Symbolic Computation of Maximal Probabilistic Reachability*

Marta Kwiatkowska, Gethin Norman, and Jeremy Sproston

School of Computer Science, University of Birmingham,
Birmingham B15 2TT, United Kingdom.
{M.Z.Kwiatkowska,G.Norman,J.Sproston}@cs.bham.ac.uk

Abstract. We study the *maximal reachability probability* problem for infinite-state systems featuring both nondeterministic and probabilistic choice. The problem involves the computation of the maximal probability of reaching a given set of states, and underlies decision procedures for the automatic verification of probabilistic systems. We extend the framework of symbolic transition systems, which equips an infinite-state system with an algebra of symbolic operators on its state space, with a symbolic encoding of probabilistic transitions to obtain a model for an infinite-state probabilistic system called a *symbolic probabilistic system*. An exact answer to the maximal reachability probability problem for symbolic probabilistic systems is obtained algorithmically via iteration of a refined version of the classical predecessor operation, combined with intersection operations. As in the non-probabilistic case, our state space exploration algorithm is semi-decidable for infinite-state systems. We illustrate our approach with examples of probabilistic timed automata, for which previous approaches to this reachability problem were either based on unnecessarily fine subdivisions of the state space, or which obtained only an upper bound on the exact reachability probability.

1 Introduction

Many systems, such as control, real-time, and embedded systems, give rise to *infinite-state* models. For instance, embedded systems can be modelled in formalisms characterised by a finite number of control states (representing a digital controller) interacting with a finite set of real-valued variables (representing an analogue environment). Motivated by the demand for automatic verification techniques for infinite-state systems, a number of results concerning the decidability of problems such as reachability, model checking and observational equivalence have been presented: isolated results concerning models such as timed automata [3], hybrid automata [2] and data independent systems [22] have been subject to unifying theories [1,10] and, in some cases, have provided the basis of efficient analysis tools, such as the timed automata model checker Uppaal [17].

In this paper, we consider a *probabilistic* model for infinite-state systems. For examples of infinite-state systems exhibiting probabilistic behaviour, consider

* Supported in part by the EPSRC grants GR/M04617 and GR/N22960.

K.G. Larsen and M. Nielsen (Eds.): CONCUR 2001, LNCS 2154, pp. 169–183, 2001.

the real-time algorithm employed in the root contention protocol of IEEE1394 (FireWire) [20], probabilistic lossy channels [12] and open queueing networks [8]. Our system model also admits nondeterministic choice, which allows the modelling of asynchronous systems, and permits the *underspecification* of aspects of a system, including probabilistic attributes. We focus on the *maximal reachability probability* problem for probabilistic systems, concerning the computation of the maximal probability with which a given set of states is reachable. In the same way that reachability underlies the verification of temporal modalities in the non-probabilistic context, probabilistic reachability provides the foundation for probabilistic model checking of temporal modalities [6,5].

To reason about properties of infinite-state systems, an implicit, *symbolic* means to describe infinite state sets is required. The operations required on such state sets include boolean and *predecessor* operations, which together enable model checking of reachability properties by *backwards exploration* through the state space. Our first contribution concerns the extension of symbolic transition systems [10], which are infinite-state systems equipped with an algebra of such operations, with a (discrete) probabilistic transition relation. Observe that, in the context of quantitative reachability properties, it is not enough to know whether a state makes a transition to another, as encoded in the traditional predecessor operation: the probability of the transition must also be known. Our approach, which is specifically designed for the computation of maximal reachability probabilities, is to encode the transitions of a probabilistic system into a number of *types* (giving a family of *typed predecessor operations*), and the probabilistic branching of the system into a set of distributions over transition types called *distribution templates*. The resulting model, which consists of symbolic encodings of both states *and* transitions, together with an algebra of operations including the typed predecessor operations, is called a *symbolic probabilistic system*.

