Rewriting Logic and Probabilities

Olivier Bournez[1] and Mathieu Hoyrup[2]

[1] LORIA/INRIA, 615 Rue du Jardin Botanique
54602 Villers lès Nancy Cedex, France
[2] ENS-Lyon, 46 Allée d'Italie
69364 Lyon Cedex 07, France

Abstract Rewriting Logic has shown to provide a general and elegant framework for unifying a wide variety of models, including concurrency models and deduction systems. In order to extend the modeling capabilities of rule based languages, it is natural to consider that the firing of rules can be subject to some probabilistic laws. Considering rewrite rules subject to probabilities leads to numerous questions about the underlying notions and results. In this paper, we discuss whether there exists a notion of probabilistic rewrite system with an associated notion of probabilistic rewriting logic.

1 Introduction

Rewriting Logic [19] is known to provide a very elegant and powerful framework for unifying a wide variety of models, including concurrency models and deduction systems. Indeed, the basic axioms of this logic, which are rewrite rules of the form $t \to t'$, where t and t' are terms over a given signature, can be read in two dual ways: *computationally*, $t \to t'$ can be read as the local transition of a concurrent system or *logically*, $t \to t'$ can be read as the inference rule of some logic [19]. Several computer systems, including MAUDE [11] and ELAN [7], are based on this framework and have been intensively used in the last decade for the prototyping of various kinds of logics and systems: see survey [18].

In order to extend the modeling capabilities of rule based languages, it seems natural to extend the framework with probabilities: for example, the modeling of concurrent systems requires often to consider that the local transitions $t \to t'$ can be subject to some probabilistic laws [8]. This leads to numerous questions about the underlying theories and results.

In a previous RTA paper [8], *strategies* were shown to provide a nice setting for expressing probabilistic choices in rule based languages. *Probabilistic abstract reduction systems* and notions like *almost-sure termination* or *probabilistic confluence* were introduced and related to the classical notions.

This paper is devoted to a next step: understand whether there exists a valid and useful notion of rewrite system and rewriting logic in presence of probabilities.

In classical (non-probabilistic) rewriting theory each rewrite system induces a reduction relation which defines the relation of an abstract reduction system

over the terms: see e.g. [2]. When considering systems with probabilistic firing of rules, the analog of abstract reduction systems seems to be the notion of probabilistic abstract reduction systems introduced in [8]. Can we build a valid and nice notion of probabilistic rewrite system, that would induce probabilistic abstract reduction systems over terms in a natural way?

A first natural idea seems to consider the following notion: define a probabilistic rewrite system as a classical rewrite system, i.e. a set of rewrite rules, plus associated probabilities (or weights see discussions later) : i.e. a probability (or a weight) for each rule.

In the classical setting, the reflexive transitive closure of the relation induced by some rewrite system can be proved to correspond to the smallest reflexive transitive relation that contains the identities involved by the rewrite system and which is closed by substitutions and Σ-operations: see e.g. [2]. That means in particular that one can build a sound and complete proof system that decides if two terms are in relation by the reflexive transitive closure of the reduction relation of a given rewrite system. This proof system corresponds to the deduction rules of Rewriting Logic [19]. Does that work in the probabilistic settings?

We prove in this paper that there is no hope to build a sound and complete proof system that would prove whether two terms are in relation by the reflexive transitive closure of the reduction relation of a given rewrite system with associated probabilities in the general case.Does there exist however a notion of probabilistic rewriting logic?

We propose a notion of probabilistic rewriting logic. One main difference between the proposed setting and the classical rewriting logic setting is that proof terms become now mandatory, in order to have completeness results: we prove that when proof terms are present, probabilistic rewriting logic is sound and complete.

One main interest of rewriting logic lies in its modeling capabilities [18,19]. We show that the proposed probabilistic rewriting logic extends the modeling capabilities of classical rewriting logic.

This paper is organized as follows: classical non-probabilistic theory is recalled in Section 2. Probabilistic abstract reduction systems are recalled in Section 3. Several computability theory results which show that this is not way to have sound and complete proof systems that deal correctly with transitivity are proved in Section 4. The proposed notion of probabilistic rewrite system with its associated semantic is introduced in Section 5. The associated sound and complete probabilistic rewriting logic is discussed in Section 6. The modeling capabilities of probabilistic rewriting logic are exemplified in Section 7. Section 8 discusses related and future work.

2 Rewriting Logic

We need first to recall some classical notions and results (we follow the notations and terminology from [2]): $T(\Sigma, X)$ denotes the set of terms over signature Σ and disjoint set of variables X. When $t \in T(\Sigma, X)$ is a term, let $Pos(t)$ be the

set of its positions. For $\rho \in Pos(t)$, let $t|_\rho$ be the subterm of t at position ρ, and let $t[s]_\rho$ denote the replacement of the subterm at position ρ in t by s. The set of all substitutions is denoted by Sub.

Definition 1 (Labeled rewrite system). *A* labeled rewrite system $(\mathcal{R}, \mathcal{L})$ *consists of a set* $\mathcal{R} \subseteq T(\Sigma, X) \times T(\Sigma, X)$ *of* rules *and a set* $\mathcal{L}$ *of* labels*, such that each rule in* $\mathcal{R}$ *is bijectively associated to a label in* $\mathcal{L}$*. We write* $g \to d \in \mathcal{R}$ *for* $(g, d) \in \mathcal{R}$ *and* $(l : g \to d)$ *when* $l \in \mathcal{L}$ *is associated to* $g \to d \in \mathcal{R}$*.*

Definition 2 (Abstract Reduction System). *An* abstract reduction system $(A, \to)$ *consists of a set* A *and a binary relation* $\to$ *on* A*, called* reduction relation*. We write* $a \to b$ *for* $(a, b) \in \to$*, and we write* $\to^*$ *for the reflexive transitive closure of* $\to$*.*

