phi_L$, let $\mathcal{P}_L$ be formed by all $w_i :_\infty \phi_i$, and let $T(L) := \{[t_1, \ldots t_n] \in (\mathcal{N} \cup \mathcal{P})^* \mid \forall i\, (\mathcal{N} \cup \mathcal{P})(t_i) = \phi_i)\}$ be the sets of *sostituable terms*. Let $\tilde{\mathcal{N}}$ be a new set of finitely typed non-terminals containing, for each $L \in \mathcal{N}$ as above and $\gamma \in T(L)$, a non-terminal $\tilde{L}_\gamma : \phi_L$. Finally, let $\mathcal{P}$ be the concatenation of all $\mathcal{P}_L$.

For each $L \in \mathcal{N}$, $\gamma \in T(L)$, and term $\mathcal{N} \mid \vec{v} : \mathcal{P}_L \mid \vec{x} : \Delta \vdash t_L : o$, define:

- an applicative term $\mathcal{N} \mid \mathcal{P} \mid \emptyset \vdash \alpha_{L,\gamma} : \phi_L$ defined by $\alpha_{L,\gamma} := L\gamma$;
- a finitary term $\widetilde{\mathcal{N}} \mid \vec{w} : \mathcal{P} \mid \vec{x} : \Delta \vdash \tilde{t}_{L,\gamma} : o$ defined by $\tilde{t}_{L,\gamma} := t_L[\gamma/\vec{v}]\left[\tilde{G}_\delta/G\delta\right]_{G\in\mathcal{N},\delta\in T(G)}$.

Notice that $\lambda\vec{x}.\tilde{t}_{L,\gamma} : \phi_L$.

We have obtained in this way a finitary parametric PHORS $\mathcal{G}_{\text{fin}} = (\widetilde{\mathcal{N}}, \mathcal{P}, \widetilde{\mathcal{D}}, S_{[]})$, whose interpretation is a FAS with coefficients in $\mathbb{R}^{+\infty}_{\geq 0}\{\!\{z, \vec{w}\}\!\}$. By inspecting the connected component formed by all non-terminals $\tilde{L}_\gamma$ "reachable" from the source $S_{[]}$, one can check that none of these contains free parameters in γ. This implies in particular that the generating function $a_{\mathcal{G}_{\text{fin}}}(z, \vec{w}) = a_{S_{[]}}(z, \vec{w})$ does not actually depend on the parameters $\vec{w}$. We will show that $a_{\mathcal{G}_{\text{fin}}}(z, \vec{w})$ coincides with $a_{\mathcal{G}}(z)$. Observe that $\mathcal{G}$ and $\mathcal{G}_{\text{fin}}$ are related by the following:

$$\mathcal{N} \mid \mathcal{P} \mid \emptyset \vdash \ \alpha_{L,\gamma}\left[\lambda\vec{v}.\lambda\vec{x}.t_L/L\right] \twoheadrightarrow \lambda\vec{x}.t_L[\gamma/\vec{v}] = \lambda\vec{x}.\tilde{t}_{L,\gamma}\left[\alpha_{G,\delta}/\tilde{G}_\delta\right]_{G\in\mathcal{N},\delta\in T(G)} \ : \ \widetilde{\mathcal{N}}(L) \quad (9)$$

The lemma below relates the minimal solutions of the systems of $\mathcal{G}$ and $\mathcal{G}_{\text{fin}}$:

▶ **Lemma 52.** *Given sets $\mathcal{N}, \widetilde{\mathcal{N}}, \mathcal{P}$ and morphisms $t \in \mathcal{R}\mathbf{Rel}_!(\mathcal{N}, \mathcal{N})$, $\tilde{t} \in \mathcal{R}\mathbf{Rel}_!(\widetilde{\mathcal{N}} + \mathcal{P}, \widetilde{\mathcal{N}})$, $\alpha \in \mathcal{R}\mathbf{Rel}_!(\mathcal{N} + \mathcal{P}, \widetilde{\mathcal{N}})$, if we have:*