Our second contribution concerns the computation of the maximal reachability probability for certain classes of symbolic probabilistic systems by reduction to a finite-state problem. First, a state space exploration algorithm successively iterates typed predecessor and intersection operations, starting from the target set. The typed predecessor operations characterise the sets of states which can make a transition of a particular type to a previously generated set of states. To reason about the probabilistic branching structure of the system, we compute sets of states in which transitions of *multiple* types are enabled through intersections of state sets. If the state space exploration algorithm terminates, then a finite set of state sets is returned. Together, the transition types available in each of these state sets, and the distribution templates, allow us to construct a finite-state probabilistic system with an equal maximal reachability probability to that of the symbolic probabilistic system.

The state space analysis algorithm is closed under typed predecessor and intersection operations, and does not take *differences* between state sets; therefore, it differs from partition refinement algorithms. Our approach keeps the number of operations on the state space to a minimum, while retaining sufficient information for the computation of the maximal reachability probability.

In particular, noting that many symbolic approaches describe state sets in terms of constraints, our algorithm avoids propagating constraints arising from difference operations. To our knowledge, reasoning about reachability probabilities using a combination of predecessor and intersection operations is novel.

Related work. Approaches to infinite-state systems with discrete probability distributions include model checking methods for probabilistic lossy channel systems [12]. Two verification methods for probabilistic timed automata are presented in [15]. The first uses the "region graph" of [3] to compute *exact* reachability probabilities, but suffers from the state explosion problem (in particular, the size of the verification problem is sensitive to the magnitudes of the model's timing constraints, which is not true of our technique). The second uses forwards reachability, but, in contrast to our technique, only computes an *upper bound* on the actual maximal probability. Verification methodologies for infinite-state systems with *continuous* distributions are given in [4,7,14].

Plan of the paper. Section 2 defines symbolic probabilistic systems, and describes how they are used to represent probabilistic timed automata [15]. We present the semi-decidable algorithm to generate a finite-state representation of a symbolic probabilistic system in Section 3. Section 4 offers a critique of the analysis method, and suggests directions for future research.

2 Symbolic Probabilistic Systems

2.1 Preliminaries

A discrete probability *distribution* (*subdistribution*) over a finite set Q is a function $\mu : Q \to [0,1]$ such that $\sum_{q \in Q} \mu(q) = 1$ ($\sum_{q \in Q} \mu(q) \leq 1$). For a possibly uncountable set Q', let $\mathsf{Dist}(Q')$ ($\mathsf{SubDist}(Q')$) be the set of distributions (subdistributions) over finite subsets of Q'.

Recall that a *transition system* is a pair (S, δ) comprising a set S of states and a transition function $\delta : S \to 2^S$. A *state transition* $s \to t$ from a given state s is determined by a nondeterministic choice of target state $t \in \delta(s)$. In contrast, a (nondeterministic-) *probabilistic system* $\mathbb{S} = (S, \mathit{Steps})$ includes a probabilistic transition function $\mathit{Steps} : S \to 2^{\mathsf{Dist}(S)}$. A *probabilistic transition* $s \xrightarrow{\mu} t$ is made from a state $s \in S$ by first nondeterministically selecting a distribution μ from the set $\mathit{Steps}(s)$, and second by making a probabilistic choice of target state t according to μ, such that $\mu(t) > 0$. A *path* of a probabilistic system is a finite or infinite sequence of probabilistic transitions of the form $\omega = s_0 \xrightarrow{\mu_0} s_1 \xrightarrow{\mu_1} s_2 \cdots$. For a path ω and $i \in \mathbb{N}$, we denote by $\omega(i)$ the $(i+1)$th state of ω, and if ω is finite, $\mathit{last}(\omega)$ the last state of ω.

We now introduce *adversaries* which resolve the nondeterminism of a probabilistic system [21]. Formally, an *adversary* of $\mathbb{S}$ is a function A mapping every finite path ω to a distribution $\mu \in \mathsf{Dist}(S)$ such that $\mu \in \mathit{Steps}(\mathit{last}(\omega))$. Let $\mathit{Adv}_{\mathbb{S}}$ be the set of adversaries of $\mathbb{S}$. For any $A \in \mathit{Adv}_{\mathbb{S}}$, let Path^A_{ful} denote the

set of infinite paths associated with A. Then, in the standard way, we define the measure $Prob^A$ over $Path^A_{ful}$ [13].