Definition 3 (Reduction relation). *Let* $\mathcal{R}$ *be a rewrite system. The* associated reduction relation $\to_\mathcal{R} \subseteq T(\Sigma, X) \times T(\Sigma, X)$*, also denoted by* $\to$ *when* $\mathcal{R}$ *is clear, is defined by* $t \to_\mathcal{R} t'$ *iff* $\exists (g \to d) \in \mathcal{R}, p \in Pos(t), \sigma \in Sub$*, such that* $t|_p = \sigma(g)$ *and* $t' = t[\sigma(d)]_p$*.*

A rule $g \to d \in \mathcal{R}$ *will be said to be* applicable at the root of term t *if position* p *can be chosen as the root position: i.e. there is a substitution* $\sigma \in Sub$ *with* $t = \sigma(g)$*. In that case,* $\sigma(d)$ *is the* the result *of its application.*

The idea of rewriting logic is, for a given rewrite system $\mathcal{R}$, to consider $\to_\mathcal{R}$ as the description of a transition system over terms.

Definition 4. *The* executional semantic *of a given rewrite system* $\mathcal{R}$ *is the abstract reduction system* $\mathcal{S}_\mathcal{R} = (T(\Sigma, X), \to_\mathcal{R})$*.*

The derivations of this abstract reduction system correspond to the provable sequents of a logic, called *rewriting logic*. This logic talks about sentences of the form $t \to t'$, meaning that t can evolve toward t' in $\mathcal{S}_\mathcal{R}$ [19].

Proposition 1. *[There exists a sound and complete proof system for* $\to^*$ *[19]] Suppose rewrite system* $\mathcal{R}$ *is fixed. Two terms* $s, t \in T(\Sigma, X)$ *are related by* $\to^*$ *iff* $t \to t'$ *can be established starting with axioms* $l \to r$ *for each rule* $l \to r \in \mathcal{R}$ *by the following proof system:*

Reflexivity : *if* $t \in T(\Sigma, X)$*,*

$$\frac{}{t \to t}$$

Congruence : *if* $f \in \Sigma_n$*,*

$$\frac{t_1 \to t'_1 \quad \cdots \quad t_n \to t'_n}{f(t_1, \ldots, t_n) \to f(t'_1, \ldots, t'_n)}$$

Replacement : *if* $l : g(x_1, ..., x_n) \to d(x_1, ..., x_n) \in \mathcal{R}$*,*

$$\frac{t_1 \to t'_1 \quad \cdots \quad t_n \to t'_n}{g(t_1, \ldots, t_n) \to d(t'_1, \ldots, t'_n)}$$

Transitivity :

$$\frac{t_1 \to t_2 \qquad t_2 \to t_3}{t_1 \to t_3}$$

Remark 1. Rewriting logic is generally defined considering *rewriting modulo*: sequents correspond to quotient set $T(\Sigma, X)/_E$ where E is a given set of identities [19]. In this paper, we will not consider terms modulo a congruence class. Furthermore, we will not allow conditional rules. We believe this restricted framework to be enough interesting by itself for the following discussions.

Remark 2. In order to represent both a reduction and the proof tree that induces this reduction, *proof terms* can also be considered: the set $\mathcal{PT}$ of proof terms is defined as the set $T(\Sigma \cup \mathcal{L} \cup \{;\}, X)$ of terms on the signature Σ extended with the labels of $\mathcal{L}$ and the binary concatenation operator ";" [10]. Rewriting logic deduction rules can then be adapted to derive sentences of the form $\pi : t \to t'$ meaning that t evolves toward t' in $\mathcal{S}_\mathcal{R}$ using path encoded by proof term π': see [10,19]. But, as shown by previous proposition, unless one wants to define the notion of model [19], or the notion of strategy [10], proof terms are not mandatory.

3 Probabilistic Abstract Reduction Systems

Let S be a countable finite or infinite set. A *stochastic sequence* $(X_n)_{n\geq 0}$ on S is a family of random variables from some fixed probability space to S.

Definition 5 (Homogeneous Markovian Stochastic Sequence). *A stochastic sequence $(X_n)_{n\geq 0}$ is Markovian if its conditional distribution function satisfies $\forall n \geq 1, i_0, \ldots, i_n \in S$, $p(X_n = i_n | X_{n-1} = i_{n-1}, \ldots, X_0 = i_0) = p(X_n = i_n | X_{n-1} = i_{n-1})$. It is said to be* homogeneous *if furthermore this probability is independent of n.*

In other words, Markov property means that the system evolution does not depend on past, but only on present state. The homogeneity property means that the dynamic is independent of time.

In that case, $P = (p_{i,j})_{i,j\in S}$ defined by $p_{i,j} = p(X_n = j | X_{n-1} = i)$ is a *stochastic matrix on S*: i.e. it satisfies for all $i, j \in S$, $p_{ij} \in [0, 1]$ and for all i, $\sum_j p_{ij} = 1$. It is called a matrix even when S is infinite. Homogeneous Markovian stochastic sequences (HMSS) and stochastic matrices are in correspondence, since conversely to any stochastic matrix $P = (p_{i,j})_{i,j\in S}$ corresponds a homogeneous Markovian stochastic sequence: if at time n the system state is $i \in S$, choose at time $(n + 1)$ system state j with probability $p_{i,j}$.

