J. Log. Algebraic Methods Program. 146 (2025) 101070

Contents lists available at ScienceDirect

Journal of Logical and Algebraic Methods in Programming

journal homepage: www.elsevier.com/locate/jlamp

A distribution semantics for probabilistic term rewriting ☆

Germán Vidal

VRAIN, Universitat Politècnica de València, Spain

ARTICLE INFO

Keywords:
Term rewriting
Probability
Semantics
Modeling

ABSTRACT

Probabilistic programming is becoming increasingly popular thanks to its ability to specify problems with a certain degree of uncertainty. In this work, we focus on term rewriting, a well-known computational formalism. In particular, we consider systems that combine traditional rewriting rules with probabilities. Then, we define a novel "distribution semantics" for such systems that can be used to model the probability of reducing a term to some value. We also show how to compute a set of "explanations" for a given reduction, which can be used to compute its probability in a more efficient way. Finally, we illustrate our approach with several examples and outline a couple of extensions that may prove useful to improve the expressive power of probabilistic rewrite systems.

1. Introduction

Term rewriting [1] is a widely recognized computational formalism with many applications, ranging from providing theoretical foundations for programming languages to problem modeling and developing analysis and verification techniques. The addition of probabilities to a programming language or formalism has proven useful for modeling domains with complex relationships and uncertainty. Surprisingly, however, we find only a few proposals for *probabilistic* term rewriting. Among the various approaches to incorporating probabilities into term rewriting, the proposal by Bournez et al. [2–4] is perhaps the most well-known. It introduces the concept PARS (Probabilistic Abstract Reduction System) and its instance PTRS (Probabilistic Term Rewrite System), together with the corresponding notions of probabilistic (abstract) reduction. Intuitively speaking, in this line of work, a *probabilistic term rewrite system* (PTRS) is a collection of probabilistic rules of the form

$$l \to p_1 : r_1; \ldots; p_m : r_m \quad \text{with} \sum_{i \in \{1,\ldots,m\}} p_i = 1$$

where $l, r_1, \ldots, r_m$ are terms, $p_1, \ldots, p_m$ are real numbers in $[0,1]$, and the right-hand side, $p_1 : r_1; \ldots; p_m : r_m$, represents a probability distribution over terms.[1] This stream of work has mainly focused on the termination properties of PTRSs, studying notions like *almost-sure termination* (AST), i.e., determine whether the probability of termination is 1 [2], as well as several variants of this notion (like Positive AST [6,3] or Strong AST [5]). Besides the work of Bournez et al., confluence and termination of PTRSs has been considered, e.g., in [5,7–12].

☆ This work has been partially supported by grant PID2019-104735RB-C41 funded by MICIU/AEI/10.13039/501100011033, by the *Generalitat Valenciana* under grant CIPROM/2022/6 (FassLow), and by TAILOR, a project funded by EU Horizon 2020 research and innovation programme under GA No 952215.

E-mail address: gvidal@dsic.upv.es.

URL: https://gvidal.webs.upv.es/.

[1] Actually, this is called a *multi-distribution* in [5] since it is a multiset and repeated occurrences of the same term are allowed.

https://doi.org/10.1016/j.jlamp.2025.101070

Received 19 October 2024; Received in revised form 18 March 2025; Accepted 13 May 2025

Available online 23 May 2025

In this paper, though, we consider a different topic within probabilistic rewriting. In particular, we aim at defining a novel *distribution semantics* inspired to Sato's distribution semantics for probabilistic logic programs [13]. Roughly speaking, Sato defines a distribution over the *least models* of a probabilistic logic program, thus generalizing the traditional least model semantics. Here, we apply similar ideas in order to define a distribution over the standard TRSs that can be obtained from a PTRS by selecting a particular choice of each probabilistic rewrite rule. Consider, for instance, a simple PTRS that only includes the following probabilistic rules:

$$\begin{array}{l} R_1 : \ \mathsf{coin1} \to 0.5 : \mathsf{heads}; 0.5 : \mathsf{tails} \\ R_2 : \ \mathsf{coin2} \to 0.6 : \mathsf{heads}; 0.4 : \mathsf{tails} \end{array}$$

In this model, we have two coins: a fair coin, coin1, that lands on heads and on tails with equal probability, and a biased coin, coin2, that lands on heads with more probability than on tails. Here, we could say that this PTRS encodes the following four "worlds" (represented by standard TRSs):

$$\mathcal{W}_1 = \left\{ \begin{array}{l} \mathsf{coin1} \to \mathsf{heads} \\ \mathsf{coin2} \to \mathsf{heads} \end{array} \right\} \qquad \mathcal{W}_2 = \left\{ \begin{array}{l} \mathsf{coin1} \to \mathsf{heads} \\ \mathsf{coin2} \to \mathsf{tails} \end{array} \right\}$$

$$\mathcal{W}_3 = \left\{ \begin{array}{l} \mathsf{coin1} \to \mathsf{tails} \\ \mathsf{coin2} \to \mathsf{heads} \end{array} \right\} \qquad \mathcal{W}_4 = \left\{ \begin{array}{l} \mathsf{coin1} \to \mathsf{tails} \\ \mathsf{coin2} \to \mathsf{tails} \end{array} \right\}$$

by considering all possible combinations of choices for the probabilistic rules.

The probability of each TRS is then determined by the product of the probabilities of the choices made in each rule. Therefore, in this example, our distribution would be as follows: $P(\mathcal{W}_1) = 0.5 \cdot 0.6 = 0.3$, $P(\mathcal{W}_2) = 0.5 \cdot 0.4 = 0.2$, $P(\mathcal{W}_3) = 0.5 \cdot 0.6 = 0.3$, and $P(\mathcal{W}_4) = 0.5 \cdot 0.4 = 0.2$, with $P(\mathcal{W}_1) + P(\mathcal{W}_2) + P(\mathcal{W}_3) + P(\mathcal{W}_4) = 1$.

Then, we will focus on calculating *the probability that a term s is reduced to another term t* (usually a *value*, i.e., a term containing no reducible function, like heads or tails in the example above), i.e., a typical reachability problem, which we denote by $s \to^* t$, where $\to^*$ is the reflexive and transitive closure of the rewrite relation $\to$ associated to some (probabilistic) rewrite system. For instance, the probability that coin1 reduces to heads, denoted by $P(\mathsf{coin1} \to^* \mathsf{heads})$, is trivially 0.5 using the rule R_1 above.

In general, the probability that s reduces to t in a PTRS is obtained from the sum of the probabilities of the worlds where such a reduction can be proved. E.g., $P(\mathsf{coin1} \to^* \mathsf{heads}) = P(\mathcal{W}_1) + P(\mathcal{W}_2) = 0.3 + 0.2 = 0.5$.

A distinguishing feature of our proposal is that the application of a probabilistic rule is considered a *one-time event*, in the sense that the selected choice cannot be changed later in the same reduction. In particular, the tuple $\mathsf{t2}(\mathsf{coin1}, \mathsf{coin1})$, where t2 is a constructor symbol that represents a tuple of two elements, can only be reduced to either $\mathsf{t2}(\mathsf{heads}, \mathsf{heads})$ or $\mathsf{t2}(\mathsf{tails}, \mathsf{tails})$. For example, $P(\mathsf{t2}(\mathsf{coin1}, \mathsf{coin1}) \to^* \mathsf{t2}(\mathsf{heads}, \mathsf{heads})) = P(\mathcal{W}_1) + P(\mathcal{W}_2) = 0.3 + 0.2 = 0.5$, while $P(\mathsf{t2}(\mathsf{coin1}, \mathsf{coin1}) \to^* \mathsf{t2}(\mathsf{heads}, \mathsf{tails})) = 0$ since there is no world where coin1 can be reduced both to heads and to tails.

Our approach contrasts to previous work (based on that of Bournez et al. [2–4]) where a different choice can be taken every time a probabilistic rule is applied in a reduction. Moreover, in their setting, the probability of a reduction is obtained from the product of the probabilities of the steps, no matter if the same probabilistic rule is applied several times. Thus, the probability of the reduction $\mathsf{t2}(\mathsf{coin1}, \mathsf{coin1}) \stackrel{0.5}{\to} \mathsf{t2}(\mathsf{heads}, \mathsf{coin1}) \stackrel{0.5}{\to} \mathsf{t2}(\mathsf{heads}, \mathsf{heads})$ would be $0.5 \cdot 0.5 = 0.25$ in the approach of Bournez et al., accounting for the two applications of rule R_1. On the other hand, $\mathsf{t2}(\mathsf{coin1}, \mathsf{coin1})$ could also be reduced to $\mathsf{t2}(\mathsf{heads}, \mathsf{tails})$ with the same probability.

We do not mean to say that one semantics is *better* than the other. For some problems, one semantics could be more appropriate, and vice versa. For example, as we will show in Example 23, our proposal allows one to represent a Bayesian network in a natural way. The main contributions of this work can be summarized as follows:

- First, we introduce a definition of a PTRS that allows us to define in a simple way a distribution semantics for probabilistic systems.
- Next, we focus on computing the probability that a term s reduces to a term t, in symbols $P(s \to^* t)$, and present a declarative definition for $P(s \to^* t)$. Unfortunately, this definition is not always feasible in practice and, thus, we also propose an extension of term rewriting that allows us to generate a (complete) set of "explanations" associated to a reduction $s \to^* t$. From these explanations, the probability $P(s \to^* t)$ can be computed using well-known techniques.
- Finally, we illustrate the usefulness our proposal through several examples and introduce a couple of extensions for PTRSs that can help to increase their expressive power.

The distribution semantics has been quite successful in the context of logic programming, where several languages have emerged that follow it, e.g., ProbLog [14], Logic Programs with Annotated Disjunctions [15], Independent Choice Logic [16], PRISM [17], etc. Therefore, we think that it is worth considering a similar approach in the field of term rewriting too.

2. Standard term rewriting

To be self-contained, in this section we recall some basic concepts from term rewriting. We refer the reader to, e.g., [18] and [1], for further details.

2.1. Terms and substitutions

A *signature* $\mathcal{F}$ is a finite set of ranked function symbols. Given a set of variables $\mathcal{V}$ with $\mathcal{F} \cap \mathcal{V} = \emptyset$, we denote the domain of *terms* by $\mathcal{T}(\mathcal{F}, \mathcal{V})$. We use $\mathsf{f}, \mathsf{g}, \ldots$ to denote functions and $x, y, \ldots$ to denote variables. Positions are used to address the nodes of a term viewed as a tree. A *position* π in a term t, in symbols $\pi \in \mathcal{P}os(t)$, is represented by a finite sequence of natural numbers, where ϵ denotes the root position. We let $t|_\pi$ denote the *subterm* of t at position π and $t[s]_\pi$ the result of *replacing the subterm* $t|_\pi$ by the term s. The set of variables appearing in term t is denoted by $\mathcal{V}\mathsf{ar}(t)$. We also let $\mathcal{V}\mathsf{ar}(t_1, \ldots, t_n)$ denote $\mathcal{V}\mathsf{ar}(t_1) \cup \cdots \cup \mathcal{V}\mathsf{ar}(t_n)$. A term t is *ground* if $\mathcal{V}\mathsf{ar}(t) = \emptyset$.

A *substitution* $\sigma : \mathcal{V} \mapsto \mathcal{T}(\mathcal{F}, \mathcal{V})$ is a mapping from variables to terms such that $\mathcal{D}om(\sigma) = \{x \in \mathcal{V} \mid x \neq \sigma(x)\}$ is its domain. A substitution σ is *ground* if $x\sigma$ is ground for all $x \in \mathcal{D}om(\sigma)$. Substitutions are extended to morphisms from $\mathcal{T}(\mathcal{F}, \mathcal{V})$ to $\mathcal{T}(\mathcal{F}, \mathcal{V})$ in the natural way. We denote the application of a substitution σ to a term t by $t\sigma$ rather than $\sigma(t)$. The identity substitution is denoted by id. We let "$\circ$" denote the composition of substitutions, i.e., $(\sigma \circ \theta)(x) = (x\theta)\sigma = x\theta\sigma$. The *restriction* $\theta\restriction_V$ of a substitution θ to a set of variables V is defined as follows: $x\theta\restriction_V = x\theta$ if $x \in V$ and $x\theta\restriction_V = x$ otherwise.

2.2. Term rewriting systems

A finite set of rewrite rules $l \to r$ such that l is a nonvariable term and r is a term whose variables appear in l is called a *term rewriting system* (TRS for short); terms l and r are called the left-hand side and the right-hand side of the rule, respectively. Given a TRS $\mathcal{R}$ over a signature $\mathcal{F}$, the *defined* symbols $\mathcal{D}_\mathcal{R}$ are the root symbols of the left-hand sides of the rules and the *constructors* are $\mathcal{C}_\mathcal{R} = \mathcal{F} \setminus \mathcal{D}_\mathcal{R}$. *Constructor terms* of $\mathcal{R}$ are terms over $\mathcal{C}_\mathcal{R}$ and $\mathcal{V}$, denoted by $\mathcal{T}(\mathcal{C}_\mathcal{R}, \mathcal{V})$. We sometimes omit $\mathcal{R}$ from $\mathcal{D}_\mathcal{R}$ and $\mathcal{C}_\mathcal{R}$ if it is clear from the context. A substitution σ is a *constructor substitution* (of $\mathcal{R}$) if $x\sigma \in \mathcal{T}(\mathcal{C}_\mathcal{R}, \mathcal{V})$ for all variables x.