The *maximal reachability probability* is the maximum probability with which a given set of states of a probabilistic system can be reached from a particular state. Formally, for the probabilistic system $\mathbb{S} = (S, Steps)$, state $s \in S$, and set $U \subseteq S$ of target states, the maximal reachability probability $ProbReach(s, U)$ of reaching U from s is defined as

$$ProbReach(s, U) \stackrel{\text{def}}{=} \sup_{A \in Adv_{\mathbb{S}}} Prob^A\{\omega \in Path^A_{ful} \mid \omega(0) = s \wedge \exists i \in \mathbb{N} \,.\, \omega(i) \in U\}.$$

The maximal reachability probability can be obtained as the solution to a linear programming problem in the case of finite probabilistic systems [6].

Computation of the maximal reachability probability allows one to verify properties of the form "with at least probability 0.99, it is possible to correctly deliver a data packet". By duality, it also applies to the validation of invariance properties such as "with probability at most 0.01, the system aborts". Furthermore, in the context of real–time systems, maximal reachability probability can be used to verify time-bounded reachability properties, also known as *soft* deadlines, such as "with probability 0.975 or greater, it is possible to deliver a message within 5 time units". For a more detailed explanation see [15].

2.2 Symbolic Probabilistic Systems: Definition and Intuition

Symbolic transition systems were introduced in [10] as (possibly infinite-state) transition systems equipped with *symbolic state algebras*, comprising a set of *symbolic states* (each element of which denotes a possibly infinite set of states), boolean, predecessor, emptiness and membership operations on symbolic states. In [10], classes of infinite-state systems for which a finitary structure can be identified by iteration of certain operations of the symbolic state algebra are defined, consequently highlighting the decidability of certain verification problems.

Symbolic probabilistic systems augment the framework of symbolic transition systems with (1) a probabilistic transition relation, (2) a symbolic encoding of probabilistic transitions, and (3) a redefined symbolic state algebra. Given the definition of probabilistic systems in the previous section, point (1) is self-explanatory. For point (2), note that information concerning probabilities is necessary for computation of maximal reachability probabilities. Let $s \to t$ be the state transition *induced* by a probabilistic transition $s \xrightarrow{\mu} t$ by abstracting the distribution μ from the transition. The symbolic representation consists of two steps: first, we encode state transitions induced by the probabilistic transitions of the system within a set of *transition types*. Second, we encode the probabilistic branching structure of the system, which is not represented in the set of transition types, by a set of *distribution templates*, which are distributions over the set of transition types. Finally, for point (3), the predecessor operation of a symbolic transition system is now replaced by a *family of predecessor operations*, each of which is defined according to the state transitions encoded by a transition type. This allow us to identify and reason about sets of states in which state

transitions of different transition types are available; in Section 3, we see that this characteristic is vital to identify a finitary structure on which the system's maximal reachability probability can be computed.

We now give the definition of symbolic probabilistic systems which generalise the symbolic transition systems of [10]. The definition of symbolic states R, extension function $\ulcorner\cdot\urcorner$, and symbolic operators $\mathsf{And}, \mathsf{Diff}, \mathsf{Empty}$ and Member agree with those given for symbolic transition systems, with the only difference being the typed predecessor operations. Conditions 1(a–c) have been added to represent probabilistic systems in such a way as to preserve maximal reachability probabilities, and are explained after the definition. In other contexts, different choices of symbolic representation and operations may be appropriate.