1. *$\alpha \circ \langle 0, 1_{\mathcal{P}}\rangle = 0 \in \mathcal{R}\mathbf{Rel}_!(\mathcal{P}, \widetilde{\mathcal{N}})$,*
2. *$\alpha \circ \langle t \circ \pi_1, \pi_2\rangle = \tilde{t} \circ \langle \alpha \circ \langle \pi_1, \pi_2\rangle, \pi_2\rangle \in \mathcal{R}\mathbf{Rel}_!(\mathcal{N} + \mathcal{P}, \widetilde{\mathcal{N}})$,*

then $\alpha \circ \langle \mathsf{fix}_{0,\mathcal{N}} t, 1_{\mathcal{P}}\rangle = \mathsf{fix}_{\mathcal{P},\widetilde{\mathcal{N}}} \tilde{t} \in \mathcal{R}\mathbf{Rel}_!(\mathcal{P}, \widetilde{\mathcal{N}}) \equiv \mathcal{R}\{\!\{\mathcal{P}\}\!\}\mathbf{Rel}_!(1, \widetilde{\mathcal{N}})$.

▶ **Proposition 53.** *For any non-terminal $L \in \mathcal{N}$, $a_L(z, \vec{w}_L, \vec{x}) = a_{\tilde{L}_{\vec{w}_L}}(z, \vec{w}, \vec{x})$, where the $\vec{w}_L$ are the variables ws belonging to $[\![\mathcal{P}_L]\!]$, and $a_L(z, \vec{w}_L, \vec{x})$ is thus algebraic.*

Proof sketch. Condition 1. of Lemma 52 clearly holds for the interpretation of $\alpha = \langle [\![\alpha_{L,\gamma}]\!]^{\mathcal{R}}\rangle_{L,\gamma}$, and condition 2. is precisely (9), hence $\mathsf{fix}\ [\![t_{\mathcal{G}^{\text{fin}}}]\!] = \alpha \circ \langle \mathsf{fix}\ [\![t_{\mathcal{G}}]\!], 1_{\mathcal{P}}\rangle$. Since $\pi_{\tilde{L},\vec{w}_L} \circ \alpha = \alpha_{L,\vec{w}_L}$ and $\alpha_{L,\vec{w}_L}(\langle L_1, \ldots, L_n\rangle, \vec{w}) = L(\vec{w}_L)$, we have $a_{\mathcal{G}_{\text{fin}}}(z, \vec{w}, \vec{x}) = \pi_{L,\vec{w}_L} \circ \mathsf{fix}\ [\![t_{\mathcal{G}_{\text{fin}}}]\!] = \pi_{L,\vec{w}_L} \circ \alpha \circ \langle \mathsf{fix}\ [\![t_{\mathcal{G}}]\!], 1_{\mathcal{P}}\rangle = \alpha_{L,\vec{w}_L} \circ \langle \mathsf{fix}\ [\![t_{\mathcal{G}}]\!], 1_{\mathcal{P}}\rangle = a_L(z, \vec{w}_L, x)$. ◀

We thus immediately obtain:

▶ **Corollary 54.** *For any closed PBHORS$^\infty$ $\mathcal{G}$ there exists a parametric PBHORS$^{<\infty}$ $\mathcal{G}_{\text{fin}}$ of size $\|\mathcal{G}_{\text{fin}}\| \leq \|\mathcal{G}\|^{|\mathcal{N}|}$ such that $a_{\mathcal{G}}(z) = a_{\mathcal{G}_{\text{fin}}}(z, 1)$ and $\mathcal{L}(\mathcal{G}) = \mathcal{L}(\mathcal{G}_{\text{fin}})$. In particular, the AST and PAST problems for $\mathcal{G}$ are decidable.*

▶ **Example 55.** If we apply this technique to the PBHORS$^\infty$ from Example 45, we must introduce all order-1 non-terminals $L_{[\gamma_1,\gamma_2]}$ with equations $L_{[\gamma_1,\gamma_2]}x = \gamma_1(L_{[\gamma_1,\gamma_2]}(L_{[\gamma_1,\gamma_2]}x) \oplus_a \gamma_2 x$, where γ_1, γ_2 are either non-terminals or parameters. With $\gamma_1, \gamma_2 = f, g$, we precisely get the equation in Example 51. By restricting ourselves to the connected component of $S_{[]}$ we obtain then a *closed* order-1 PBHORS$^{<\infty}$ with non-terminals $S_{[]}, A, B, L_{[A,B]}$.