For a TRS $\mathcal{R}$, we define the associated rewrite relation $\to_\mathcal{R}$ as the smallest binary relation on terms satisfying the following: given terms $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$, we have $s \to_\mathcal{R} t$ iff there exist a position π in s, a rewrite rule $l \to r \in \mathcal{R}$, and a substitution σ such that $s|_\pi = l\sigma$ and $t = s[r\sigma]_\pi$; the rewrite step is sometimes denoted by $s \to_{\pi, l \to r} t$ to make explicit the position and rule used in this step. The instantiated left-hand side $l\sigma$ is called a *redex*. A term s is called *irreducible* or in *normal form* with respect to a TRS $\mathcal{R}$ if there is no term t with $s \to_\mathcal{R} t$; we often omit "with respect to $\mathcal{R}$" if it is clear from the context. A *derivation* is a (possibly empty) sequence of rewrite steps. Given a binary relation $\to$, we denote by $\to^*$ its reflexive and transitive closure, i.e., $s \to^*_\mathcal{R} t$ means that s can be reduced to t in $\mathcal{R}$ in zero or more steps; furthermore, we let $s \to^! t$ if $s \to^* t$, and, moreover, t is a normal form.

3. A distribution semantics for probabilistic TRSs

In this section, we first present an extension of rewrite systems to model uncertainty. Then, we define a *distribution semantics* for probabilistic systems. Finally, we consider the problem of computing the probability of $s \to^* t$ for some given terms s, t.

3.1. Probabilistic TRSs

In the following, we assume that the domain of defined function symbols $\mathcal{D}$ is partitioned into a set $\mathcal{D}^p$ of *probabilistic* functions and a set $\mathcal{D}^r$ of *regular* (non-probabilistic) functions, which are disjoint, i.e., $\mathcal{D} = \mathcal{D}^p \uplus \mathcal{D}^r$. A *probabilistic rule* has the form

$$R : l \to p_1 : r_1; \ldots; p_n : r_n$$

where R is a label that uniquely identifies the rule,[2] $l, r_1, \ldots, r_n$ are *ground* terms and $p_1, \ldots, p_n$ are real numbers in the interval $[0, 1]$ (their respective probabilities) such that $\sum_{i=1}^n p_i \leqslant 1$. When $\sum_{i=1}^n p_i < 1$, we implicitly assume that a special term $\perp$ is added to the right-hand side of the rule, where $\perp$ is a fresh constructor symbol of arity zero (a constant) which does not occur in the original system, with associated probability $1 - \sum_{i=1}^n p_i$. Here, choosing $\perp$ is equivalent to not reducing the selected term (nor any of its descendants). Thus, in the following, we assume w.l.o.g. that $\sum_{i=1}^n p_i = 1$ for all probabilistic rules. A *probabilistic term rewrite system*[3] (PTRS) $\mathcal{S} = \mathcal{S}_p \uplus \mathcal{S}_r$ with $\mathcal{D}_\mathcal{S} = \mathcal{D}^p_\mathcal{S} \uplus \mathcal{D}^r_\mathcal{S}$ consists of a set of probabilistic rules $\mathcal{S}_p$ defining the defined function symbols in $\mathcal{D}^p_\mathcal{S}$ and a set of regular rewrite rules $\mathcal{S}_r$ defining the defined function symbols in $\mathcal{D}^r_\mathcal{S}$.

Example 1. Consider the following PTRS with two probabilistic rules and two regular rules:

$$\begin{array}{llll} R_1: & \mathsf{coin1} \to 0.5 : \mathsf{heads}; 0.5 : \mathsf{tails} & R_3: & \mathsf{main} \to \mathsf{flip2coins}(\mathsf{coin1}, \mathsf{coin2}) \\ R_2: & \mathsf{coin2} \to 0.6 : \mathsf{heads}; 0.4 : \mathsf{tails} & R_4: & \mathsf{flip2coins}(x, y) \to \mathsf{t2}(x, y) \end{array}$$

In this example, function main represents the flipping of two coins. Functions coin1 and coin2 represent the two coins already mentioned in the Introduction. We have also a rewrite rule for the regular function flip2coins that simply reduces to a tuple t2(_, _) with the outcome of flipping the two coins.

[2] We will omit labels when they are not relevant.

[3] As long as there is no confusion, we will use the same terminology (PTRS) for probabilistic term rewrite systems as in the works based on the approach of [2–4] despite the differences.

Although the ground requirement for probabilistic rules may seem too restrictive, it is quite natural for many practical problems. This condition is essential for the number of possible "worlds" to be finite, as we will see below. Nevertheless, in the next section we will consider several ways to get around this restriction.

3.2. A distribution semantics for PTRSs

Let us now consider the definition of a distribution semantics for PTRSs. Essentially, each probabilistic rule $R : (l \to p_1 : r_1; \ldots ; p_n : r_n)$ represents a *choice* between n regular rules: $l \to r_1, \ldots, l \to r_n$. In other words, each probabilistic rule represents a *random variable* of the model represented by the PTRS. A particular choice is denoted by a tuple (R, i), $i \in \{1, \ldots, n\}$, which is called an *atomic choice*, and has as associated probability $\mathsf{p}(R, i)$, i.e., p_i in the rule above. Following [19], we introduce the following notions:

Definition 2 *(composite choice, selection).* We say that a set κ of atomic choices is *consistent* if it does not contain two different atomic choices for the same probabilistic rule, i.e., it cannot contain (R, i) and (R, j) with $i \neq j$.

A set of consistent atomic choices is called a *composite choice*.

Two composite choices, κ_1, κ_2, are *incompatible* if $\kappa_1 \cup \kappa_2$ is not consistent, i.e., we have $(R, i) \in \kappa_1$ and $(R, j) \in \kappa_2$ with $i \neq j$.

A set K of composite choices is *mutually incompatible* if for all $\kappa_1 \in K$, $\kappa_2 \in K$, $\kappa_1 \neq \kappa_2$ implies $\kappa_1 \cup \kappa_2$ is not consistent.

A composite choice is called a *selection* when it includes an atomic choice for each probabilistic rule of the PTRS. We let $\mathbb{S}_{\mathcal{S}}$ denote the set of all possible selections for a given PTRS $\mathcal{S}$.

Each selection κ identifies a (possible) *world* which contains a regular rule $l \to r_i$ for each atomic choice $(R, i) \in \kappa$, together with the rules for regular functions. Formally:

Definition 3 *(world).* Let $\mathcal{S}$ be a PTRS. A selection $\kappa \in \mathbb{S}_{\mathcal{S}}$ identifies a *world* $\mathcal{W}_\kappa$ defined as follows:

$$\mathcal{W}_\kappa = \{l \to r_i \mid R : (l \to p_1 : r_1; \ldots ; p_n : r_n) \in \mathcal{S}_p \wedge (R, i) \in \kappa\} \cup \mathcal{S}_r$$

The probability of world $\mathcal{W}_\kappa$ is then given by

$$P(\mathcal{W}_\kappa) = P(\kappa) = \prod_{(R,i)\in\kappa} \mathsf{p}(R, i)$$

We let $\mathbb{W}_{\mathcal{S}}$ denote the set of all possible worlds, i.e., $\mathbb{W}_{\mathcal{S}} = \{\mathcal{W}_\kappa \mid \kappa \in \mathbb{S}_{\mathcal{S}}\}$. Here, $P(\mathcal{W})$ defines a probability distribution over $\mathbb{W}_{\mathcal{S}}$. By definition, the sum of the probabilities of all possible worlds is equal to 1.

In this context, given a PTRS $\mathcal{S}$, we are interested in computing the probability of whether a term s can be reduced to term t in $\mathcal{S}$—a typical reachability problem—, in symbols $P(s \to^*_{\mathcal{S}} t)$ or simply $P(s \to^* t)$ when the PTRS $\mathcal{S}$ is clear from the context. This probability can be obtained by marginalization from the joint distribution $P(s \to^*_{\mathcal{S}} t, \mathcal{W})$ as follows:

Definition 4 *(probability of $s \to^* t$).* Let $\mathcal{S}$ be a PTRS and s, t be terms. Then, the probability of $s \to^*_{\mathcal{S}} t$ is given by

$$P(s \to^*_{\mathcal{S}} t) = \sum_{\mathcal{W}\in\mathbb{W}_{\mathcal{S}}} P(s \to^*_{\mathcal{S}} t, \mathcal{W}) = \sum_{\mathcal{W}\in\mathbb{W}_{\mathcal{S}}} P(s \to^*_{\mathcal{S}} t \mid \mathcal{W}) \cdot P(\mathcal{W})$$

where $P(s \to^*_{\mathcal{S}} t \mid \mathcal{W}) = 1$ if there exists a derivation $s \to^*_{\mathcal{W}} t$ and $P(s \to^*_{\mathcal{S}} t \mid \mathcal{W}) = 0$ otherwise.

Intuitively speaking, the probability of $s \to^* t$ is equal to the sum of the probabilities of all the worlds where the term s can be reduced to the term t. Often, we will be interested in knowing the probability $P(s \to^* t)$ where t is a normal form (a possible "value" of s).

Example 5. Consider again the PTRS of Example 1. Here, we have four possible worlds:

$$\begin{array}{l} \mathcal{W}_1 = \{\mathsf{coin1} \to \mathsf{heads}, \mathsf{coin2} \to \mathsf{heads}\} \\ \mathcal{W}_2 = \{\mathsf{coin1} \to \mathsf{heads}, \mathsf{coin2} \to \mathsf{tails}\} \\ \mathcal{W}_3 = \{\mathsf{coin1} \to \mathsf{tails}, \mathsf{coin2} \to \mathsf{heads}\} \\ \mathcal{W}_4 = \{\mathsf{coin1} \to \mathsf{tails}, \mathsf{coin2} \to \mathsf{tails}\} \end{array} \bigcup \left\{ \begin{array}{l} \mathsf{main} \to \mathsf{flip2coins}(\mathsf{coin1}, \mathsf{coin2}) \\ \mathsf{flip2coins}(x, y) \to \mathsf{t2}(x, y) \end{array} \right\}$$

with associated probabilities $P(\mathcal{W}_1) = 0.5 \cdot 0.6 = 0.3$, $P(\mathcal{W}_2) = 0.5 \cdot 0.4 = 0.2$, $P(\mathcal{W}_3) = 0.5 \cdot 0.6 = 0.3$, and $P(\mathcal{W}_4) = 0.5 \cdot 0.4 = 0.2$. Then, for instance, the probability of reducing main to $\mathsf{t2}(\mathsf{heads}, \mathsf{heads})$ is given by $P(\mathsf{main} \to^* \mathsf{t2}(\mathsf{heads}, \mathsf{heads})) = P(\mathcal{W}_1) = 0.3$. Let us now add the following (nondeterministic) regular rules to the considered PTRS:

$$\begin{array}{ll} R_5: & \mathsf{switch}(\mathsf{t2}(x, y)) \to \mathsf{t2}(x, y) \\ R_6: & \mathsf{switch}(\mathsf{t2}(x, y)) \to \mathsf{t2}(y, x) \end{array}$$

The number of worlds and their probabilities do not change since the probabilistic rules are the same; the only difference is that, now, rules R_5 and R_6 are added to every world. Consider now the probability of reducing $\mathsf{switch(main)}$ to $\mathsf{t2(heads,tails)}$, i.e., the probability of getting one heads and one tails, no matter which coin. Here, we have the following derivation D_1 in world $\mathcal{W}_2$:

$$\begin{array}{lll}\mathsf{switch(main)} & \to & \mathsf{switch(flip2coins(coin1,coin2))}\\ & \to & \mathsf{switch(t2(coin1,coin2))}\\ & \to & \mathsf{t2(coin1,coin2)}\\ & \to & \mathsf{t2(heads,coin2)}\\ & \to & \mathsf{t2(heads,tails)}\end{array}$$

and the following one, D_2, in world $\mathcal{W}_3$:

$$\begin{array}{lll}\mathsf{switch(main)} & \to & \mathsf{switch(flip2coins(coin1,coin2))}\\ & \to & \mathsf{switch(t2(coin1,coin2))}\\ & \to & \mathsf{t2(coin2,coin1)}\\ & \to & \mathsf{t2(heads,coin1)}\\ & \to & \mathsf{t2(heads,tails)}\end{array}$$

Thus, $P(\mathsf{switch(main)} \to^* \mathsf{t2(heads,tails)}) = P(\mathcal{W}_2) + P(\mathcal{W}_3) = 0.2 + 0.3 = 0.5$, i.e., the probability of getting heads and tails with the two coins does not depend on whether one of them is biased (it is always 50%).

We note that $P(s \to^* t)$ does not depend on the number of possible derivations from s to t in each world, but only on the number of worlds where the derivation is possible.

The usual properties over standard TRSs can be lifted to PTRS straightforwardly by considering the associated worlds. E.g., we say that a PTRS $\mathcal{S}$ is terminating (resp. confluent) if every world $\mathcal{W} \in \mathbb{W}_{\mathcal{S}}$ is terminating (resp. confluent). In general:

Definition 6. Let $\mathcal{S}$ be a PTRS. We say that a property P holds for PTRS $\mathcal{S}$ if P holds for each world $\mathcal{W} \in \mathbb{W}_{\mathcal{S}}$.