In [8], we suggested to extend abstract reduction systems in a homogeneous Markovian way:

Definition 6 (PARS). *A Probabilistic Abstract Reduction System $\mathcal{A} = (A, \rightsquigarrow)$ consists of a countable (finite or infinite) set A and a mapping $\rightsquigarrow$ from $A \times A$ to $[0, 1]$ such that for all $s \in A$, $\sum_{t\in A} s \rightsquigarrow t = 0$ or 1.*

A PARS $\mathcal{A}$ is like a HMSS on A whose stochastic matrix is $P = (s \rightsquigarrow t)_{s,t}$. However contrary to a HMSS, a state can be *irreducible*, that is such that $\sum_{t \in A} s \rightsquigarrow t = 0$. Actually, a PARS can be transformed into a stochastic matrix by adding a new state $\perp$ and reducing irreducible states to $\perp$: let $S = A \cup \{\perp\}$ the extension of A with $\perp$. Extend $\rightsquigarrow$ on $S \times S$ by

$$\begin{array}{ll} s \rightsquigarrow \perp = 1 & \text{if s} \in \text{A is irreducible} \\ s \rightsquigarrow \perp = 0 & \text{if s} \in \text{A is reducible} \\ \perp \rightsquigarrow t = 0 & \text{for all t} \in \text{A} \\ \perp \rightsquigarrow \perp = 1 & \end{array}$$

Definition 7 (Derivation). *A derivation of $\mathcal{A}$ is a corresponding HMSS on S.*

PARS correspond to the extension of Abstract Reduction Systems (ARS) with probabilities. Indeed, to a PARS $\mathcal{A} = (A, \rightsquigarrow)$ can be associated a unique ARS $(A, \rightarrow)$, called its *projection,* obtained by forgetting probabilities: $s \rightarrow t$ if and only if $s \rightsquigarrow t > 0$. Conversely, to any ARS can be associated several PARS by distributing probabilities over the possible derivations: the projection of these PARS will be the original ARS: see [8] for a full discussion.

4 Probabilities and Transitivity

We now come to the main object of this paper, that is to discuss whether there exists a notion of *probabilistic rewrite system* with some *executional semantic* for which there exists some associated notion of *probabilistic rewriting logic.*

We have not yet defined what *probabilistic rewrite systems* are, but one may expect a *probabilistic rewrite system* to correspond to a classical rewrite system with somehow the addition of probabilities. One may also expect its *executional semantic* to be defined as a probabilistic abstract reduction system over terms. In other words, one would expect to define *probabilistic rewrite systems* and their *executional semantics* by distributing in some manner the probabilities over the executional semantic of classical rewrite systems.

The point is to get something "nice": one may in particular want to have results in the spirit of Proposition 1: there is some associated sound and complete proof system that could derive whether two terms are related in the corresponding *executional semantic.* If it were so, guided by classical theory, we would then call this complete proof system *probabilistic rewriting logic.*

However, we prove in this section that there is no hope to get such a sound and complete proof system.

We start by a computability theory result about homogeneous Markovian stochastic sequences: observe that, when P is a stochastic matrix, and n is an integer, P^n is a stochastic matrix whose entries $(P^n)_{i,j}$ give the probability of going from i to j in n steps [9]. We show that even two steps transitions, that is P^2, is not computable in the general case: a stochastic matrix $P = (P_{i,j})_{i,j}$ is said *recursive* if all its entries are rational and there exists a Turing machine that given i, j outputs $P_{i,j}$. Such a matrix can be represented by an index of a corresponding Turing Machine.

Theorem 1. *The decision problem "given a stochastic matrix P, and some rational q, decide if top-left entry of P^2 is q" is not recursively enumerable.*

Proof. The halting problem "given integer w, decide if Turing machine number w accepts input w" is recursively enumerable non-recursive, and hence, its complement $Co-Halt$ can not be recursively enumerable. We only need to prove that problem $Co-Halt$ reduces to our problem.

Given an input w of $Co-Halt$, consider the matrix $P=(P_{i,j})_{i,j}$ where $P_{1,j}=\frac{1}{2^j}$ for all j, $P_{i,j}=0$ for all $j>2$, $P_{i,1}=0$ (respectively: $P_{i,2}=1$) if Turing machine number w over input w halts in less than i steps, $\frac{1}{2^i}$ otherwise (resp. $1-\frac{1}{2^i}$ otherwise). P is a recursive stochastic matrix: all its entries are computable rationals of $[0,1]$, and for $i=1$, we have $\sum_{j\geq 1} p_{1j}=\sum_{j\geq 1}\frac{1}{2^j}=1$, and for $i>1$, $\sum_{j\geq 1} p_{ij}$ is $p_{i1}+p_{i2}=0+1$ or $\frac{1}{2^i}+(1-\frac{1}{2^i})=1$ according to whether Turing machine number w stops on input w in less than i steps or not.

Assume that Turing machine number w does not accept input w. For all $i>1$ we have $p_{i1}=\frac{1}{2^i}$ and $p_{i2}=1-\frac{1}{2^i}$. The top-left entry of P^2 is given by $\sum_{k\geq 1} p_{1k}p_{k1}=\sum_{k\geq 1}\left(\frac{1}{2^k}\right)^2=\frac{1}{3}$.

Assume that Turing machine number w accepts input w at time i_0.We have $p_{i1}=\frac{1}{2^i}$ and $p_{i2}=1-\frac{1}{2^i}$ for all $i\leq i_0$ and $p_{i1}=0$ and $p_{i2}=1$ for all $i>i_0$. After a certain row, the first column elements of matrix P are 0 and the second column elements are 1. The top-left entry of P^2 is given by $\sum_{k\geq 1} p_{1k}p_{k1}=\sum_{k=1}^{i_0}\left(\frac{1}{2^{i_0}}\right)^2=\frac{1}{3}\left(1-\frac{1}{4^{i_0}}\right)<\frac{1}{3}$.

Hence, problem $Co-Halt$ reduces to our problem considering matrix P and rational $q=1/3$.

Remark 3. The previous proof also shows that the problem of determining if the top-left entry of P^2 is $\geq q$ is not recursively enumerable. The problem of determining whether it is $>q$ can be shown to be recursively enumerable but non recursive.