8 Related Work

The connections between the semantics of linear logic and combinatorics/formal power series can be traced back to Girard's quantitative semantics [25] (and Lamarche's infinitary algebras [41]) and their connections with Joyal's theory of analytic functors and combinatorial species [31]. As in [7], we here take at face value the correspondence between morphisms in $\mathcal{R}\mathbf{Rel}_!$ and formal power series. Similar ideas were already exploited in [18] to establish the full abstraction of the related semantics of probabilistic coherent spaces.

An extensive literature in analytic combinatorics has explored different kinds of combinatorial structures having an algebraic generating function, in particular with respect to their asymptotic behaviour (for a comprehensive treatment, see [22]). In the setting of imperative programming, the idea of using generating functions to analyse probabilistic programs is developed in [34] and [35]; of particular interest, with respect to our work is the characterization of while loops giving rise to rational generating functions in [34]. On the side of formal language theory, there exists a huge amount of work about generating functions; particularly relevant to this work is the paper [1], which studies the generating functions of Aho's indexed languages (corresponding to order-2 safe HORS, thus going beyond the usual framework of context-freeness) and they give an explicit procedure to compute these functions under some restrictions, but do not give a characterization of grammars with an algebraic generating function.

The connection between HORS and the relational semantics of linear logic first appears in Melliès and Grellois' work [28], [27]. Subsequently, in [12] a linear-non linear typing system for HORS was introduced, allowing for a finer complexity analysis of model checking; its affine fragment is studied in [13]. None of these works involve probabilistic behavior.

In [36], the problem of AST for PHORS is studied by directly translating a PHORS into (possibly higher-order) real functional equations: only in the case of order 1 PHORS, this produces a fixpoint algebraic system analogous the those considered in this work, thus implying a decidability result (while in the same work, undecidability since order-2 is proved). Techniques to approximate from above the probability of termination are also considered. In [42], the decidability of AST/PAST for affine PHORS is proved. With respect to our approach, the translation from PHORS to real equations is defined via an intermediate automata-theoretic step, based on the equivalence between affine HORS and restricted pushdown automata proved in [13].

9 Conclusion

In this work we have shown that the combinatorial method of generating functions can be adapted, via the weighted relational semantics of linear logic, to the study of higher-order *probabilistic* languages. We think that the main value of this work resides in opening the possibility of applying well-established methods from algebraic and analytic combinatorics to higher-order languages. We provided a first demonstration of this by showing how the decidability of termination for certain classes of PHORS (proved in the literature with syntactic methods like automata theory or game semantics) can be re-established, and actually *extended*, in a relatively elementary way, via the notion of algebraic power series.

At the same time, several other directions of application of combinatorial methods can be mentioned: on the one hand, could one capture a class of PHORS, extending the algebraic ones, giving rise to *D-finite* power series [33], i.e. fps defined by linear *differential* equations, and could this be related to the recently explored connections between higher-order differentiation and fixpoints in the weighted relational model [23]? On the other hand, could the vast body of work on *asymptotic estimations* [22] of generating functions be applied to extract approximated information about the probabilistic behavior of even larger classes of PHORS? And, more broadly, could all these methods help shed light on the open problem of exactly placing the AST problem for PHORS within the arithmetic hierarchy?

References

1 Jared Adams, Eric Freden, and Marni Mishna. From indexed grammars to generating functions. *RAIRO Theor. Informatics Appl.*, 47(4):325–350, 2013. doi:10.1051/ITA/2013041.