Unfortunately, the number of worlds grows exponentially with the number of random variables represented in the model, i.e., with the number of probabilistic rules in the PTRS. For example, n probabilistic rules with only two choices give rise to 2^n worlds. This makes computing $P(s \to^* t)$ by summing up the probabilities of the worlds where s can be reduced to t intractable.

3.3. Knowledge compilation for probabilistic inference

In the following, we present an alternative method based on so-called *knowledge compilation* [20]. Essentially, this approach is based on compiling a Boolean formula (representing a set of worlds) into a representation where inference is tractable. This approach has been used, e.g., for performing inference in ProbLog [14], where the process consists basically in the following two steps:

1. First, a propositional formula that represents the worlds where a given query is true is computed. For this purpose, a *grounding* stage is usually required. Then, some variant of SLD resolution is used to evaluate the query and produce the so-called *explanations* (see below), a representation of the worlds where the query is true.
2. Then, the propositional formula is compiled into a binary decision diagram (BDD) [21] or some other similar data structure where *weighted model counting* [22] can be done efficiently (often, in time linear in the size of the compiled representation). In this context, weights are associated to the probabilities of the original program.

Obviously, the complexity of the inference is the same (namely, the problem is PP-complete [23]) but now it is moved into the compilation stage. Moreover, it has been experimentally shown that the approach works well in many practical cases, scaling up to queries with a few hundred thousand explanations [24]. Alternatively, when exact inference is too expensive, we can still resort to *approximate* inference (some techniques for approximate inference are briefly described at the end of Section 3.4).

In the remainder of this section, we present a knowledge compilation approach to probabilistic inference in our setting. Similarly to [25,19] in the context of a formalism based on Horn clauses, we consider the notion of *explanation*. First, we say that a selection κ *extends* a composite choice κ' if $\kappa \supseteq \kappa'$; analogously, a world $\mathcal{W}$ is *compatible* with a composite choice κ' if there is a selection κ that extends κ' and $\mathcal{W} = \mathcal{W}_\kappa$. In general, a composite choice κ' identifies the set of worlds $\mathcal{W}(\kappa')$ that are compatible with κ', i.e., $\mathcal{W}(\kappa') = \{\mathcal{W}_\kappa \mid \kappa \in \mathbb{S}_{\mathcal{S}} \wedge \kappa \supseteq \kappa'\}$. Moreover, given a (finite) set of composite choices K, we let $\mathcal{W}(K) = \cup_{\kappa\in K}\ \mathcal{W}(\kappa)$.

Definition 7 *(explanation, covering)*. Let $\mathcal{S}$ be a PTRS and κ a composite choice for the probabilistic rules of $\mathcal{S}$. Given terms s,t, we say that κ is an *explanation* for $s \to_{\mathcal{S}}^* t$ iff there exists a derivation $s \to^* t$ in each world that is compatible with κ, i.e., there is a derivation $s \to_{\mathcal{W}}^* t$ for all $\mathcal{W} \in \mathcal{W}(\kappa)$.

Let K be a (finite) set of explanations for $s \to_{\mathcal{S}}^* t$. We say that K is *covering* for $s \to_{\mathcal{S}}^* t$ if for all $\mathcal{W} \in \mathbb{W}_{\mathcal{S}}$ such that $s \to_{\mathcal{W}}^* t$ we have $\mathcal{W} \in \mathcal{W}(K)$.

A set of covering explanations for $s \to^*_{\mathcal{S}} t$ can be seen as a compact representation of all the worlds where s can be reduced to t. In some cases, using this set of explanations—rather than all possible worlds—can make a significant difference regarding the time needed to compute the probability of $s \to^*_{\mathcal{S}} t$.

Now, in order to compute a covering set of explanations for a given reachability problem, we introduce an extension of rewriting over pairs $\langle t, \kappa \rangle$, where t is a term and κ is a composite choice.

Definition 8 *(probabilistic rewriting).* Let s, t be terms and κ, κ' be composite choices. A PTRS $\mathcal{S}$ induces a *probabilistic rewrite* relation $\rightharpoonup_{\mathcal{S}}$ where $\langle s, \kappa \rangle \rightharpoonup_{\mathcal{S}} \langle t, \kappa' \rangle$ if either

- there is a position π in s, a regular rewrite rule $l \to r \in \mathcal{S}$, and a substitution σ such that $s|_\pi = l\sigma$, $t = s[r\sigma]_\pi$, and $\kappa' = \kappa$, or
- there exists a position π in s, a probabilistic rewrite rule $R : (l \to p_1 : r_1; \ldots; p_n : r_n) \in \mathcal{S}$, and a substitution σ such that $s|_\pi = l\sigma$, $t = s[r_i\sigma]_\pi$, $i \in \{1, \ldots, n\}$, and $\kappa' = \kappa \cup \{(R, i)\}$ is consistent.

We often label a probabilistic rewrite step with the probability of the choice made: given $\langle s, \kappa \rangle \overset{p}{\rightharpoonup} \langle t, \kappa' \rangle$, we have $p = \mathsf{p}(R, i)$ if $\kappa' \setminus \kappa = \{(R, i)\}$, and $p = 1$ otherwise. Probabilities can then be lifted to derivations by computing the product of the probabilities of their steps, i.e., $\langle s_0, \kappa_0 \rangle \overset{p}{\rightharpoonup}{}^* \langle s_n, \kappa_n \rangle$ if $\langle s_0, \kappa_0 \rangle \overset{p_1}{\rightharpoonup} \langle s_1, \kappa_1 \rangle \overset{p_2}{\rightharpoonup} \ldots \overset{p_n}{\rightharpoonup} \langle s_n, \kappa_n \rangle$ and $p = p_1 \cdot p_2 \cdot \ldots \cdot p_n$.

It is interesting to note that we only label a step with probability $\mathsf{p}(R, i)$ when the rule R is probabilistic and, moreover, it is the *first time* it is applied in the derivation. Therefore, if we apply the same probabilistic rule several times in a derivation, we only count the associated probability once. Furthermore, the consistency requirement avoids applying the same rule with a different choice.

The next result proves that our definition of probabilistic rewriting is indeed sound and complete regarding the computation of explanations.

Theorem 9. *Let $\mathcal{S}$ be a PTRS and let s, t be terms. Then, for all selections $\kappa \in \mathbb{S}_{\mathcal{S}}$, we have $s \to^*_{\mathcal{W}_\kappa} t$ iff $\langle s, \emptyset \rangle \rightharpoonup^*_{\mathcal{S}} \langle t, \kappa' \rangle$ with $\kappa \supseteq \kappa'$.*

Proof. First, we consider the "only if" direction. Let $s \to^*_{\mathcal{W}_\kappa} t$. Since κ is consistent by definition, we can construct a derivation $\langle s, \emptyset \rangle \rightharpoonup^*_{\mathcal{S}} \langle t, \kappa' \rangle$ that mimicks the same steps of $s \to^*_{\mathcal{W}_\kappa} t$. Trivially, κ' is then a (consistent) composite choice with $\kappa \supseteq \kappa'$, as required.

Consider now the "if" direction. Let $\langle s, \emptyset \rangle \rightharpoonup^*_{\mathcal{S}} \langle t, \kappa' \rangle$ be a derivation with the probabilistic rewrite relation. By definition, κ' is a (consistent) composite choice and, thus, $s \to^*_{\mathcal{W}} t$ can be proved in every world that is compatible with κ', since the relevant choices (those for the probabilistic rules used in the derivation) are the same. $\square$

In the following, we let $\mathbb{E}(s \to^*_{\mathcal{S}} t)$ denote the set of explanations computed by probabilistic rewriting. Formally:

Definition 10. Let $\mathcal{S}$ be a PTRS and s, t be terms. Then, we define the explanations of $s \to^* t$ as follows:

$$\mathbb{E}(s \to^*_{\mathcal{S}} t) = \{\kappa \mid \langle s, \emptyset \rangle \rightharpoonup^*_{\mathcal{S}} \langle t, \kappa \rangle\}$$

Furthermore, the probability of an explanation is equal to the probability of the associated derivation, i.e., given $D = \langle s, \emptyset \rangle \rightharpoonup^*_{\mathcal{S}} \langle t, \kappa \rangle$, we have

$$P(\kappa) = P(D) = \prod_{(R,i) \in \kappa} \mathsf{p}(R, i).$$

As a consequence of Theorem 9, we have the following result:

Corollary 11. *Let $\mathcal{S}$ be a PTRS and let s, t be terms. Then, $\mathbb{E}(s \to^*_{\mathcal{S}} t)$ is a covering set of explanations for $s \to^*_{\mathcal{S}} t$.*

Example 12. Consider again the PTRS of Example 5 and the two derivations D_1 and D_2 from $\mathsf{switch(main)}$ to $\mathsf{t2(heads, tails)}$. The corresponding derivations with the probabilistic rewrite relation, D'_1 and D'_2, are shown in Fig. 1.

Computing a covering set of explanations is generally undecidable. However, there are classes of PTRSs for which this task becomes decidable, e.g., for terminating PTRSs (i.e., for PTRSs whose associated worlds are all terminating). On the other hand, we can easily reuse existing results from *reachability analysis* in term rewriting (see, e.g., [26,27], and references therein) in order to prune some derivations during the computation of a covering set of explanations. In particular, we are concerned with the *infeasibility* of a reachability problem: given a TRS $\mathcal{R}$ and (ground) terms s, t, determine if $s \to^*_{\mathcal{R}} t$ is not provable.

In principle, we could trivially extend the notion of reachability to PTRSs by considering all the associated worlds: a reachability problem $s \to^*_{\mathcal{S}} t$ is infeasible in a PTRS $\mathcal{S}$ if $s \to^*_{\mathcal{W}} t$ is infeasible in all worlds $\mathcal{W} \in \mathbb{W}_{\mathcal{S}}$. Moreover, we can take advantage of the *partial* explanation computed so far in order to discard some possible worlds. In the following, given a PTRS $\mathcal{S}$ and a composite choice κ, we let $\mathcal{S} \restriction \kappa$ denote the *restriction of $\mathcal{S}$ by κ*, which is defined as follows: $\mathcal{S} \restriction \kappa = \mathcal{S}^\kappa_p \cup \mathcal{S}_r$, where

$$\begin{array}{rcl} \langle \mathsf{switch}(\mathsf{main}), \emptyset\rangle & \rightharpoondown & \langle \mathsf{switch}(\mathsf{flip2coins}(\mathsf{coin1},\mathsf{coin2})), \emptyset\rangle \\ & \rightharpoondown & \langle \mathsf{switch}(\mathsf{t2}(\mathsf{coin1},\mathsf{coin2})), \emptyset\rangle \\ & \rightharpoondown & \langle \mathsf{t2}(\mathsf{coin1},\mathsf{coin2}), \emptyset\rangle \\ & \overset{0.5}{\rightharpoondown} & \langle \mathsf{t2}(\mathsf{heads},\mathsf{coin2}), \{(R_1,1)\}\rangle \\ & \overset{0.4}{\rightharpoondown} & \langle \mathsf{t2}(\mathsf{heads},\mathsf{tails}), \{(R_1,1),(R_2,2)\}\rangle \end{array}$$

$$\begin{array}{rcl} \langle \mathsf{switch}(\mathsf{main}), \emptyset\rangle & \rightharpoondown & \langle \mathsf{switch}(\mathsf{flip2coins}(\mathsf{coin1},\mathsf{coin2})), \emptyset\rangle \\ & \rightharpoondown & \langle \mathsf{switch}(\mathsf{t2}(\mathsf{coin1},\mathsf{coin2})), \emptyset\rangle \\ & \rightharpoondown & \langle \mathsf{t2}(\mathsf{coin2},\mathsf{coin1}), \emptyset\rangle \\ & \overset{0.6}{\rightharpoondown} & \langle \mathsf{t2}(\mathsf{heads},\mathsf{coin1}), \{(R_2,1)\}\rangle \\ & \overset{0.5}{\rightharpoondown} & \langle \mathsf{t2}(\mathsf{heads},\mathsf{tails}), \{(R_1,2),(R_2,1)\}\rangle \end{array}$$

Fig. 1. Probabilistic derivations for $\mathsf{switch}(\mathsf{main}) \to^* \mathsf{t2}(\mathsf{heads},\mathsf{tails})$.

$$\begin{array}{rl} \mathcal{S}_p^\kappa = & \{ l \to r_i \mid R : (l \to p_1 : r_1; \ldots ; p_n : r_n) \in \mathcal{S}_p \wedge (R,i) \in \kappa \} \\ & \cup \{ R : (l \to p_1 : r_1; \ldots ; p_n : r_n) \mid R : (l \to p_1 : r_1; \ldots ; p_n : r_n) \in \mathcal{S}_p \\ & \qquad \wedge \nexists j \text{ such that } (R,j) \in \kappa \} \end{array}$$

Intuitively speaking, $\mathcal{S} \restriction \kappa$ is equal to $\mathcal{S}$ except for the probabilistic rules whose choice already occurs in the partial explanation κ; these probabilistic rules are replaced by regular rules by removing all choices in their right-hand sides but the selected one (and also removing its probability).