We now come back to rewriting and probabilities. A point is that one expects the notion of *probabilistic rewrite system* to cover at least homogeneous Markovian stochastic sequences: indeed, any stochastic matrix $P=(p_{i,j})_{i,j}$ on set of states S can be considered as a rewrite system with probabilities: take a constant for each element $i\in S$ and write a rule $i\to j$ with associated probability $p_{i,j}$ for each i,j.

Suppose there were a sound and complete proof system that could derive whether two terms are related in the *executional semantic* of a given *probabilistic rewrite system.* It is rather natural to expect this proof system not only to talk about whether there is a path between two terms in the executional semantic but also to talk about the probability of this path: otherwise it would have nothing to do with probabilities. In other words, it is natural to expect such a proof system to derive sentences of type $t\rightsquigarrow_p^* t'$ (or $t\rightsquigarrow_p^n t'$) meaning "term t can evolve to term t with probability p (respectively in n steps).

We show this is impossible (observe that you can fix $n=2$ in what follows):

Theorem 2 (There is no sound and complete proof system for $\rightsquigarrow^n$). *There is no way to conceive a sound and complete proof system (axioms + deduction rules) that could derive in the general case for all terms s,t and integer n the probability $s \rightsquigarrow^n t$ of going from s to t in n steps.*

Proof. Assume there were a finite (or even a recursively enumerable) set of axioms and a finite (or even recursively enumerable) set of deductions rules that would allow to give probabilities $s \rightsquigarrow^n t$ for all s, t, n. By enumerating recursively axioms and proofs we could enumerate all the possible proofs. Hence, the problem "given some probabilistic rewrite system, some terms s, t and some rational q, decide if $q = s \rightsquigarrow^n t$" would be recursively enumerable. This is in contradiction with Theorem 1 considering systems describing a homogeneous Markovian stochastic sequence.

One may argue that the previous arguments relies on systems with a non-finite set of rules, or that we do not talk about reachability in any number of steps. Actually, we prove:

Theorem 3. *The decision problem "given a PARS represented by a finite set of rewrite rules with probabilities, some states s,t, decide if the probability $s \rightsquigarrow^* t$ of going from s to t in any number of steps is q" is not recursively enumerable.*

Proof. We only need to reduce non-recursively enumerable decision problem $Co-Halt$ to our problem. Let $E \subseteq \mathbb{N}^2$ be the set of couples (w, t) such that Turing Machine number w halts on input w in less than t steps. E is a recursive set. By Bergstra-Tucker theorem [6], there exists a confluent rewriting system on a signature $\Sigma \supset \{0, s, In\}$, where 0 is a constant symbol, s is an unary (successor) function symbol, and In is a binary function symbol, such that for all $x, t \geq 0$,

$$\begin{array}{rl} In(s^x(0), s^t(0)) \rightarrow^* & 0 \quad \text{if } (x,t) \in E \\ \rightarrow^* & s(0) \quad \text{if } (x,t) \notin E \end{array}$$

Consider signature $\Sigma' = \Sigma \cup \{F, Run\}$, where F is binary, Run unary (and these symbols are not in Σ). Consider the rewrite system $\mathcal{R}$ composed of the rules of the rewriting system associated to E plus the rules:

$$\begin{array}{rl} Run(x) & \rightarrow F(x, 0) \\ F(x,t) & \rightarrow F(x, s(t)) \\ F(x,t) & \rightarrow In(x,t) \end{array}$$

Build a PARS on $T(\Sigma', X)$ by assigning probabilities to the reductions of $\mathcal{R}$: put probability 1/2 on the reductions $F(x,t) \rightarrow F(x, s(t))$ and $F(x,t) \rightarrow In(x,t)$, and probability 1 on all other reductions.

By construction, the probability $p(w)$ that $Run(s^w(0))$ reduces to $s(0)$ is

$$\sum_{n | (w,n) \notin E} \frac{1}{2^{n+1}}.$$

Indeed, a reduction that leads to $s(0)$ can be written as

$$F(s^w(0),0) \to \cdots \to F(s^w(0),s^n(0)) \to In(s^w(0),s^n(0)) \to \cdots \to s(0)$$

and the probability of such a reduction is $\frac{1}{2^{n+1}}$.

Observing definition of E, probability $p(w)$ is 1 iff $w \in Co-Halt$, and is < 1 otherwise. Hence, problem $Co-Halt$ reduces to our problem.

Remark 4. The previous proof also shows that the problem of deciding whether $s \rightsquigarrow^* t$ is $\geq q$ is non-recursively enumerable. Deciding whether it is $> q$ can be shown to be recursively enumerable but non recursive.

Using similar arguments to those used to establish Theorem 2, we get.

Theorem 4 (There is no sound and complete proof system for $\rightsquigarrow^*$). *Even when restricting to systems described by a finite set of rewriting rules, there is no way to conceive a proof system that could derive in the general case for all terms s,t the probability $s \rightsquigarrow^* t$ of going from s to t.*

5 Probabilistic Rewrite Systems

We now present the notion of *probabilistic rewrite system* with the associated notion of *executional semantic* that we propose.

The rules that can be applied on some term t depend on t. For example for the following rewrite system

$$\mathcal{R} \begin{cases} l_1 : f(a,x) \to x \\ l_2 : f(x,b) \to c \end{cases}$$

over signature $\Sigma = \{f,a,b,c\}$, on term $f(a,b)$ both rules l_1 and l_2 apply, but on term $f(a,a)$ only rule l_1 applies.

Furthermore, on a given term t, one may have the choice to apply a rule at the root of the term, that is to use *replacement* rule, or to rewrite concurrently only (one or several) subterms, that is to use *congruence* rule.