2 Alfred V. Aho. Indexed grammars-an extension of context free grammars. In *8th Annual Symposium on Switching and Automata Theory, Austin, Texas, USA, October 18-20, 1967*, pages 21–31. IEEE Computer Society, 1967. doi:10.1109/FOCS.1967.16.

3 Krzysztof R. Apt and Dexter C. Kozen. Limits for automatic verification of finite-state concurrent systems. *Information Processing Letters*, 22(6):307–309, 1986. doi:10.1016/0020-0190(86)90071-2.

4 Arthur Azevedo de Amorim, Marco Gaboardi, Justin Hsu, Shin-ya Katsumata, and Ikram Cherigui. A semantic account of metric preservation. In *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages*, POPL '17, pages 545–556, New York, NY, USA, 2017. Association for Computing Machinery. doi:10.1145/3009837.3009890.

5 Christel Baier and Joost-Pieter Katoen. *Principles of Model-Checking*. The MIT Press, 2008.

6 Cyril Banderier and Michael Drmota. Formulae and asymptotics for coefficients of algebraic functions. *Combinatorics, Probability and Computing*, 24(1):1–53, 2015. doi:10.1017/S0963548314000728.

7 Davide Barbarossa and Paolo Pistone. Tropical mathematics and the lambda-calculus II: Tropical geometry of probabilistic programming languages. *Proc. ACM Program. Lang.*, 10(POPL), 2026. doi:10.1145/3776675.

8 Aloïs Brunel, Marco Gaboardi, Damiano Mazza, and Steve Zdancewic. A core quantitative coeffect calculus. In *Proceedings of the 23rd European Symposium on Programming Languages and Systems – Volume 8410*, pages 351–370, Berlin, Heidelberg, 2014. Springer-Verlag. doi:10.1007/978-3-642-54833-8_19.

9 Arnaud Carayol and Olivier Serre. Higher-order recursion schemes and their automata models. In Jean-Éric Pin, editor, *Handbook of Automata Theory*, pages 1295–1341. European Mathematical Society Publishing House, Zürich, Switzerland, 2021. doi:10.4171/AUTOMATA-2/13.

10 N. Chomsky and M.P. Schützenberger. The algebraic theory of context-free languages*. In P. Braffort and D. Hirschberg, editors, *Computer Programming and Formal Systems*, volume 35 of *Studies in Logic and the Foundations of Mathematics*, pages 118–161. Elsevier, 1963. doi:10.1016/S0049-237X(08)72023-8.

11 D.V Chudnovsky and G.V Chudnovsky. On expansion of algebraic functions in power and puiseux series, i. *Journal of Complexity*, 2(4):271–294, 1986. doi:10.1016/0885-064X(86)90006-3.

12 Pierre Clairambault, Charles Grellois, and Andrzej S. Murawski. Linearity in higher-order recursion schemes. *Proc. ACM Program. Lang.*, 2(POPL):39:1–39:29, 2018. doi:10.1145/3158127.

13 Pierre Clairambault and Andrzej S. Murawski. On the expressivity of linear recursion schemes. In Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen, editors, *44th International Symposium on Mathematical Foundations of Computer Science, MFCS 2019, Aachen, Germany, August 26-30, 2019*, volume 138 of *LIPIcs*, pages 50:1–50:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.MFCS.2019.50.

14 Edmund M. Clarke, E. Allen Emerson, and Joseph Sifakis. Model checking: algorithmic verification and debugging. *Commun. ACM*, 52(11):74–84, 2009. doi:10.1145/1592761.1592781.

15 Ugo Dal Lago, Guido Fiorillo, and Paolo Pistone. On higher-order probabilistic verification via the weighted relational model of linear logic, 2026. arXiv:2604.27986.

16 Manfred Droste and Werner Kuich. *Semirings and Formal Power Series*, pages 3–28. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. doi:10.1007/978-3-642-01492-5_1.

17 Thomas Ehrhard. Differentials and distances in probabilistic coherence spaces. In Herman Geuvers, editor, *4th International Conference on Formal Structures for Computation and Deduction, FSCD 2019, Dortmund, Germany, June 24-30, 2019*, volume 131 of *LIPIcs*, pages 17:1–17:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.FSCD.2019.17.