Definition 13 *(infeasibility w.r.t. κ)*. Let $\mathcal{S}$ be a PTRS and s,t be terms. Given a composite choice κ, we say that the reachability problem $s \to_{\mathcal{S}}^* t$ is infeasible w.r.t. κ if $s \to_{\mathcal{S}\restriction\kappa}^* t$ is infeasible.

Therefore, during the computation of $\mathbb{E}(s \to_{\mathcal{S}}^* t)$, we can prune those derivations $\langle s,\emptyset\rangle \rightharpoondown_{\mathcal{S}}^* \langle s',\kappa\rangle$ in which $s' \to_{\mathcal{S}}^* t$ is infeasible w.r.t. κ.

Reachability (and, thus, the infeasibility of a reachability problem) is decidable for ground rewrite systems [28]. There are also a number of techniques for proving the infeasibility of some reachability problems when certain conditions are met. Many of these techniques are based on some abstraction of the rewrite rules, together with a graph that represents an approximation of the rewrite relation. One of the earliest such techniques can be found in [29], in which the TRS abstraction consists of replacing each rule $l \to r$ by the set of rules $l \to \widehat{r_1}, \ldots, l \to \widehat{r_n}$, where $r_1,\ldots,r_n$ are the subterms of r rooted by a defined function symbol and $\widehat{f(t_1,\ldots,t_n)} = f(\lfloor t_1\rfloor,\ldots,\lfloor t_n\rfloor)$ with $\lfloor t\rfloor = c(\lfloor t_1\rfloor,\ldots,\lfloor t_n\rfloor)$ if c is a constructor symbol and $\lfloor t\rfloor = x$ otherwise, with x a fresh variable. Then, a *graph of functional dependencies* is built by *connecting* l to $\widehat{r_i}$, $i = 1,\ldots,n$, for each rule and, then, adding edges from $\widehat{r_i}$ to each left-hand side with which it unifies. This graph of dependencies was used to prune infeasible equations in the context of basic narrowing [30] (an extension of term rewriting).

A similar technique was presented and popularized some years later in the context of the well-known *dependency pairs* method for proving the termination of rewrite systems [31,32]. In this case, the elements of the graph are *dependency pairs* of the form $l^\sharp \to t^\sharp$ for each rule $l \to r$, where t is a subterm of r rooted by a defined function symbol and $t^\sharp = f^\sharp(t_1,\ldots,t_n)$ if $t = f(t_1,\ldots,t_n)$ and $f^\sharp$ is a fresh function symbol. Then, dependency pairs $l^\sharp \to s^\sharp$ and $t^\sharp \to r^\sharp$ are *connected* if there is a substitution σ such that $s^\sharp\sigma \to^* t^\sharp\sigma$. This relation is often approximated, so that there is an edge from $l^\sharp \to s^\sharp$ to $t^\sharp \to r^\sharp$ in the graph if $\widehat{s^\sharp}$ unifies with $t^\sharp$, thus producing a similar result as the graph from [29] mentioned above.

The development of probabilistic rewrite strategies that prune derivations when the reachability problem is guaranteed to be infeasible is an interesting problem that is outside the scope of this paper and is left as future work.

3.4. Computing the probability from a covering set of explanations

Finally, in this section we focus on how to calculate the probability of a reachability problem from a set of covering explanations. The following result is an easy consequence:

Lemma 14. *Let $\mathcal{S}$ be a PTRS and κ be a composite choice. Then, $P(\kappa) = \sum_{\mathcal{W}\in\mathcal{W}(\kappa)} P(\mathcal{W})$.*

Proof. By definition, we have $P(\kappa) = \prod_{(R,i)\in\kappa} \mathsf{p}(R,i)$. Moreover, we have $\sum_{\mathcal{W}\in\mathcal{W}(\kappa)} P(\mathcal{W}) = \sum_{\kappa\cup\kappa'\in\mathbb{S}_{\mathcal{S}}} P(\mathcal{W}_{\kappa\cup\kappa'})$, by considering all selections that extend κ instead of the worlds in $\mathcal{W}(\kappa)$. Now,

$$\begin{array}{rcl} \sum_{\kappa\cup\kappa'\in\mathbb{S}_{\mathcal{S}}} P(\mathcal{W}_{\kappa\cup\kappa'}) & = & \sum_{\kappa\cup\kappa'\in\mathbb{S}_{\mathcal{S}}} \prod_{(R,i)\in\kappa\cup\kappa'} \mathsf{p}(R,i) \\ & = & \prod_{(R,i)\in\kappa} \mathsf{p}(R,i) \cdot (\sum_{\kappa\cup\kappa'\in\mathbb{S}_{\mathcal{S}}} \prod_{(R,i)\in\kappa'} \mathsf{p}(R,i)) \\ & = & P(\kappa) \cdot (\sum_{\kappa\cup\kappa'\in\mathbb{S}_{\mathcal{S}}} \prod_{(R,i)\in\kappa'} \mathsf{p}(R,i)) \end{array}$$

Finally, since $\sum_{\kappa\cup\kappa'\in\mathbb{S}_{\mathcal{S}}} \prod_{(R,i)\in\kappa'} \mathsf{p}(R,i)$ considers all possibilities for the probabilistic rules in κ', we have $(\sum_{\kappa\cup\kappa'\in\mathbb{S}_{\mathcal{S}}} \prod_{(R,i)\in\kappa'} \mathsf{p}(R,i)) = 1$, and the claim follows. $\square$

In the following, we let $P(K)$ denote the sum of the probabilities of all the worlds that are compatible with the explanations in K, i.e., $P(K) = \sum_{\mathcal{W}\in\mathcal{W}(K)} P(\mathcal{W})$. Interestingly, for sets of mutually incompatible explanations (cf. Definition 2), the following property holds:

Lemma 15. *Let $\mathcal{S}$ be a PTRS and K be a mutually incompatible set of explanations. Then, $P(K) = \sum_{\kappa\in K} P(\kappa)$.*

Proof. The result is an straightforward consequence of [19, Lemma 4.4] which states that, if K is mutually incompatible, then there is no world in which more than one element of K is true. Let $K = \{\kappa_1, \ldots, \kappa_n\}$. Then, we have $\mathcal{W}(K) = \mathcal{W}(\kappa_1) \cup \ldots \cup \mathcal{W}(\kappa_n)$ with $\mathcal{W}(\kappa_1) \cap \ldots \cap \mathcal{W}(\kappa_n) = \emptyset$. Hence, by Lemma 14, $P(K) = \sum_{\mathcal{W}\in\mathcal{W}(K)} P(\mathcal{W}) = \sum_{\mathcal{W}\in\mathcal{W}(\kappa_1)} P(\mathcal{W}) + \ldots + \sum_{\mathcal{W}\in\mathcal{W}(\kappa_n)} P(\mathcal{W}) = P(\kappa_1) + \ldots + P(\kappa_n)$, which proves the claim. □

Finally, when a set of explanations is both covering and mutually incompatible, the probability of a reachability problem can be computed in a straightforward way[4]:

Proposition 16. *Let $\mathcal{S}$ be a PTRS and s, t be terms. Let $K = \mathbb{E}(s \to^*_{\mathcal{S}} t)$. If K is mutually incompatible, then $P(s \to^*_{\mathcal{S}} t) = \sum_{\kappa\in K} P(\kappa)$.*

Proof. By definition, $P(s \to^*_{\mathcal{S}} t) = \sum_{\mathcal{W}\in\mathbb{W}_{\mathcal{S}}} P(s \to^*_{\mathcal{S}} t \mid \mathcal{W}) \cdot P(\mathcal{W})$. We know that K is a covering set of explanations by Corollary 11. Hence, we have $\sum_{\mathcal{W}\in\mathbb{W}_{\mathcal{S}}} P(s \to^*_{\mathcal{S}} t \mid \mathcal{W}) \cdot P(\mathcal{W}) = \sum_{\mathcal{W}\in\mathcal{W}(K)} P(\mathcal{W})$. Moreover, since K is mutually incompatible, by Lemma 15, we have $\sum_{\mathcal{W}\in\mathcal{W}(K)} P(\mathcal{W}) = \sum_{\kappa\in K} P(\kappa)$. □

For instance, let κ_1 and κ_2 be the explanations computed in Example 12, i.e., $\kappa_1 = \{(R_1, 1), (R_2, 2)\}$, $\kappa_2 = \{(R_1, 2), (R_2, 1)\}$. Then, we have $P(\kappa_1) = \mathsf{p}(R_1, 1) \cdot \mathsf{p}(R_2, 2) = 0.5 \cdot 0.4 = 0.2$ and $P(\kappa_2) = \mathsf{p}(R_1, 2) \cdot \mathsf{p}(R_2, 1) = 0.5 \cdot 0.6 = 0.3$. Since this set of explanations is covering and mutually incompatible, we have $P(\mathsf{switch}(\mathsf{main}) \to^* \mathsf{t2}(\mathsf{heads}, \mathsf{tails})) = P(\kappa_1) + P(\kappa_2) = 0.2 + 0.3 = 0.5$.

We note that, in Example 12, we have only considered two derivations, although there are actually quite a few more. In this case, they all compute one of the two explanations shown above (either κ_1 or κ_2). Setting a rewrite strategy that is complete for a certain class of TRSs (e.g., leftmost innermost rewriting [33]) may help to minimize the number of derivations to consider.

Unfortunately, when a covering set of explanations K for $s \to^* t$ is not mutually incompatible, the equality $P(s \to^* t) = \sum_{\kappa\in K} P(\kappa)$ does not generally hold, since the sets of worlds that are compatible with the explanations may overlap, i.e., we can have $\kappa_1, \kappa_2 \in K$ such that $\mathcal{W}(\kappa_1) \cap \mathcal{W}(\kappa_2) \neq \emptyset$ and, thus, $P(\kappa_1) + P(\kappa_2) \neq \sum_{\mathcal{W}\in\mathcal{W}(\kappa_1)\cup\mathcal{W}(\kappa_2)} P(\mathcal{W})$.

Example 17. Consider again the PTRS of Example 5, where we now add the following two regular rewrite rules:

$$
\begin{array}{ll}
R_7: & \mathsf{switch}(\mathsf{t2}(x, \mathsf{heads})) \to \mathsf{t2}(\mathsf{heads}, \mathsf{tails}) \\
R_8: & \mathsf{switch}(\mathsf{t2}(\mathsf{heads}, y)) \to \mathsf{t2}(\mathsf{heads}, \mathsf{tails})
\end{array}
$$

They represent a sort of shortcut when one coin lands on heads. Now, for instance, we have the following derivation:

$$
\begin{array}{rl}
\langle \mathsf{switch}(\mathsf{main}), \emptyset \rangle & \rightharpoondown \langle \mathsf{switch}(\mathsf{flip2coins}(\mathsf{coin1}, \mathsf{coin2})), \emptyset \rangle \\
& \rightharpoondown \langle \mathsf{switch}(\mathsf{t2}(\mathsf{coin1}, \mathsf{coin2})), \emptyset \rangle \\
& \overset{0.5}{\rightharpoondown} \langle \mathsf{switch}(\mathsf{t2}(\mathsf{heads}, \mathsf{coin2})), \{(R_1, 1)\} \rangle \\
& \rightharpoondown \langle \mathsf{t2}(\mathsf{heads}, \mathsf{tails}), \{(R_1, 1)\} \rangle
\end{array}
$$

that computes the explanation $\kappa_3 = \{(R_1, 1)\}$ with $P(\kappa_3) = 0.5$. Here, given $\kappa_1 = \{(R_1, 1), (R_2, 2)\}$ computed before, we have $\mathcal{W}(\kappa_1) \cap \mathcal{W}(\kappa_3) \neq \emptyset$. Analogously, the derivation:

$$
\begin{array}{rl}
\langle \mathsf{switch}(\mathsf{main}), \emptyset \rangle & \rightharpoondown \langle \mathsf{switch}(\mathsf{flip2coins}(\mathsf{coin1}, \mathsf{coin2})), \emptyset \rangle \\
& \rightharpoondown \langle \mathsf{switch}(\mathsf{t2}(\mathsf{coin1}, \mathsf{coin2})), \emptyset \rangle \\
& \overset{0.6}{\rightharpoondown} \langle \mathsf{switch}(\mathsf{t2}(\mathsf{coin1}, \mathsf{heads})), \{(R_2, 1)\} \rangle \\
& \rightharpoondown \langle \mathsf{t2}(\mathsf{heads}, \mathsf{tails}), \{(R_1, 1)\} \rangle
\end{array}
$$

computes the explanation $\kappa_4 = \{(R_2, 1)\}$ with $P(\kappa_4) = 0.6$. If we add up all their probabilities, we get $P(\kappa_1) + P(\kappa_2) + P(\kappa_3) + P(\kappa_4) = 0.2 + 0.3 + 0.5 + 0.6 = 1.6$, which is obviously incorrect.