We would like to distribute probabilities over the possibilities: a first difficulty is that we can not hardwire directly probabilities: if we wanted to put probability p_i to rule l_i, for $i = 1,2$, on term $f(a,b)$ we would expect $p_1 + p_2 = 1$, on term $f(a,a)$ we would expect $p_1 = 1$. This is impossible unless p_2 is 0, i.e. l_2 never applies.

Our proposition is to consider that we do not assign probabilities but *weights*: a weight is some positive real number. The following strategy is then proposed: on a term t, choose some applicable possibility (that is to say a rule that applies at the root of t or *congruence* rule for symbol f if term t is of type $t = f(t_1,\ldots,t_n)$ and some of the t_i is reducible) selecting possibility i of weight w_i with probability $\frac{w_i}{w}$, where w is the sum of the weights of applicable possibilities.

This strategy, even if often considered to avoid problems (see e.g. [14]), which requires to normalize weights to have true probabilities, and then choosing an applicable solution accordingly, may seem artificial.

However, we claim that this is equivalent to a more natural strategy: since the previous strategy is unchanged if all weights are multiplied by some real positive constant, assume that weights w_i are chosen such that $\sum_i w_i = 1$. It can then be also obtained as follows: on a term t, choose *any* possibility selecting possibility i with probability w_i. As long as the chosen possibility can not be applied to t, repeat. When one succeeds to get one that applies to t, apply it.

This is indeed a restatement of following easy observation.

Proposition 2. *Suppose that we have n alternatives that can be partitioned into "bad ones" and "good ones". Suppose that weights $w_1, \ldots, w_n$ (i.e. positive real numbers) are assigned to the alternatives in such a way that $\sum_i w_i = 1$.*

Then the following algorithm:

1. Choose $l \in \{1, \ldots, n\}$ selecting i with probability w_i.
2. If alternative number l is a bad one, then repeat: i.e. goto 1.
3. Answer "alternative number l".

never stops if there is no good alternative, returns with probability 1 some good one otherwise, returning alternative number i with probability $\frac{w_i}{\sum_{j \text{ good alternative}} w_j}$.

The following problem remains: suppose $t = f(t_1, \ldots, t_n)$ and *congruence* is chosen. In the spirit of classical rewriting logic, we want to allow concurrent rewriting, that is to allow several of the t_i to be rewritten simultaneously. How should we distribute probabilities? We propose to choose the subterms in an independent way. Indeed, n probabilities $q_1^f, \ldots, q_n^f$ (i.e. n real numbers of $[0, 1]$) are associated to each function symbol of the signature of arity n: in an application of *congruence* rule, subterm t_i will be chosen to be rewritten with probability q_i^f. One technical point is that we assume that always at least one subterm is rewritten, and hence the probabilities are probabilities conditioned by this fact.

In a same spirit, we want to allow concurrent rewriting of subterms in application of *replacement* rule. We assume that all the variables in the right member of a rule $l : g \to r$ of the rewrite system appear in the left member. Every rule can then be written as $l : g(x_1, \ldots, x_n, \ldots, x_{n+k}) \to r(x_1, \ldots, x_n)$ where variables $x_1, \ldots, x_n$ are in both members and variables $x_{n+1}, \ldots, x_{n+k}$ are only in left member. We then suppose that to every such rule are associated n probabilities $q_1^l, \ldots, q_n^l$: in an application of *replacement* rule subterm $t_i$will be chosen to be rewritten with probability q_i^l. Since replacement involves at least one rewrite, we do not expect that at least one subterm is rewritten.

We have now all the ingredients.

Definition 8 (Probabilistic Rewrite System). *A probabilistic rewrite system $(\mathcal{R}, \mathcal{L}, \mathcal{W})$ is given by a labeled rewrite system $(\mathcal{R}, \mathcal{L})$, where all variables in a right member of a rule of $\mathcal{R}$ appears in the left member, with the addition of the following:*

1) a weight (positive real number) w_l for each rule $l \in \mathcal{R}$,

2) a weight w_f for each function symbol of the signature,

3) n reals $q_1^f, \ldots, q_n^f$ of $[0, 1]$ for each function symbol f of arity n,

4) n reals $q_1^l, \ldots, q_n^l$ *of* $[0,1]$ *for each rule* $l : g(x_1, \ldots, x_n, \ldots, x_{n+k}) \to r(x_1, \ldots, x_n)$ *of* $\mathcal{R}$.
The weights are assumed to be chosen such that $\sum_f w_f + \sum_l w_l = 1$.

We can then introduce the following reduction algorithm:

Definition 9. *Given some probabilistic rewrite system, Reduction is the following recursive algorithm:*
Input: a reducible term t.
Output: a term t'.
Algorithm:
1. Choose either a rule $l \in \mathcal{R}$ *or a symbol* f *of the signature, according to the probability distribution given by the weights.*
2. If a rule $l : g(x_1, \ldots, x_n, \ldots, x_{n+k}) \to r(x_1, \ldots, x_n)$ *was chosen then*
2.1 If $\nexists \sigma \in Sub$ *with* $\sigma(g) = t$ *then repeat: i.e. goto* 1.
/ From now on,* $t = g(t_1, \ldots, t_{n+k})$ *for some* $t_1, \ldots, t_{n+k}$ **/*
2.2 Choose $X_1, \ldots, X_n \in \{0,1\}$ *with probability*$(X_i = 1) = q_i^l$.
2.3 For $i = 1, \ldots, n$, *let* t'_i *be the result of the recursive call of algorithm Reduction on* t_i *when* $X_i = 1$ *and* t_i *reducible and let* $t'_i = t_i$ *otherwise.*
2.4 Return $r(t'_1, \ldots, t'_n)$.
3. If a symbol f *was chosen*
3.1 If t *is not* $f(t_1, \ldots, t_n)$ *for some* $t_1, \ldots, t_n$ *then repeat: i.e. goto 1.*
/ From now on,* $t = f(t_1, \ldots, t_n)$ *for some* $t_1, \ldots, t_n$ **/*
3.2 Choose $X_1, \ldots, X_n \in \{0,1\}$ *with probability*$(X_i = 1) = q_i^f$.
3.3 If $X_i = 0$ *for all* i *with* t_i *reducible then repeat : i.e. goto 1.*
3.4 For $i = 1, \ldots, n$, *let* t'_i *be the result of the recursive call of algorithm Reduction on* t_i *when* $X_i = 1$ *and* t_i *reducible and let* $t'_i = t_i$ *otherwise.*
3.5 Return $f(t'_1, \ldots, t'_n)$.