Definition of symbolic probabilistic systems. *A* symbolic probabilistic system $\mathbb{P} = (S, \mathit{Steps}, R, \ulcorner\cdot\urcorner, \mathcal{T}ra, \mathsf{D})$ *comprises: a probabilistic system* (S, Steps)*; a set of symbolic states* R*; an extension function* $\ulcorner\cdot\urcorner : R \to 2^S$*; a set of transition types* $\mathcal{T}ra$*, and, associated with each* $a \in \mathcal{T}ra$*, a transition function* $\delta_a : S \to 2^S$*; and a set of distribution templates* $\mathsf{D} \subseteq \mathsf{Dist}(\mathcal{T}ra)$*, such that the following conditions are satisfied.*

1. *For all states* $s \in S$*, let* $\mathcal{T}ra(s) \subseteq \mathcal{T}ra$ *be such that for any* $a \in \mathcal{T}ra$*:* $a \in \mathcal{T}ra(s)$ *if and only if* $\delta_a(s) \neq \emptyset$*. Then, for all* $t \in S$*:*
   a) *if* $a \in \mathcal{T}ra$ *and* $t \in \delta_a(s)$*, then there exists* $\mu \in \mathit{Steps}(s)$ *such that* $\mu(t) > 0$*;*
   b) *if* $\mu \in \mathit{Steps}(s)$*, then there exists* $\nu \in \mathsf{D}$ *and a vector of states* $\langle t_a \rangle_{a \in \mathcal{T}ra(s)} \in \prod_{a \in \mathcal{T}ra(s)} \delta_a(s)$ *such that:*

$$\sum_{a \in \mathcal{T}ra(s) \wedge t = t_a} \nu(a) = \mu(t);$$

   c) *if* $\nu \in \mathsf{D}$ *and* $\langle t_a \rangle_{a \in \mathcal{T}ra(s)}$ *is a vector of states in* $\prod_{a \in \mathcal{T}ra(s)} \delta_a(s)$*, then there exists* $\mu \in \mathit{Steps}(s)$ *such that:*

$$\mu(t) \geq \sum_{a \in \mathcal{T}ra(s) \wedge t = t_a} \nu(a).$$

2. *There exists a family of computable functions* $\{\mathsf{pre}_a\}_{a \in \mathcal{T}ra}$ *of the form* $\mathsf{pre}_a : R \to R$*, such that, for all* $a \in \mathcal{T}ra$ *and* $\sigma \in R$*:*

$$\ulcorner \mathsf{pre}_a(\sigma) \urcorner = \{ s \in S \mid \exists t \in \delta_a(s) \,.\, t \in \ulcorner \sigma \urcorner \} \,.$$

3. *There is a computable function* $\mathsf{And} : R \times R \to R$ *such that* $\ulcorner \mathsf{And}(\sigma, \tau) \urcorner = \ulcorner \sigma \urcorner \cap \ulcorner \tau \urcorner$ *for each pair of symbolic states* $\sigma, \tau \in R$*.*
4. *There is a computable function* $\mathsf{Diff} : R \times R \to R$ *such that* $\ulcorner \mathsf{Diff}(\sigma, \tau) \urcorner = \ulcorner \sigma \urcorner \setminus \ulcorner \tau \urcorner$ *for each pair of symbolic states* $\sigma, \tau \in R$*.*
5. *There is a computable function* $\mathsf{Empty} : R \to \mathbb{B}$ *such that* $\mathsf{Empty}(\sigma)$ *if and only if* $\ulcorner \sigma \urcorner = \emptyset$ *for each symbolic state* $\sigma \in R$*.*
6. *There is a computable function* $\mathsf{Member} : S \times R \to \mathbb{B}$ *such that* $\mathsf{Member}(s, \sigma)$ *if and only if* $s \in \ulcorner \sigma \urcorner$ *for each state* $s \in S$ *and symbolic state* $\sigma \in R$*.*

We proceed to describe transition types and distribution templates in greater depth.

Transition types. Recall that a transition type encodes a set of state transitions of a symbolic probabilistic system. Hence, for each transition type $a \in \mathcal{T}ra$ there is a transition relation $\delta_a : S \to 2^S$ encoding all of the state transitions of type a. This grouping is *not* necessarily a partition of the state transitions and a given state transition may correspond to more than one type. It follows from the lemma below that every probabilistic transition is represented by a state transition encoded in some transition type, and vice versa.