18 Thomas Ehrhard, Michele Pagani, and Christine Tasson. Full abstraction for probabilistic PCF. *J. ACM*, 65(4):23:1–23:44, 2018. doi:10.1145/3164540.

19 Javier Esparza, Antonin Kucera, and Richard Mayr. Model checking probabilistic pushdown automata. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science*, LICS '04, pages 12–21, USA, 2004. IEEE Computer Society. doi:10.1109/LICS.2004.1319596.

20 Kousha Etessami and Mihalis Yannakakis. Recursive markov chains, stochastic grammars, and monotone systems of nonlinear equations. *J. ACM*, 56(1):1:1–1:66, 2009. doi:10.1145/1462153.1462154.

21 Kousha Etessami and Mihalis Yannakakis. Model checking of recursive probabilistic systems. *ACM Trans. Comput. Logic*, 13(2), 2012. doi:10.1145/2159531.2159534.

22 Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009. URL: http://www.cambridge.org/uk/catalogue/catalogue.asp?isbn=9780521898065.

23 Zeinab Galal and Jean-Simon Pacaud Lemay. Combining fixpoint and differentiation theory. In Pawel Sobocinski, Ugo Dal Lago, and Javier Esparza, editors, *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2024, Tallinn, Estonia, July 8-11, 2024*, pages 37:1–37:14. ACM, 2024. doi:10.1145/3661814.3662108.

24 Dan R. Ghica and Alex I. Smith. Bounded linear types in a resource semiring. In *Proceedings of the 23rd European Symposium on Programming Languages and Systems – Volume 8410*, pages 331–350, Berlin, Heidelberg, 2014. Springer-Verlag. doi:10.1007/978-3-642-54833-8_18.

25 Jean-Yves Girard. Normal functors, power series and λ-calculus. *Annals of Pure and Applied Logic*, 37(2):129–177, 1988. doi:10.1016/0168-0072(88)90025-5.

26 Jean-Yves Girard, Andre Scedrov, and Philip J. Scott. Bounded linear logic: a modular approach to polynomial-time computability. *Theoretical Computer Science*, 97(1):1–66, 1992. doi:10.1016/0304-3975(92)90386-T.

27 Charles Grellois and Paul-André Melliès. Finitary semantics of linear logic and higher-order model-checking. In Giuseppe F. Italiano, Giovanni Pighizzini, and Donald Sannella, editors, *Mathematical Foundations of Computer Science 2015 – 40th International Symposium, MFCS 2015, Milan, Italy, August 24-28, 2015, Proceedings, Part I*, volume 9234 of *Lecture Notes in Computer Science*, pages 256–268. Springer, 2015. doi:10.1007/978-3-662-48057-1_20.

28 Charles Grellois and Paul-André Melliès. Relational semantics of linear logic and higher-order model checking. In Stephan Kreutzer, editor, *24th EACSL Annual Conference on Computer Science Logic, CSL 2015, Berlin, Germany, September 7-10, 2015*, volume 41 of *LIPIcs*, pages 260–276. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPIcs.CSL.2015.260.

29 Masahito Hasegawa. On traced monoidal closed categories. *Math. Struct. Comput. Sci.*, 19(2):217–244, 2009. doi:10.1017/S0960129508007184.

30 Gerard J. Holzmann. The model checker spin. *IEEE Trans. Softw. Eng.*, 23(5):279–295, 1997. doi:10.1109/32.588521.

31 André Joyal. Une théorie combinatoire des séries formelles. *Advances in Mathematics*, 42(1):1–82, 1981. doi:10.1016/0001-8708(81)90052-9.

32 Benjamin Lucien Kaminski, Joost-Pieter Katoen, and Christoph Matheja. On the hardness of analyzing probabilistic programs. *Acta Informatica*, 56(3):255–285, 2019. doi:10.1007/S00236-018-0321-1.