Remark 5. This algorithm terminates with probability 1 when given some reducible t. If given some non-reducible t it runs for ever: this is a consequence of Proposition 2.

We can then define:

Definition 10. *The* executional semantic *of a given probabilistic rewrite system* $(\mathcal{R}, \mathcal{L}, \mathcal{W})$ *is the corresponding probabilistic abstract reduction system on terms: it is defined as* $\mathcal{S}_{\mathcal{R}} = (T(\Sigma, X), \rightsquigarrow)$ *where for all* s, t, $s \rightsquigarrow p$ *is 0 if* s *is not reducible, and the probability that algorithm Reduction returns* t *on input* s *if* s *is reducible.*

When $(\mathcal{R}, \mathcal{L}, \mathcal{W})$ is a probabilistic rewrite system, call $(\mathcal{R}, \mathcal{L})$ its projection: that is, the classical rewrite system obtained by forgetting probabilities. We have from definitions:

Theorem 5. *The projection of the executional semantic of any probabilistic rewrite system is the executional semantic of its projection.*

6 Probabilistic Rewriting Logic

We now show that there is a sound and complete proof system if proof terms are explicit, i.e. if paths between terms are given.

We propose a logic that works with sequents of type $\pi : t \rightarrow_p t'$: when p is a positive real number and $t' \neq \bot$, such a sequent means that term t can evolve to term t' in the executional semantic using the path given by proof term π and that the probability of this path is p. The logic consists of three rules: *reflexivity*, *congruence*, *replacement*. Transitivity is not here because of results of Section 4.

A sequent deduced from reflexivity in classical rewriting logic does not correspond to a reduction of the rewriting reduction relation. We suggest to distinguish such a sequent from the others with the use of a new symbol replacing the probability : $\bullet$.

Reflexivity : for all reducible constant a,

$$\mathbf{Ref} : \frac{}{a : a \rightarrow_\bullet a}$$

We need a way to express that a term is non-reducible: we propose to use symbol $\bot$. We assume that rules have been added to the rewrite system so that we have the rule $\{\bot_a : a \rightarrow \bot\}$ for every non-reducible constant a. When t is a term, we denote by $R(t)$ the set of rewrite rules that can be applied at its root. In particular, we assume $R(a) = \{\bot_a : a \rightarrow \bot\}$ for every non-reducible constant a. A sentence of type $\pi : t \rightarrow_p \bot$ will mean that t is non-reducible.

Congruence : for all $f \in \Sigma_n$,

$$\mathbf{C} : \frac{\pi_1 : t_1 \rightarrow_{p_1} t'_1 \quad \cdots \quad \pi_n : t_n \rightarrow_{p_n} t'_n}{f(\pi_1, \ldots, \pi_n) : f(t_1, \ldots, t_n) \rightarrow_p f(t''_1, \ldots, t''_n)}$$

with $p = \theta_f^I$, $I = \left\{ i \in \{1, \ldots, n\} | t'_i \neq \bot \right\}$, $t''_i = \begin{cases} t'_i \text{ if } i \in I \\ t_i \text{ if } i \notin I \end{cases}$

$$\theta_f^I = \begin{cases} \bullet \text{ if } \forall i,\ p_i = \bullet \\ \left(\frac{w_f}{w_f + \sum_{R(t)} w_l}\right) \left(\frac{1}{1 - \prod_{i \in I}(1 - q_i^f)}\right) \left(\prod_{\substack{i \in I \\ p_i \neq \bullet}} q_i^f p_i\right) \left(\prod_{\substack{i \in I \\ p_i = \bullet}} (1 - q_i^f)\right) \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad otherwise \end{cases}$$

$t = f(t_1, \ldots, t_n)$.

Here, I is the set of subterms that can be reduced. The rule is valid if $I \neq \emptyset$. If $I = \emptyset$, since $f(t_1, \ldots, t_n)$ is non-reducible, the rule becomes

$$\frac{\bot_{t_1} : t_1 \rightarrow_1 \bot \quad \ldots \quad \bot_{t_n} : t_n \rightarrow_1 \bot}{\bot_{f(t_1, \ldots, t_n)} : f(t_1, \ldots, t_n) \rightarrow_1 \bot}$$

Replacement : for all $l : g(x_1, \ldots, x_{n+k}) \rightarrow d(x_1, \ldots, x_n) \in \mathcal{R}$,

$$\mathbf{Rep} : \frac{\pi_1 : t_1 \rightarrow_{p_1} t'_1 \quad \cdots \quad \pi_n : t_n \rightarrow_{p_n} t'_n}{l(\pi_1, \ldots, \pi_n, t_{n+1}, \ldots, t_{n+k}) : g(t_1, \ldots, t_n, \ldots, t_{n+k}) \rightarrow_p d(t''_1, \ldots, t''_n)}$$

with $p = \theta_l^I$, $I = \left\{ i \in \{1, \ldots, n\} | t'_i \neq \bot \right\}$, $t''_i = \begin{cases} t'_i \text{ if } i \in I \\ t_i \text{ if } i \notin I \end{cases}$
and $\theta_l^I = \begin{cases} \bullet \text{ if } \forall i,\ p_i = \bullet \\ \left(\frac{w_l}{w_f + \sum_{R(t)} w_{l'}} \right) \left(\prod_{i \in I | p_i \neq \bullet} q_i^l p_i \right) \left(\prod_{i \in I | p_i = \bullet} (1 - q_i^l) \right) \ otherwise \end{cases}$
$t = g(t_1, \ldots, t_{n+k})$.