As in the literature on probabilistic logic programming (see, e.g., [14]), we can compile the covering set of explanations to a BDD (or some of its more efficient variants). Consider, for instance, the above set of covering explanations:

$$K = \{\{(R_1, 1)\}, \{(R_2, 1)\}, \{(R_1, 1), (R_2, 2)\}, \{(R_1, 2), (R_2, 1)\}\}$$

Here, since probabilistic rules only have two choices, we can associate a Boolean variable to each rule, so that X_1 represents the first choice in rule R_1 (heads) while $\neg X_1$ represents the second choice in this rule (tails), and similarly X_2 (first choice of rule R_2, i.e.,

[4] A similar result for ICL can be found in [19].

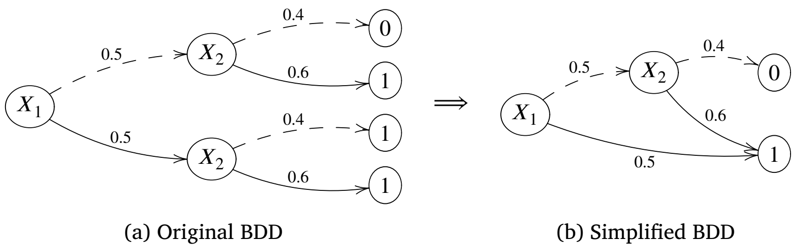


Fig. 2. BDD for formula $X_1 \vee X_2 \vee (X_1 \wedge \neg X_2) \vee (\neg X_1 \wedge X_2)$.

heads) and $\neg X_2$ (second choice of rule R_2, i.e., tails). Then, the probability of the considered reachability problem is given by the probability of the following formula:

$$f_K(\mathbf{X}) = X_1 \vee X_2 \vee (X_1 \wedge \neg X_2) \vee (\neg X_1 \wedge X_2)$$

that is obtained from the set K. In turn, this formula can be represented with the BDD of Fig. 2 (a), where edges are labeled with the associated probability. As usual, a solid line issuing from a node represents that the variable is true and a dashed line that it is false. This BDD can be simplified as shown in Fig. 2 (b) by avoiding repeated nodes and deleting those nodes where both edges reach the same value (a so-called Reduced Ordered BDD).

The probability of the formula being true can now be computed as the sum of the probabilities of the paths from X_1 to 1, where the probability of a path is given by the product of the probabilities of its edges. Therefore, $P(\mathsf{switch}(\mathsf{main}) \to^* \mathsf{t2}(\mathsf{heads}, \mathsf{tails})) = 0.5 + 0.5 \cdot 0.6 = 0.8$.

This technique can also be generalized to arbitrary rules with more than two choices using *multivalued* decision diagrams (MDDs, [34]). Computing the probability from a covering set of explanations is out of the scope of this work. Nevertheless, the process is essentially identical to that in the literature on (exact) inference in probabilistic logic programming; thus we refer the interested reader to [35, Chapter 8] and references therein.

Finally, for those problems where exact inference is not viable, we can consider *approximate* inference. We distinguish two possible approaches:

- The first technique adapts [14,36] and is based on using iterative deepening when computing a covering set of explanations. Given a reachability problem $s \to^* t$ and a given *depth* n, the following two sets of explanations are produced[5]:

$$\begin{aligned}
&\mathbb{E}_l(s \to^*_{\mathcal{S}} t) = \{\kappa \mid \langle s, \emptyset\rangle \rightharpoonup^m_{\mathcal{S}} \langle t, \kappa\rangle,\ m \leqslant n\}\\
&\mathbb{E}_u(s \to^*_{\mathcal{S}} t) = \mathbb{E}_l(s \to^*_{\mathcal{S}} t) \cup \{\kappa \mid \langle s, \emptyset\rangle \rightharpoonup^n_{\mathcal{S}} \langle t', \kappa\rangle \ \wedge\ t \neq t'\}
\end{aligned}$$

  Here, $\mathbb{E}_l$ computes the explanations associated with those derivations of length less than or equal to n that reach the term t, while $\mathbb{E}_u$ includes, in addition to the previous explanations, those associated with derivations that are still incomplete after n reduction steps. By computing the probability associated to each set (using the same techniques from exact inference seen before), one gets an interval $[p_u, p_l]$ for the probability. If the accuracy is deemed adequate, the process ends. Otherwise, one can increase the depth n and repeat the process.
- Another technique is based on Monte Carlo sampling (see, e.g., [35, Chapter 10]). In this case, the idea is to generate a sample of possible worlds by considering random choices for the probabilistic rewrite rules. Then, we check whether $s \to^* t$ can be proved in each generated world. The probability is then obtained as a fraction of the worlds where $s \to^* t$ could be proved. Here, the number of worlds determines the confidence interval, so the more worlds considered, the more accurate the computed probability will be.

4. Modeling problems with uncertainty

In this section, we explore the expressive power of PTRSs for modeling problems with uncertainty and propose some extensions.

Example 18. Consider the PTRS shown in Fig. 3 which specifies the level of protection (against covid-19) depending on the vaccination status and the mask worn. Note that vaccination status depends on a person's age while wearing a mask is considered a generic property which is independent of age in this example. Here, we may be interested in knowing the probability that a senior person has strong protection against covid, i.e., $P(\mathsf{protection}(\mathsf{senior}) \to^* \mathsf{strong})$.

For this purpose, we get a set of explanations K from the probabilistic rewrite derivations shown in Fig. 4. The computed set of explanations is thus as follows:

[5] We use $\langle s, \kappa\rangle \rightharpoonup^n \langle t, \kappa'\rangle$ to denote that $\langle s, \kappa\rangle$ is reduced to $\langle t, \kappa'\rangle$ in exactly n steps.

$$
\begin{array}{lrcl}
R_1: & \mathsf{status(senior)} & \to & 0.9 : \mathsf{vacc}; 0.1 : \mathsf{non\_vacc} \\
R_2: & \mathsf{status(adult)} & \to & 0.5 : \mathsf{vacc}; 0.5 : \mathsf{non\_vacc} \\
R_3: & \mathsf{status(young)} & \to & 0.2 : \mathsf{vacc}; 0.8 : \mathsf{non\_vacc} \\
R_4: & \mathsf{mask} & \to & 0.2 : \mathsf{ffp2}; 0.1 : \mathsf{surgical}; 0.7 : \mathsf{none} \\
R_5: & \mathsf{protection}(x) & \to & \mathsf{lookup}(\mathsf{status}(x), \mathsf{mask}) \\
R_6: & \mathsf{lookup(vacc, ffp2)} & \to & \mathsf{strong} \\
R_7: & \mathsf{lookup(vacc, surgical)} & \to & \mathsf{strong} \\
R_8: & \mathsf{lookup(non\_vacc, ffp2)} & \to & \mathsf{strong} \\
R_9: & \mathsf{lookup(non\_vacc, surgical)} & \to & \mathsf{weak} \\
R_{10}: & \mathsf{lookup}(x, \mathsf{none}) & \to & \mathsf{no}
\end{array}
$$

Fig. 3. A PTRS specifying the level of protection against covid-19.

$$
\begin{array}{rcl}
\langle \mathsf{protection(senior)}, \emptyset \rangle & \rightharpoonup & \langle \mathsf{lookup(status(senior), mask)}, \emptyset \rangle \\
& \stackrel{0.9}{\rightharpoonup} & \langle \mathsf{lookup(vacc, mask)}, \{(R_1, 1)\} \rangle \\
& \stackrel{0.2}{\rightharpoonup} & \langle \mathsf{lookup(vacc, ffp2)}, \{(R_1, 1), (R_4, 1)\} \rangle \\
& \rightharpoonup & \langle \mathsf{strong}, \{(R_1, 1), (R_4, 1)\} \rangle \\
\langle \mathsf{protection(senior)}, \emptyset \rangle & \rightharpoonup & \langle \mathsf{lookup(status(senior), mask)}, \emptyset \rangle \\
& \stackrel{0.9}{\rightharpoonup} & \langle \mathsf{lookup(vacc, mask)}, \{(R_1, 1)\} \rangle \\
& \stackrel{0.1}{\rightharpoonup} & \langle \mathsf{lookup(vacc, surgical)}, \{(R_1, 1), (R_4, 2)\} \rangle \\
& \rightharpoonup & \langle \mathsf{strong}, \{(R_1, 1), (R_4, 2)\} \rangle \\
\langle \mathsf{protection(senior)}, \emptyset \rangle & \rightharpoonup & \langle \mathsf{lookup(status(senior), mask)}, \emptyset \rangle \\
& \stackrel{0.1}{\rightharpoonup} & \langle \mathsf{lookup(no\_vacc, mask)}, \{(R_1, 2)\} \rangle \\
& \stackrel{0.2}{\rightharpoonup} & \langle \mathsf{lookup(no\_vacc, ffp2)}, \{(R_1, 2), (R_4, 1)\} \rangle \\
& \rightharpoonup & \langle \mathsf{strong}, \{(R_1, 2), (R_4, 1)\} \rangle
\end{array}
$$

Fig. 4. Probabilistic derivations for $\mathsf{protection(senior)} \to^* \mathsf{strong}$.

$$
K = \{\underbrace{\{(R_1,1),(R_4,1)\}}_{\kappa_1},\ \underbrace{\{(R_1,1),(R_4,2)\}}_{\kappa_2},\ \underbrace{\{(R_1,2),(R_4,1)\}}_{\kappa_3}\}
$$

with $P(\kappa_1) = 0.18$, $P(\kappa_2) = 0.09$ and $P(\kappa_3) = 0.02$. In this case, the explanations in K are already mutually incompatible and, thus, we have $P(\mathsf{protection(senior)} \to^* \mathsf{strong}) = P(\kappa_1) + P(\kappa_2) + P(\kappa_3) = 0.29$.

4.1. Probabilistic rules with variables

Variables in rewrite rules represent a very expressive resource. In the previous section, we have only considered probabilistic *ground* rules. Fortunately, there are several ways we can extend the definition of probabilistic rewrite rule to also include variables. First, one can have probabilistic rules with variables as long as the domain of such variables is finite. Here, probabilistic rules with variables are seen only as a compact representation of a (finite) set of ground rules. These rules can then be replaced by the corresponding instances in a *grounding* preprocessing stage. For this purpose, introducing types would be useful (as in simply typed term rewriting [37]).

Here, we extend the notion of atomic choice to a triple (R, θ, i) where R, i are the rule and the choice (as before), and the ground substitution θ identifies the considered (ground) instance of rule R. The associated probability is still $\mathsf{p}(R, i)$ since all instances have the same choices and probabilities. In this case, we can say that each ground instance of each probabilistic rule represents a random variable of the model.

Example 19. Consider again the PTRS of Example 18. Now, however, we consider that function mask takes the age group (senior, adult, and young) as argument. Thus, we have the PTRS shown in Fig. 5.

We assume that, according to the type of function mask, it can only take an argument from $\{\mathsf{senior}, \mathsf{adult}, \mathsf{young}\}$. Then, rule R_4 will be replaced in a grounding stage by

$$
\begin{array}{rcl}
\mathsf{mask(senior)} & \to & 0.2 : \mathsf{ffp2}; 0.1 : \mathsf{surgical}; 0.7 : \mathsf{none} \\
\mathsf{mask(adult)} & \to & 0.2 : \mathsf{ffp2}; 0.1 : \mathsf{surgical}; 0.7 : \mathsf{none} \\
\mathsf{mask(young)} & \to & 0.2 : \mathsf{ffp2}; 0.1 : \mathsf{surgical}; 0.7 : \mathsf{none}
\end{array}
$$

Note that, in contrast to Example 18 where everyone wore the same type of mask, now we can have different choices for each age group.

$$
\begin{array}{rrcl}
R_1: & \mathsf{status(senior)} & \to & 0.9:\mathsf{vacc}; 0.1:\mathsf{non\_vacc}\\
R_2: & \mathsf{status(adult)} & \to & 0.5:\mathsf{vacc}; 0.5:\mathsf{non\_vacc}\\
R_3: & \mathsf{status(young)} & \to & 0.2:\mathsf{vacc}; 0.8:\mathsf{non\_vacc}\\
R_4: & \mathsf{mask}(x) & \to & 0.2:\mathsf{ffp2}; 0.1:\mathsf{surgical}; 0.7:\mathsf{none}\\
R_5: & \mathsf{protection}(x) & \to & \mathsf{lookup}(\mathsf{status}(x),\mathsf{mask}(x))\\
R_6: & \mathsf{lookup(vacc,ffp2)} & \to & \mathsf{strong}\\
R_7: & \mathsf{lookup(vacc,surgical)} & \to & \mathsf{strong}\\
R_8: & \mathsf{lookup(non\_vacc,ffp2)} & \to & \mathsf{strong}\\
R_9: & \mathsf{lookup(non\_vacc,surgical)} & \to & \mathsf{weak}\\
R_{10}: & \mathsf{lookup}(x,\mathsf{none}) & \to & \mathsf{no}
\end{array}
$$

Fig. 5. New version of the PTRS of Example 18.

In principle, the case above does not imply any relevant change with respect to the theory seen in Section 3, since the same conditions apply after the grounding stage.

Let us now consider the general case in which variables can be bound to infinitely many different terms and, thus, it makes no sense to consider a grounding stage. In this case, a PTRS encodes a possibly infinite number of random variables (as many as ground instances of the probabilistic rules exist). Therefore, the probability of a world cannot be defined as the product of the choices in a selection, $P(\mathcal{W}_\kappa) = \prod_{(R,\theta,i)\in\kappa} \mathsf{p}(R,i)$, since it would converge to zero whenever a selection includes infinitely many choices.