33 Manuel Kauers and Peter Paule. *The Concrete Tetrahedron – Symbolic Sums, Recurrence Equations, Generating Functions, Asymptotic Estimates*. Texts & Monographs in Symbolic Computation. Springer, 2011. doi:10.1007/978-3-7091-0445-3.

34 Lutz Klinkenberg, Kevin Batz, Benjamin Lucien Kaminski, Joost-Pieter Katoen, Joshua Moerman, and Tobias Winkler. Generating functions for probabilistic programs. In *Theoretical Computer Science and General Issues*, volume 12561, pages 231–248. Springer, 2021. doi:10.1007/978-3-030-68446-4_12.

35 Lutz Klinkenberg, Christian Blumenthal, Mingshuai Chen, Darion Haase, and Joost-Pieter Katoen. Exact bayesian inference for loopy probabilistic programs using generating functions. *Proceedings of the ACM on programming languages*, 8(OOPSLA1):pages 127, 2024. doi:10.1145/3649844.

36 Naoki Kobayashi, Ugo Dal Lago, and Charles Grellois. On the termination problem for probabilistic higher-order recursive programs. *Log. Methods Comput. Sci.*, 16(4), 2020. URL: https://lmcs.episciences.org/6817.

37 Werner Kuich and Arto Salomaa. *Semirings, Automata, Languages*, volume 5 of *EATCS Monographs on Theoretical Computer Science*. Springer, 1986. doi:10.1007/978-3-642-69959-7.

38 Marta Z. Kwiatkowska, Gethin Norman, and David Parker. Prism: Probabilistic symbolic model checker. In *Proceedings of the 12th International Conference on Computer Performance Evaluation, Modelling Techniques and Tools*, TOOLS '02, pages 200–204, Berlin, Heidelberg, 2002. Springer-Verlag. doi:10.1007/3-540-46029-2_13.

39 Ugo Lago and Martin Hofmann. Bounded linear logic, revisited. In *Proceedings of the 9th International Conference on Typed Lambda Calculi and Applications*, TLCA '09, pages 80–94, Berlin, Heidelberg, 2009. Springer-Verlag. doi:10.1007/978-3-642-02273-9_8.

40 Jim Laird, Giulio Manzonetto, Guy McCusker, and Michele Pagani. Weighted relational models of typed lambda-calculi. In *28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25-28, 2013*, pages 301–310. IEEE Computer Society, 2013. doi:10.1109/LICS.2013.36.

41 François Lamarche. Quantitative domains and infinitary algebras. *Theoretical Computer Science*, 94(1):37–62, 1992. doi:10.1016/0304-3975(92)90323-8.

42 Guanyan Li, Andrzej S. Murawski, and Luke Ong. Probabilistic verification beyond context-freeness. In Christel Baier and Dana Fisman, editors, *LICS '22: 37th Annual ACM/IEEE Symposium on Logic in Computer Science, Haifa, Israel, August 2 – 5, 2022*, pages 33:1–33:13. ACM, 2022. doi:10.1145/3531130.3533351.

43 David Marker. *Model Theory : An Introduction*. Graduate Texts in Mathematics. Springer, New York, NY, 2002 edition, August 2002.

44 C.-H. L. Ong. On model-checking trees generated by higher-order recursion schemes. In *Proceedings of the 21st Annual IEEE Symposium on Logic in Computer Science*, LICS '06, pages 81–90, USA, 2006. IEEE Computer Society. doi:10.1109/LICS.2006.38.

45 Luke Ong. Higher-order model checking: An overview. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, July 6-10, 2015*, pages 1–15. IEEE Computer Society, 2015. doi:10.1109/LICS.2015.9.

46 K. Vijay-Shanker, David J. Weir, and Aravind K. Joshi. Characterizing structural descriptions produced by various grammatical formalisms. In Candy L. Sidner, editor, *25th Annual Meeting of the Association for Computational Linguistics, Stanford University, Stanford, California, USA, July 6-9, 1987*, pages 104–111. ACL, 1987. doi:10.3115/981175.981190.