Here the rule is correct even when $I = \emptyset$.

The previous rules distribute correctly probabilities onto rewrite rules (the proof can be found in [17]).

Proposition 3. *Let t be a reducible term. Let $S(t)$ be the set of sequents $\pi : t \rightarrow_p t'$ deductible from the rules [Reflexivity,Congruence,Replacement], and such that $p \neq \bullet$. Then $\sum_{S(t)} p = 1$.*

The main property of this proof system is given by following result (the proof, based on repeated applications of Proposition 2, can be found in [17]).

Theorem 6 (The above logic provides a sound and complete proof systems for sequents with proof terms). *Suppose probabilistic rewrite system $\mathcal{R}$ is fixed. For all $t, t' \in T(\Sigma, X)$, there is a path encoded by π between t and t' in the executional semantic of $\mathcal{R}$ of positive probability p iff $\pi : t \rightarrow_p t'$ with a positive p is provable using the previous three rules.*

7 Modeling Randomized Systems

In order to argue that our notions of probabilistic rewrite systems, executional semantic and associated logic are natural, we now show how some systems can easily be modeled. We write $l : g \rightarrow_p d$ when weight p is associated to rule $l : g \rightarrow d$.

Example 1 (Coin flipping). We use constant symbols *head* and *tail* and the following system.

$$\mathcal{R} \begin{cases} h : x \rightarrow_{1/2} head \\ t : x \rightarrow_{1/2} tail \end{cases}$$

Example 2 (Two players games). Each player has n euros at beginning. At each run, a coin is flipped. If it falls on *head* player 1 wins 1 euro from player 2. If it falls on *tail*, player 2 wins 2 euros from player 1. Game stops when one player is ruined.

Current amount of a player is encoded using constant 0 and unary function s (successor). Binary function *game* is used to group both players, and two constants W_1 and W_2 are used to mean that player 1 or 2 wins. Weight 0 is assigned to function symbol *game*. The game is modeled by the derivations starting from $game(s^n(0), s^n(0))$.

$$\mathcal{R}\begin{cases} h_1 : game(n_1, s(s(n_2))) \rightarrow_{1/2} game(s(n_1), s(n_2)) \\ h_2 : game(n_1, s(0)) \rightarrow_{1/2} W_1 \\ t_1 : game(s(s(s(n_1))), n_2) \rightarrow_{1/2} game(s(n_1), s(s(n_2))) \\ t_2 : game(s(s(0)), n_2) \rightarrow_{1/2} W_2 \end{cases}$$

Example 3 (Two players with two urns). Two players can not see one another and have each an urn. At beginning there are n balls in each urn. At each round they can choose between taking a ball in their urn or doing nothing. With probability p urns are exchanged at each run by some external person. A player with an empty urn loses.

We do as before with constant 0, W_1, W_2 and functions s and $game$. We put weight 0 to functions $game$ and s. If the probability that player i takes a ball is q_i, we set $q_1^{ech} = q_1^l = q_1$ and $q_2^{ech} = q_y^l = q_2$.

$$\mathcal{R}\begin{cases} choose : s(x) \rightarrow_1 x \\ ech : game(s(x), s(y)) \rightarrow_p game(s(y), s(x)) \\ l : game(s(x), s(y)) \rightarrow_{(1-p)} game(s(x), s(y)) \\ g_1 : game(0, s(y)) \rightarrow_1 W_1 \\ g_2 : game(s(x), 0) \rightarrow_1 W_2 \\ n : game(0, 0) \rightarrow_1 Tie \end{cases}$$

8 Related Works, Discussions

In this paper, we discussed the existence of a notion of rewriting logic in presence of probabilities. We proved that, unlike what happens for classical theory, accessibility can not be effectively axiomatized, and thus that there is no hope to get a sound and complete logic that would cover transitivity. When transitivity is avoided, in particular when proof terms are explicit and mandatory, we proved that one can define a natural notion of probabilistic rewrite system with some associated semantic, and an associated sound and complete probabilistic rewriting logic.

First-order logics have been proposed to deal with probabilities: see e.g. [3,15]. The impossibility of effective axiomatizations of several first-order logics with probabilities has been proved [1,15], but our results do not seem to follow directly.

The idea of considering rewriting rules with probabilities has already been proposed and illustrated on several examples in [8,14,20], where it is observed that the probabilities cannot be hardwired directly to rules. Paper [8] proposes to avoid the problem by considering the notion of strategy. Papers [14,20] propose a solution similar to the one adopted here considering weights instead of probabilities. Observe that this trick has similarities with classical techniques used to extract a discrete time Markov chain from a continuous one [9], and hence is sometime implicitly or explicitly used for high level modeling of continuous time Markovian systems (see e.g. [13]).

Probabilistic rewriting logic provides a high-level tool for modeling probabilistic systems. Low level models include Markov chains [9] and Markov decision processes if non-determinism is allowed [22]. Other high-level models include models based on Petri nets (cf survey [4]), on process algebra (cf survey [16]), or on automata (cf e.g. [5,13,21,23]). According to the classification [24], our proposition falls into the "generative" case. Observe that our proposition for defininig congruence and replacement is similar to (covers) what [12] proposes for the semantic of parallel composition.