Following [19], in order to overcome the above problem, one can introduce a measure over (possibly infinite) *sets* of possible worlds:

- Given a finite set of (finite) composite choices K, the probability of the (possibly infinite) set of worlds $\mathcal{W}(K)$ is given by $\mu(K)$, which is defined in terms of a set K' of mutually incompatible composite choices that is *equivalent* to K, i.e., that represents the same (possibly infinite) set of worlds, $\mathcal{W}(K) = \mathcal{W}(K')$. Then, $\mu(K) = \sum_{\kappa\in K'} P(\kappa) = \sum_{\kappa\in K'} \prod_{(R,\theta,i)\in\kappa} \mathsf{p}(R,i)$. Such a set K' can always be obtained from K by *splitting* [19], a transformation that applies the following two operations repeatedly until a mutually incompatible set is obtained:
  1. If $\kappa_1,\kappa_2 \in K$ with $\kappa_1 \subset \kappa_2$, then remove κ_2 from K.
  2. If $\kappa_1,\kappa_2 \in K$ with $\kappa_1 \cup \kappa_2$ consistent, find some probabilistic rule R which only occurs in κ_1 and replace κ_2 in K by the sets $\kappa_2 \cup \{(R,1)\}, \ldots, \kappa_2 \cup \{(R,n)\}$ if R has n possible choices.
- Then, given a reachability problem, $s \to_S^* t$ and a finite set of (finite) covering explanations K, we let $P(s \to_S^* t) = \mu(K)$.

Therefore, introducing probabilistic rules with variables over possibly infinite domains is not a problem as long as we can obtain a finite covering set of (finite) explanations. In particular, we would need θ to be ground in every atomic choice (R,θ,i). For this purpose, we could require, e.g., that the term to be reduced is ground (and, consequently, all the terms in its derivations since the considered PTRSs have no extra variables), which is a reasonable requirement in practice.

Example 20. Consider the PTRS of Example 19 (with no grounding stage). For the reachability problem $\mathsf{protection(senior)} \to^* \mathsf{strong}$, we can generate the following probabilistic rewrite derivations:

$$
\begin{array}{rcl}
\langle \mathsf{protection(senior)}, \emptyset\rangle & \rightharpoondown & \langle \mathsf{lookup(status(senior),mask(senior))}, \emptyset\rangle\\
& \xrightharpoondown{0.9} & \langle \mathsf{lookup(vacc,mask(senior))}, \{(R_1,id,1)\}\rangle\\
& \xrightharpoondown{0.2} & \langle \mathsf{lookup(vacc,ffp2)}, \{(R_1,id,1),(R_4,\{x/\mathsf{senior}\},1)\}\rangle\\
& \rightharpoondown & \langle \mathsf{strong}, \{(R_1,id,1),(R_4,\{x/\mathsf{senior}\},1)\}\rangle\\
\langle \mathsf{protection(senior)}, \emptyset\rangle & \rightharpoondown & \langle \mathsf{lookup(status(senior),mask(senior))}, \emptyset\rangle\\
& \xrightharpoondown{0.9} & \langle \mathsf{lookup(vacc,mask(senior))}, \{(R_1,id,1)\}\rangle\\
& \xrightharpoondown{0.1} & \langle \mathsf{lookup(vacc,surgical)}, \{(R_1,id,1),(R_4,\{x/\mathsf{senior}\},2)\}\rangle\\
& \rightharpoondown & \langle \mathsf{strong}, \{(R_1,id,1),(R_4,\{x/\mathsf{senior}\},2)\}\rangle\\
\langle \mathsf{protection(senior)}, \emptyset\rangle & \rightharpoondown & \langle \mathsf{lookup(status(senior),mask(senior))}, \emptyset\rangle\\
& \xrightharpoondown{0.1} & \langle \mathsf{lookup(no\_vacc,mask(senior))}, \{(R_1,id,2)\}\rangle\\
& \xrightharpoondown{0.2} & \langle \mathsf{lookup(no\_vacc,ffp2)}, \{(R_1,id,2),(R_4,\{x/\mathsf{senior}\},1)\}\rangle\\
& \rightharpoondown & \langle \mathsf{strong}, \{(R_1,id,2),(R_4,\{x/\mathsf{senior}\},1)\}\rangle
\end{array}
$$

The computed set $K = \{\kappa_1,\kappa_2,\kappa_3\}$ of explanations is now as follows:

$$
\begin{array}{rcll}
\kappa_1 & = & \{(R_1,id,1),(R_4,\{x/\mathsf{senior}\},1)\}, & \text{with } P(\kappa_1) = 0.9\cdot 0.2 = 0.18\\
\kappa_2 & = & \{(R_1,id,1),(R_4,\{x/\mathsf{senior}\},2)\}, & \text{with } P(\kappa_2) = 0.9\cdot 0.1 = 0.09\\
\kappa_3 & = & \{(R_1,id,2),(R_4,\{x/\mathsf{senior}\},1)\}, & \text{with } P(\kappa_3) = 0.1\cdot 0.2 = 0.02
\end{array}
$$

and, thus, the probability of protection(senior) $\to^*$ strong is the same as before, i.e., $P(\text{protection(senior)} \to^* \text{strong}) = P(\kappa_1) + P(\kappa_2) + P(\kappa_3) = 0.29$.

4.2. Conditional PTRSs

Another interesting extension from the point of view of expressive power consists of considering conditional rewrite rules. For this purpose, let us first briefly introduce *conditional* term rewrite systems (CTRSs); namely oriented 3-CTRSs, i.e., CTRSs where extra variables are allowed as long as $\mathcal{V}\mathsf{ar}(r) \subseteq \mathcal{V}\mathsf{ar}(l) \cup \mathcal{V}\mathsf{ar}(C)$ for any rule $l \to r \Leftarrow C$ [38]. In *oriented* CTRSs, a conditional rule $l \to r \Leftarrow C$ has the form $l \to r \Leftarrow s_1 \twoheadrightarrow t_1, \dots, s_m \twoheadrightarrow t_m$, where each oriented equation $s_i \twoheadrightarrow t_i$ is interpreted as reachability.

For a CTRS $\mathcal{R}$, the associated rewrite relation $\to_{\mathcal{R}}$ is defined as the smallest binary relation satisfying the following: given terms $s, t \in \mathcal{T}(\mathcal{F})$, we have $s \to_{\mathcal{R}} t$ iff there exist a position p in s, a rewrite rule $l \to r \Leftarrow s_1 \twoheadrightarrow t_1, \dots, s_m \twoheadrightarrow t_m \in \mathcal{R}$, and a substitution σ such that $s|_p = l\sigma$, $s_i\sigma \to^*_{\mathcal{R}} t_i\sigma$ for all $i = 1, \dots, m$, and $t = s[r\sigma]_p$.

The extension of PTRSs and probabilistic rewriting to the conditional case is very natural. A probabilistic *conditional* rule has now the form

$$R : l \to p_1 : r_1; \dots; p_n : r_n \Leftarrow s_1 \twoheadrightarrow t_1, \dots, s_m \twoheadrightarrow t_m$$

A *probabilistic conditional term rewrite system* (PCTRS) is a disjoint union of a set of probabilistic conditional rules and a set of regular conditional rewrite rules. A *world* is then obtained from a selection κ as in the unconditional case, i.e., by choosing one right-hand side per probabilistic conditional rule[6]:

$$\mathcal{W}_\kappa = \{l \to r_i \Leftarrow C \mid R : (l \to p_1 : r_1; \dots; p_n : r_n \leftarrow C) \in \mathcal{S}_p \wedge (R, i) \in \kappa\} \cup \mathcal{S}_r$$

where C is a (possibly empty) sequence of oriented equations of the form $s_i \twoheadrightarrow t_i$, $i = 1, \dots, m$, $m \geq 0$. Definition 6 applies to PCTRSs too: a property P holds for PCTRS $\mathcal{S}$ if P holds for each world $\mathcal{W} \in \mathbb{W}_{\mathcal{S}}$.

Probabilistic conditional rewriting can then be defined as follows:

Definition 21 *(probabilistic conditional rewriting).* Let s, t be terms and κ, κ' be composite choices. A PCTRS $\mathcal{S}$ induces a *probabilistic conditional rewrite* relation $\rightharpoonup_{\mathcal{S}}$ where $\langle s, \kappa \rangle \rightharpoonup_{\mathcal{S}} \langle t, \kappa' \rangle$ if either

- there is a position π in s, a regular rewrite rule $l \to r \Leftarrow s_1 \twoheadrightarrow t_1, \dots, s_m \twoheadrightarrow t_m \in \mathcal{S}$, and a substitution σ such that $s|_\pi = l\sigma$, $\langle s_i\sigma, \kappa \rangle \rightharpoonup^*_{\mathcal{S}} \langle t_i\sigma, \kappa_i \rangle$ for all $i = 1, \dots, m$, $t = s[r\sigma]_\pi$, and $\kappa' = \kappa_1 \cup \ldots \cup \kappa_m$ is consistent, or
- there is a position π in s, a probabilistic rewrite rule $R : l \to p_1 : r_1; \dots; p_n : r_n \Leftarrow s_1 \twoheadrightarrow t_1, \dots, s_m \twoheadrightarrow t_m \in \mathcal{S}$, and a substitution σ such that $s|_\pi = l\sigma$, $\langle s_i\sigma, \kappa \rangle \rightharpoonup^*_{\mathcal{S}} \langle t_i\sigma, \kappa_i \rangle$ for all $i = 1, \dots, m$, $t = s[r_j\sigma]_\pi$, $j \in \{1, \dots, n\}$, and $\kappa' = \kappa_1 \cup \ldots \cup \kappa_m \cup \{(R, j)\}$ is consistent.

We label probabilistic conditional steps with a probability, as in the unconditional case: given $\langle s, \kappa \rangle \xrightharpoonup{p} \langle t, \kappa' \rangle$, we have $p = P(\kappa'')$ if $\kappa' \setminus \kappa = \kappa'' \neq \emptyset$, and $p = 1$ otherwise. The only difference is that, now, we can add more than one atomic choice in a single step (because of the evaluation of the conditions). Probabilities are then lifted to derivations by computing the product of the probabilities of their steps.

Example 22. Consider the following PCTRS:

$$\begin{array}{rrcl}
R_1 : & \mathsf{crime} & \to & 0.2 : \mathsf{burglary}; 0.8 : \mathsf{other} \\
R_2 : & \mathsf{earthquake} & \to & 0.2 : \mathsf{strong}; 0.5 : \mathsf{mild}; 0.3 : \mathsf{none} \\
R_3 : & \mathsf{alarm} & \to & 0.9 : \mathsf{ring} \Leftarrow \mathsf{crime} \twoheadrightarrow \mathsf{burglary}, \mathsf{earthquake} \twoheadrightarrow \mathsf{strong} \\
R_4 : & \mathsf{alarm} & \to & 0.8 : \mathsf{ring} \Leftarrow \mathsf{crime} \twoheadrightarrow \mathsf{burglary}, \mathsf{earthquake} \twoheadrightarrow \mathsf{mild} \\
R_5 : & \mathsf{alarm} & \to & 0.7 : \mathsf{ring} \Leftarrow \mathsf{crime} \twoheadrightarrow \mathsf{burglary}, \mathsf{earthquake} \twoheadrightarrow \mathsf{none} \\
R_6 : & \mathsf{alarm} & \to & 0.3 : \mathsf{ring} \Leftarrow \mathsf{crime} \twoheadrightarrow \mathsf{other}, \mathsf{earthquake} \twoheadrightarrow \mathsf{strong} \\
R_7 : & \mathsf{alarm} & \to & 0.1 : \mathsf{ring} \Leftarrow \mathsf{crime} \twoheadrightarrow \mathsf{other}, \mathsf{earthquake} \twoheadrightarrow \mathsf{mild}
\end{array}$$

We note that “other” in rule R_1 represents either other types of crimes and no crime at all. Also, notice that the right-hand sides of rules $R_3 - R_7$ have an implicit choice (i.e., $\bot$) representing that the alarm does not ring.

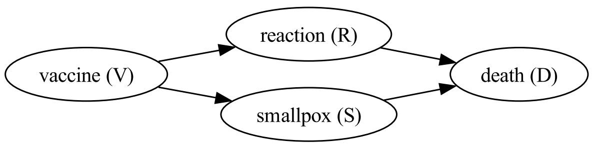
In this case, we are interested in computing the probability that the alarm rings, i.e., that alarm can be reduced to ring. Here, we have the probabilistic conditional rewrite derivations shown in Fig. 6. The probability labeling the first step of derivation D_1 is obtained from the product $\mathsf{p}(R_1, 1) \cdot \mathsf{p}(R_2, 1) \cdot \mathsf{p}(R_3, 1) = 0.2 \cdot 0.2 \cdot 0.9 = 0.036$, and analogously for the remaining derivations.

Let $K = \{\kappa_1, \dots, \kappa_5\}$ be the computed set of explanations, where k_i is the explanation computed by derivation D_i, $i = 1, \dots, 5$. Then, we have $P(\kappa_1) = 0.036$, $P(\kappa_2) = 0.08$, $P(\kappa_3) = 0.042$, $P(\kappa_4) = 0.048$, and $P(\kappa_5) = 0.04$. Hence, the most likely explanation for the alarm to ring is κ_2, i.e., there was a burglary and, in parallel, a mild earthquake. The overall probability that the alarm rings is $P(\mathsf{alarm} \to^* \mathsf{ring}) = P(\kappa_1) + P(\kappa_2) + P(\kappa_3) + P(\kappa_4) + P(\kappa_5) = 0.246$ since K is not only covering but also mutually incompatible.

[6] Here, we assume that probabilistic conditional rules are ground for simplicity, but the same considerations of Section 4.1 regarding variables apply.

$$
\begin{aligned}
D_1 = \langle \mathsf{alarm}, \emptyset \rangle &\overset{0.036}{\to} \langle \mathsf{ring}, \{(R_1,1),(R_2,1),(R_3,1)\} \rangle \\
&\quad \text{since } \langle \mathsf{crime}, \emptyset \rangle \overset{0.2}{\to} \langle \mathsf{burglary}, \{(R_1,1)\} \rangle \\
&\quad \text{and } \langle \mathsf{earthquake}, \emptyset \rangle \overset{0.2}{\to} \langle \mathsf{strong}, \{(R_2,1)\} \rangle \\
D_2 = \langle \mathsf{alarm}, \emptyset \rangle &\overset{0.08}{\to} \langle \mathsf{ring}, \{(R_1,1),(R_2,2),(R_4,1)\} \rangle \\
&\quad \text{since } \langle \mathsf{crime}, \emptyset \rangle \overset{0.2}{\to} \langle \mathsf{burglary}, \{(R_1,1)\} \rangle \\
&\quad \text{and } \langle \mathsf{earthquake}, \emptyset \rangle \overset{0.5}{\to} \langle \mathsf{mild}, \{(R_2,2)\} \rangle \\
D_3 = \langle \mathsf{alarm}, \emptyset \rangle &\overset{0.042}{\to} \langle \mathsf{ring}, \{(R_1,1),(R_2,3),(R_5,1)\} \rangle \\
&\quad \text{since } \langle \mathsf{crime}, \emptyset \rangle \overset{0.2}{\to} \langle \mathsf{burglary}, \{(R_1,1)\} \rangle \\
&\quad \text{and } \langle \mathsf{earthquake}, \emptyset \rangle \overset{0.3}{\to} \langle \mathsf{none}, \{(R_2,3)\} \rangle \\
D_4 = \langle \mathsf{alarm}, \emptyset \rangle &\overset{0.048}{\to} \langle \mathsf{ring}, \{(R_1,2),(R_2,1),(R_6,1)\} \rangle \\
&\quad \text{since } \langle \mathsf{crime}, \emptyset \rangle \overset{0.8}{\to} \langle \mathsf{other}, \{(R_1,2)\} \rangle \\
&\quad \text{and } \langle \mathsf{earthquake}, \emptyset \rangle \overset{0.2}{\to} \langle \mathsf{strong}, \{(R_2,1)\} \rangle \\
D_5 = \langle \mathsf{alarm}, \emptyset \rangle &\overset{0.04}{\to} \langle \mathsf{ring}, \{(R_1,2),(R_2,2),(R_7,1)\} \rangle \\
&\quad \text{since } \langle \mathsf{crime}, \emptyset \rangle \overset{0.8}{\to} \langle \mathsf{other}, \{(R_1,2)\} \rangle \\
&\quad \text{and } \langle \mathsf{earthquake}, \emptyset \rangle \overset{0.5}{\to} \langle \mathsf{mild}, \{(R_2,2)\} \rangle
\end{aligned}
$$

Fig. 6. Probabilistic derivations for $\mathsf{alarm} \to^* \mathsf{ring}$.



V	t	f
	0.8	0.2

R	t	f
V=t	0.05	0.95
V=f	0	1.0

S	t	f
V=t	0.01	0.99
V=f	0.3	0.7

D	t	f
R=t,S=t	0.2	0.8
R=t,S=f	0.01	0.99
R=f,S=t	0.3	0.7
R=f,S=f	0	1.0

Fig. 7. Bayesian network modeling the probability of dying from smallpox.

Similarly to the case of probabilistic logic programming, we can also encode Bayesian networks [39] using PCTRSs. Basically, each random variable is encoded as a Boolean function and each row of its conditional probability table is encoded as a conditional rule as follows:

Example 23. Consider the Bayesian network shown in Fig. 7 (freely adapted from [40]). It can be encoded using the following PCTRS:

$$
\begin{aligned}
R_1: &\quad \mathsf{vaccine} \to 0.8 : \mathsf{true}; 0.2 : \mathsf{false} \\[4pt]
R_2: &\quad \mathsf{reaction} \to 0.05 : \mathsf{true}; 0.95 : \mathsf{false} \Leftarrow \mathsf{vaccine} \twoheadrightarrow \mathsf{true} \\
R_3: &\quad \mathsf{reaction} \to 0 : \mathsf{true}; 1 : \mathsf{false} \Leftarrow \mathsf{vaccine} \twoheadrightarrow \mathsf{false} \\[4pt]
R_4: &\quad \mathsf{smallpox} \to 0.01 : \mathsf{true}; 0.99 : \mathsf{false} \Leftarrow \mathsf{vaccine} \twoheadrightarrow \mathsf{true} \\
R_5: &\quad \mathsf{smallpox} \to 0.3 : \mathsf{true}; 0.7 : \mathsf{false} \Leftarrow \mathsf{vaccine} \twoheadrightarrow \mathsf{false} \\[4pt]
R_6: &\quad \mathsf{death} \to 0.2 : \mathsf{true}; 0.8 : \mathsf{false} \Leftarrow \mathsf{reaction} \twoheadrightarrow \mathsf{true}, \mathsf{smallpox} \twoheadrightarrow \mathsf{true} \\
R_7: &\quad \mathsf{death} \to 0.01 : \mathsf{true}; 0.99 : \mathsf{false} \Leftarrow \mathsf{reaction} \twoheadrightarrow \mathsf{true}, \mathsf{smallpox} \twoheadrightarrow \mathsf{false} \\
R_8: &\quad \mathsf{death} \to 0.3 : \mathsf{true}; 0.7 : \mathsf{false} \Leftarrow \mathsf{reaction} \twoheadrightarrow \mathsf{false}, \mathsf{smallpox} \twoheadrightarrow \mathsf{true} \\
R_9: &\quad \mathsf{death} \to 0 : \mathsf{true}; 1 : \mathsf{false} \Leftarrow \mathsf{reaction} \twoheadrightarrow \mathsf{false}, \mathsf{smallpox} \twoheadrightarrow \mathsf{false}
\end{aligned}
$$

Here, we are interested in the probability of death, i.e., $P(\mathsf{death} \to^* \mathsf{true})$. In this case, there exist many possible derivations, some of them shown in Fig. 8.

$$
\begin{aligned}
D_1 = \langle \mathsf{death}, \emptyset \rangle & \overset{0.00008}{\to} \langle \mathsf{true}, \{(R_1,1),(R_2,1),(R_4,1),(R_6,1)\}\rangle \\
& \quad \text{since } \langle \mathsf{reaction}, \emptyset\rangle \overset{0.04}{\to} \langle \mathsf{true}, \{(R_1,1),(R_2,1)\}\rangle \\
& \qquad \text{since } \langle \mathsf{vaccine}, \emptyset\rangle \overset{0.8}{\to} \langle \mathsf{true}, \{(R_1,1)\}\rangle \\
& \quad \text{and } \langle \mathsf{smallpox}, \emptyset\rangle \overset{0.008}{\to} \langle \mathsf{true}, \{(R_1,1),(R_4,1)\}\rangle \\
& \qquad \text{since } \langle \mathsf{vaccine}, \emptyset\rangle \overset{0.8}{\to} \langle \mathsf{true}, \{(R_1,1)\}\rangle \\
D_2 = \langle \mathsf{death}, \emptyset \rangle & \overset{0.000396}{\to} \langle \mathsf{true}, \{(R_1,1),(R_2,1),(R_4,2),(R_7,1)\}\rangle \\
& \quad \text{since } \langle \mathsf{reaction}, \emptyset\rangle \overset{0.04}{\to} \langle \mathsf{true}, \{(R_1,1),(R_2,1)\}\rangle \\
& \qquad \text{since } \langle \mathsf{vaccine}, \emptyset\rangle \overset{0.8}{\to} \langle \mathsf{true}, \{(R_1,1)\}\rangle \\
& \quad \text{and } \langle \mathsf{smallpox}, \emptyset\rangle \overset{0.792}{\to} \langle \mathsf{false}, \{(R_1,1),(R_4,2)\}\rangle \\
& \qquad \text{since } \langle \mathsf{vaccine}, \emptyset\rangle \overset{0.8}{\to} \langle \mathsf{true}, \{(R_1,1)\}\rangle \\
& \ldots
\end{aligned}
$$

Fig. 8. Some probabilistic derivations for death $\to^*$ true.

It is worth noting that this encoding works well because, once the choice is fixed for a probabilistic rule, it cannot be changed in the remaining derivation. For example, in derivation D_1, if the reduction from reaction to true (in the condition of rule R_6) considers that vaccine reduces to true in order to apply rule R_2, then the reduction from smallpox to true can only consider rule R_4 (where vaccine should also be reduced to true). Here, rule R_5 would no longer be applicable because that rule requires vaccine to be reduced to false. This is a relevant aspect that other approaches—like those of Bournez et al. [2–4] and more recent works based on them, e.g., [5,8–10]—to probabilistic rewriting do not consider; typically, in these works, the same probabilistic rule can be used repeatedly throughout a derivation, even selecting different choices in different steps.

Note that a potential advantage of PCTRSs over Bayesian networks is that, in a PCTRS, it is possible to combine a Bayesian network with the definition of arbitrary (possibly recursive) functions.

5. Related work

As mentioned in the Introduction, the most popular approach to probabilistic rewriting is that of Bournez et al. [2–4]. The notions of probabilistic abstract reduction system and probabilistic term rewrite systems have been later refined in [5], where *multidistributions* are introduced to appropriately account for both the probabilistic behavior of each rule and the nondeterminism in rule selection. A similar notion of PTRS is considered in [8–10]. Our definition of PTRS is closely related, but our work has a number of significant differences. First, we partition the rules of a PTRS into a set of probabilistic rules and a set of regular rules, and require the probabilistic rules to be ground. This is required for the distribution semantics to be well-defined (although an extension to non-ground probabilistic rules is possible along the lines of Section 4.1).

Secondly, the probability of a reduction is defined in these works as the product of the probabilities of the rewrite steps in this reduction. This is significantly different in our distribution semantics, where we consider the probability of a reduction to be the product of the atomic choices computed in this reduction. In particular, this implies the following key differences:

- First, once we apply a probabilistic rule and choose one of the terms in the right-hand side, no other choice can be considered in the remaining steps of the same derivation. Note that this difference becomes relevant, e.g., to model problems like the Bayesian network of Example 23.
- On the other hand, if we apply a probabilistic rule several times in the same derivation, its probability is computed only once.

Finally, while these works focus on studying properties such as termination or confluence of the PTRSs, we emphasize their use for modeling problems with a certain degree of uncertainty.

Other approaches to defining a probabilistic extension of rewriting include [41], where the aim is to assign probabilities (or weights) to (ordinary) rewrite rules, or an extension of Maude [42] where extra-variables in the right-hand sides of the rules are instantiated according to some probability distribution [43]. These works consider different ways of integrating probabilities in term rewriting and, in particular, different from our proposal.

Finally, we can also find probability extensions in some related, rule-based formalisms like the λ-calculus (see, e.g., [44], and references therein) and constraint handling rules [45]. In particular, the language Church [46] has a memoization mode in which, once a probabilistic function returning a certain value is used, subsequent calls will always return the same value, ignoring all other choices. This mode resembles what happens in our distribution semantics, where the application of a probabilistic rule in a given reduction is considered a one-time event that cannot be changed later during the same reduction. However, Church is a language with a very different orientation than our PTRSs. Assume, e.g., that we define in Church a function coin that can return heads or tails with equal probability. This means that a call to this function will randomly return either heads or tails (but only one value). Moreover, if we call function coin a sufficiently large number of times, we will observe that the obtained values are uniformly distributed. In other words, Church can be used for describing stochastic generative processes, so that the output of a program follows a particular distribution. In contrast, if we define a probabilistic rule like $\mathsf{coin} \to 0.5 : \mathsf{heads}; 0.5 : \mathsf{tails}$, the evaluation of coin is actually deterministic and always return *both* possible values, heads and tails, each one with its associated probability. In other words, PTRSs are used to specify models and perform probabilistic inference.

6. Concluding remarks and future work

In this paper, we have presented a new approach to combine term rewriting and probabilities, resulting in a more expressive formalism that can model complex relationships and uncertainty. We have defined a distribution semantics inspired to Sato's proposal for logic programs [13], which forms the basis of several probabilistic logic languages (e.g., ProbLog [14], Logic Programs with Annotated Disjunctions [15], Independent Choice Logic [16] or PRISM [17]). We thus think that a distribution semantics can also play a significant role in the context of term rewriting. We have also shown how the probability of a reduction can be obtained from a set of (covering) explanations. Finally, we have illustrated the expressive power of our approach by means of several examples, and outlined some possible extensions.