The benefits of using a given approach for describing probabilistic systems, compared to another one, depend on the preferred way of describing world, but we believe that our setting is a rather natural and expressive setting, as classical rewriting logic is a rather natural and expressive setting for describing non-probabilistic reactive systems: see survey [18].

Future work includes investigating more deeply the expressive power of the logic. Considering rewriting with congruence classes may constitute a future work direction. Allowing conditional rewriting is another possibility. Another important direction seems also to understand model theory of these systems: Definition 10 reads like the notion of canonical model associated to some given probabilistic rewrite system. What is the notion of model of a given probabilistic rewrite theory? Which results of classical theory (see for e.g. the results in [18,19]) do generalize in this context?

Acknowledgments

The authors would like to thank Claude Kirchner for many helpful discussions and comments about this work.

References

1. Martín Abadi and Joseph Y. Halpern. Decidability and expressiveness for first-order logics of probability. *Information and Computation*, 112(1):1–36, July 1994.
2. Franz Baader and Tobias Nipkow. *Term Rewriting and all That*. Cambridge University Press, 1998.
3. F. Bacchus. *Representing and reasoning with probabilistic knowledge*. MIT-Press, 1990.
4. Gianfranco Balbo. Introduction to stochastic Petri nets. *Lecture Notes in Computer Science*, 2090:84, 2001.
5. Benveniste, Levy, Fabre, and Le Guernic. A calculus of stochastic systems for the specification, simulation, and hidden state estimation of mixed stochastic/non-stochastic systems. *TCS: Theoretical Computer Science*, 152, 1995.
6. A. Bergstra and J.V. Tucker. A characterisation of computable data types by means of a finite equational specification method. In Springer Verlag, editor, *Automata Languages and Programming, Seventh Colloquium*, Lecture Notes in Computer Science, pages 76–90, 1980.
7. P. Borovanský, C. Kirchner, H. Kirchner, P.-E. Moreau, and Ch. Ringeissen. An Overview of ELAN. In C. Kirchner and H. Kirchner, editors, *Second Workshop on*

Rewriting Logic and its Applications WRLA'98, volume 15 of *Electronic Notes in Theoretical Computer Science*, Pont-à-Mousson (France), 1998. Elsevier Science B. V. URL: http://www.elsevier.nl/locate/entcs/volume15.html.
8. Olivier Bournez and Claude Kirchner. Probabilistic rewrite strategies: Applications to ELAN. In Sophie Tison, editor, *Rewriting Techniques and Applications*, volume 2378 of *Lecture Notes in Computer Science*, pages 252–266. Springer-Verlag, July 22-24 2002.
9. Pierre Brémaud. *Markov Chains.* Springer, 1991.
10. C. Castro. Solving Binary CSP using Computational Systems. In J. Meseguer, editor, *Proceedings of 1st International Workshop on Rewriting Logic*, volume 4, Asilomar (CA, USA), September 1996. Electronic Notes in Theoretical Computer Science.
11. M. Clavel, F. Durán, S. Eker, J. Meseguer P. Lincoln, N. Martí-Oliet, and J.F. Quesada. Towards Maude 2.0. In *3rd International Workshop on Rewriting Logic and its Applications (WRLA'00)*, volume 36 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 2000.
12. Pedro R. D'Argenio, Holger Hermanns, and Joost-Pieter Katoen. On generative parallel composition. In *Electronic Notes In Computer Science*, volume 22, 1999.
13. L. De Alfaro. Stochastic transition systems. *Lecture Notes in Computer Science*, 1466:423, 1998.
14. Thom Frühwirth, Alexandra Di Pierro, and Herbert Wiklicky. Toward probabilistic constraint handling rules. In Slim Abdennadher and Thom Frühwirth, editors, *Proceedings of the third Workshop on Rule-Based Constraint Reasoning and Programming (RCoRP'01)*, Paphos, Cyprus, December 2001. Under the hospice of the International Conferences in Constraint Programming and Logic Programming.
15. Joseph Y. Halpern. *Discourse, Interaction, and Communication*, chapter A logical approach to reasoning about uncertainty: a tutorial, pages 141–55. Kluwer, 1998.
16. H. Hansson. *Time and Probability in Formal Design of Distributed Systems.* Series in Real-Time Safety Critical Systems. Elsevier, 1994.
17. Mathieu Hoyrup. Réécriture en présence de choix probabilistes. Master's thesis, Ecole Normale Supérieure de Lyon, 2002.
18. Narciso Martí-Oliet and José Meseguer. Rewriting logic: Roadmap and bibliography. *Theoretical Computer Science*, 285(2):121–154, 2002.
19. J. Meseguer. Conditional rewriting logic as a unified model of concurrency. *Theoretical Computer Science*, 96(1):73–155, 1992.
20. Alessandra Di Pierro and Herbert Wiklicky. An operational semantics for probabilistic concurrent constraint programming. In *Proceedings of the 1998 International Conference on Computer Languages*, pages 174–183. IEEE Computer Society Press, 1998.
21. B. Plateau and K. Atif. Stochastic automata network for modelling parallel systems. *IEEE Transactions on Software Engineering*, 17:1093–1108, 1991.
22. M.L. Puternam. *Markov Decision Processes - Discrete Stochastic Dynamic Programming.* Wiley series in probability and mathematical statistics. John Wiley & Sons, 1994.
23. R. Segala and N. Lynch. Probabilistic simulations for probabilistic processes. *Lecture Notes in Computer Science*, 836:481, 1994.
24. Rob van Glabbeek, Scott A. Smolka, Bernhard Steffen, and Chris M. N. Tofts. Reactive, generative, and stratified models of probabilistic processes. In *Proceedings, Fifth Annual IEEE Symposium on Logic in Computer Science*, pages 130–141, Philadelphia, Pennsylvania, 4–7 June 1990. IEEE Computer Society Press.