As future work, we consider it interesting to determine the conditions under which a PTRS/PCTRS will always produce a set of mutually incompatible explanations, which would allow us to calculate the probability of a reachability problem very easily as the sum of the probabilities of its explanations. Another useful extension involves the design of appropriate probabilistic rewrite strategies that prune derivations when the reachability problem is guaranteed to be infeasible, e.g., by reusing existing results from reachability analysis. This will allow us to address the implementation of a software tool to calculate the probability of a reachability problem. Finally, we will also consider the definition of methods to present the computed explanations in a way that is understandable for non-experts, similarly to what we have done in [47] for probabilistic logic programs.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Germán Vidal reports equipment, drugs, or supplies and travel were provided by Spanish Ministry of Science, Innovation and Universities. Germán Vidal reports equipment, drugs, or supplies and travel were provided by Generalitat Valenciana. Germán Vidal reports equipment, drugs, or supplies and travel were provided by EU Horizon 2020. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

I would like to thank the anonymous reviewers for their suggestions to improve this paper. Funding for open access charge: CRUE-Universitat Politècnica de València, Spain.

Data availability

No data was used for the research described in the article.

References

[1] Terese, Term Rewriting Systems, Cambridge Tracts in Theoretical Computer Science, vol. 55, Cambridge University Press, 2003.
[2] O. Bournez, C. Kirchner, Probabilistic rewrite strategies. Applications to ELAN, in: S. Tison (Ed.), RTA 2002, in: Lecture Notes in Computer Science, vol. 2378, Springer, 2002, pp. 252–266.
[3] O. Bournez, F. Garnier, Proving positive almost-sure termination, in: J. Giesl (Ed.), RTA 2005, in: Lecture Notes in Computer Science, vol. 3467, Springer, 2005, pp. 323–337.
[4] O. Bournez, F. Garnier, Proving positive almost sure termination under strategies, in: F. Pfenning (Ed.), RTA 2006, in: Lecture Notes in Computer Science, vol. 4098, Springer, 2006, pp. 357–371.
[5] M. Avanzini, U.D. Lago, A. Yamada, On probabilistic term rewriting, Sci. Comput. Program. 185 (2020), https://doi.org/10.1016/J.SCICO.2019.102338.
[6] N. Saheb-Djahromi, Probabilistic LCF, in: J. Winkowski (Ed.), Proceedings of the 7th Symposium on Mathematical Foundations of Computer Science, in: Lecture Notes in Computer Science, vol. 64, Springer, 1978, pp. 442–451.
[7] A. Díaz-Caro, G. Martínez, Confluence in probabilistic rewriting, in: S. Alves, R. Wasserman (Eds.), 12th Workshop on Logical and Semantic Frameworks, with Applications, LSFA 2017, in: Electronic Notes in Theoretical Computer Science, vol. 338, Elsevier, 2017, pp. 115–131.
[8] M.H. Kirkeby, H. Christiansen, Confluence and convergence in probabilistically terminating reduction systems, in: F. Fioravanti, J.P. Gallagher (Eds.), LOPSTR 2017, in: Lecture Notes in Computer Science, vol. 10855, Springer, 2017, pp. 164–179.
[9] M.H. Kirkeby, H. Christiansen, Confluence and convergence modulo equivalence in probabilistically terminating reduction systems, Int. J. Approx. Reason. 105 (2019) 217–228, https://doi.org/10.1016/J.IJAR.2018.11.018.
[10] J. Kassing, F. Frohn, J. Giesl, From innermost to full almost-sure termination of probabilistic term rewriting, in: N. Kobayashi, J. Worrell (Eds.), FoSSaCS 2024, in: Lecture Notes in Computer Science, vol. 14575, Springer, 2024, pp. 206–228.
[11] J. Kassing, S. Dollase, J. Giesl, A complete dependency pair framework for almost-sure innermost termination of probabilistic term rewriting, in: J. Gibbons, D. Miller (Eds.), FLOPS 2024, in: Lecture Notes in Computer Science, vol. 14659, Springer, 2024, pp. 62–80.
[12] C. Faggian, Probabilistic rewriting and asymptotic behaviour: on termination and unique normal forms, Log. Methods Comput. Sci. 18 (2) (2022), https://doi.org/10.46298/LMCS-18(2:5)2022.
[13] T. Sato, A statistical learning method for logic programs with distribution semantics, in: L. Sterling (Ed.), Logic Programming, Proceedings of the Twelfth International Conference on Logic Programming, Tokyo, Japan, June 13–16, 1995, MIT Press, 1995, pp. 715–729.

[14] L.D. Raedt, A. Kimmig, H. Toivonen, ProbLog: a probabilistic Prolog and its application in link discovery, in: M.M. Veloso (Ed.), Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007), 2007, pp. 2462–2467, http://ijcai.org/Proceedings/07/Papers/396.pdf.

[15] J. Vennekens, S. Verbaeten, M. Bruynooghe, Logic programs with annotated disjunctions, in: B. Demoen, V. Lifschitz (Eds.), Logic Programming, 20th International Conference, ICLP 2004, Saint-Malo, France, September 6-10, 2004, Proceedings, in: Lecture Notes in Computer Science, vol. 3132, Springer, 2004, pp. 431–445.

[16] D. Poole, The independent choice logic for modelling multiple agents under uncertainty, Artif. Intell. 94 (1–2) (1997) 7–56, https://doi.org/10.1016/S0004-3702(97)00027-1.

[17] T. Sato, Y. Kameya, PRISM: a language for symbolic-statistical modeling, in: Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, IJCAI 97, Nagoya, Japan, August 23-29, 1997, 2 Volumes, Morgan Kaufmann, 1997, pp. 1330–1339, http://ijcai.org/Proceedings/97-2/Papers/078.pdf, 1997.

[18] F. Baader, T. Nipkow, Term Rewriting and All That, Cambridge University Press, 1998.

[19] D. Poole, Abducing through negation as failure: stable models within the independent choice logic, J. Log. Program. 44 (1–3) (2000) 5–35, https://doi.org/10.1016/S0743-1066(99)00071-0.

[20] A. Darwiche, P. Marquis, A knowledge compilation map, J. Artif. Intell. Res. 17 (2002) 229–264, https://doi.org/10.1613/JAIR.989.

[21] R.E. Bryant, Graph-based algorithms for Boolean function manipulation, IEEE Trans. Comput. 35 (8) (1986) 677–691, https://doi.org/10.1109/TC.1986.1676819.

[22] M. Chavira, A. Darwiche, On probabilistic inference by weighted model counting, Artif. Intell. 172 (6–7) (2008) 772–799, https://doi.org/10.1016/J.ARTINT.2007.11.002.

[23] F.G. Cozman, D.D. Mauá, On the semantics and complexity of probabilistic logic programs, J. Artif. Intell. Res. 60 (2017) 221–262, https://doi.org/10.1613/JAIR.5482.

[24] J. Vlasselaer, A. Kimmig, A. Dries, W. Meert, L.D. Raedt, Knowledge compilation and weighted model counting for inference in probabilistic logic programs, in: A. Darwiche (Ed.), Beyond NP, Papers from the 2016 AAAI Workshop, vol. WS-16-05 of AAAI Technical Report, AAAI Press, 2016, http://www.aaai.org/ocs/index.php/WS/AAAIW16/paper/view/12631.

[25] D. Poole, Probabilistic Horn abduction and Bayesian networks, Artif. Intell. 64 (1) (1993) 81–129, https://doi.org/10.1016/0004-3702(93)90061-F.

[26] G. Feuillade, T. Genet, V.V.T. Tong, Reachability analysis over term rewriting systems, J. Autom. Reason. 33 (3–4) (2004) 341–383, https://doi.org/10.1007/S10817-004-6246-0.

[27] C. Sternagel, A. Yamada, Reachability analysis for termination and confluence of rewriting, in: T. Vojnar, L. Zhang (Eds.), 25th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, TACAS 2019, in: Lecture Notes in Computer Science, vol. 11427, Springer, 2019, pp. 262–278.

[28] M. Oyamaguchi, The reachability problem for quasi-ground term rewriting systems, J. Inf. Process. 9 (1986) 232–236, https://api.semanticscholar.org/CorpusID:58721911.

[29] M. Alpuente, M. Falaschi, M.J. Ramis, G. Vidal, Narrowing approximations as an optimization for equational logic programs, in: M. Bruynooghe, J. Penjam (Eds.), Proceedings of the 5th International Symposium on Programming Language Implementation and Logic Programming, PLILP'93, in: Lecture Notes in Computer Science, vol. 714, Springer, 1993, pp. 391–409.

[30] J.-M. Hullot, Canonical forms and unification, in: W. Bibel, R.A. Kowalski (Eds.), Proceedings of the 5th International Conference on Automated Deduction, in: Lecture Notes in Computer Science, vol. 87, Springer, 1980, pp. 318–334.

[31] T. Arts, Termination by absence of infinite chains of dependency pairs, in: H. Kirchner (Ed.), Proceedings of the 21st International Colloquium on Trees in Algebra and Programming, CAAP'96, in: Lecture Notes in Computer Science, vol. 1059, Springer, 1996, pp. 196–210.

[32] T. Arts, J. Giesl, Termination of term rewriting using dependency pairs, Theor. Comput. Sci. 236 (1–2) (2000) 133–178, https://doi.org/10.1016/S0304-3975(99)00207-8.

[33] L. Fribourg, Handling function definitions through innermost superposition and rewriting, in: J. Jouannaud (Ed.), Proceedings of the First International Conference on Rewriting Techniques and Applications, RTA-85, in: Lecture Notes in Computer Science, vol. 202, Springer, 1985, pp. 325–344.

[34] A. Srinivasan, T. Kam, S. Malik, R.K. Brayton, Algorithms for discrete function manipulation, in: IEEE/ACM International Conference on Computer-Aided Design, ICCAD 1990, IEEE Computer Society, 1990, pp. 92–95.

[35] F. Riguzzi, Foundations of Probabilistic Logic Programming: Languages, Semantics, Inference and Learning, 2nd ed., River Publishers, 2023.

[36] A. Kimmig, V.S. Costa, R. Rocha, B. Demoen, L.D. Raedt, On the efficient execution of ProbLog programs, in: M.G. de la Banda, E. Pontelli (Eds.), Proceedings of the 24th International Conference on Logic Programming, ICLP 2008, in: Lecture Notes in Computer Science, vol. 5366, Springer, 2008, pp. 175–189.

[37] T. Yamada, Confluence and termination of simply typed term rewriting systems, in: A. Middeldorp (Ed.), Rewriting Techniques and Applications, 12th International Conference, RTA 2001, in: Lecture Notes in Computer Science, vol. 2051, Springer, 2001, pp. 338–352.

[38] A. Middeldorp, E. Hamoen, Completeness results for basic narrowing, Appl. Algebra Eng. Commun. Comput. 5 (1994) 213–253.

[39] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, 1988.

[40] J. Pearl, D. Mackenzie, The Book of Why: The New Science of Cause and Effect, 1st edition, Basic Books, Inc., USA, 2018.

[41] O. Bournez, M. Hoyrup, Rewriting logic and probabilities, in: R. Nieuwenhuis (Ed.), RTA 2003, in: Lecture Notes in Computer Science, vol. 2706, Springer, 2003, pp. 61–75.

[42] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, J.F. Quesada, Maude: specification and programming in rewriting logic, Theor. Comput. Sci. 285 (2) (2002) 187–243, https://doi.org/10.1016/S0304-3975(01)00359-0.

[43] G.A. Agha, J. Meseguer, K. Sen, Pmaude: rewrite-based specification language for probabilistic object systems, in: A. Cerone, H. Wiklicky (Eds.), Proceedings of the Third Workshop on Quantitative Aspects of Programming Languages, QAPL 2005, in: Electronic Notes in Theoretical Computer Science, vol. (2), Elsevier, 2005, pp. 213–239.

[44] U.D. Lago, M. Zorzi, Probabilistic operational semantics for the lambda calculus, RAIRO Theor. Inform. Appl. 46 (3) (2012) 413–450, https://doi.org/10.1051/ITA/2012012.

[45] T.W. Frühwirth, A.D. Pierro, H. Wiklicky, Probabilistic constraint handling rules, in: M. Comini, M. Falaschi (Eds.), 11th International Workshop on Functional and (Constraint) Logic Programming, WFLP 2002, in: Electronic Notes in Theoretical Computer Science, vol. 76, Elsevier, 2002, pp. 115–130.

[46] N.D. Goodman, V.K. Mansinghka, D.M. Roy, K.A. Bonawitz, J.B. Tenenbaum, Church: a language for generative models, in: D.A. McAllester, P. Myllymäki (Eds.), Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence, UAI 2008, AUAI Press, 2008, pp. 220–229, https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=1346&proceeding_id=24.

[47] G. Vidal, Explaining explanations in probabilistic logic programming, in: O. Kiselyov (Ed.), Programming Languages and Systems - 22nd Asian Symposium, APLAS 2024, in: Lecture Notes in Computer Science, vol. 15194, Springer, 2024, pp. 130–152.