Lemma 1. *Let $\mathbb{P} = (S, Steps, R, \ulcorner \cdot \urcorner, \mathcal{T}ra, \mathsf{D})$ be a symbolic probabilistic system. For any $s, t \in S$: $\mu(t) > 0$ for some $\mu \in Steps(s)$ if and only if $t \in \delta_a(s)$ for some $a \in \mathcal{T}ra$.*

Distribution templates. Recall that we use the set of distribution templates to encode the actual probabilities featured in the system. Point 1(b) requires that the probabilistic branching structure of the system is represented in the distribution templates. Conversely, condition 1(c) expresses the fact that, in all states, for any transition encoded by a distribution template and transition type, there exists a system transition which assigns an equal or greater probability to all target states. This implies that there may be combinations of distribution templates and transition types which do not correspond to actual probabilistic transitions of the system. However, condition 1(c) together with 1(b) ensures that our model is nevertheless sufficient for the computation of the maximal reachability probability.

Example 1. Consider a system in which the state space takes the form of valuations of a single real-valued variable x. In state $s \in \mathbb{R}$, the variable x can be reset nondeterministically in the intervals (1,3) and (2,4), each with probability 0.5. Consider representing the system as a symbolic probabilistic system, where the set of symbolic states is the set of integer-bounded intervals of $\mathbb{R}$. The above behaviour can then be encoded by transition types a and b, such that $\delta_a(s) = (1, 3)$ and $\delta_b(s) = (2, 4)$, and the distribution template $\nu \in \mathsf{Dist}(\{a, b\})$ given by $\nu(a) = \nu(b) = 0.5$. Now, for any $s' \in (2, 3)$ there exists a distribution $\mu_{s'} \in Steps(s)$ which corresponds to moving from s and resetting x to s' with probability 1. For any such $\mu_{s'}$, the corresponding vector $\langle t_a, t_b \rangle$, described in point 1(b), is given by $t_a = t_b = s'$.

Finiteness of transition types and templates. Observe that the sets of transition types and distribution templates associated with a symbolic probabilistic system may be infinite. However, in Section 3, we restrict the analysis techniques to systems with finite sets of distribution templates and transition types. This assumption implies that the analysis method is appropriate for classes of infinite-state system exhibiting finite *regularity* in probabilistic transitions. For example, the probabilistic lossy channels of [12] cannot be modelled using a finite set of

distribution templates, because the probability of message loss varies with the quantity of data in the unbounded buffer.

2.3 Example: Probabilistic Timed Automata

In this section, we show that probabilistic timed automata [15] can be represented as symbolic probabilistic systems. We assume familiarity with the classical, non-probabilistic timed automaton model [3,11]. For an in-depth introduction to probabilistic timed automata, refer to [15].

Let $\mathcal{X}$ be a set of real-valued variables called *clocks*. Let $Zones(\mathcal{X})$ be the set of *zones* over $\mathcal{X}$, which are conjunctions of atomic constraints of the form $x \sim c$ and $x - y \sim c$, for $x, y \in \mathcal{X}$, $\sim \in \{<, \leq, \geq, >\}$, and $c \in \mathbb{N}$. A point $v \in \mathbb{R}^{|\mathcal{X}|}$ is referred to as a *clock valuation*. The clock valuation v *satisfies* the zone ζ, written $v \models \zeta$, if and only if ζ resolves to true after substituting each clock $x \in \mathcal{X}$ with the corresponding clock value v_x from v.

A *probabilistic timed automaton* is a tuple $\mathsf{PTA} = (L, \mathcal{X}, inv, prob, \langle g_l \rangle_{l \in L})$, where: L is a finite set of *locations*; the function $inv : L \to Zones(\mathcal{X})$ is the *invariant condition*; the function $prob : L \to 2^{\mathsf{Dist}(L \times 2^{\mathcal{X}})}$ is the *probabilistic edge relation* such that $prob(l)$ is finite for all $l \in L$; and, for each $l \in L$, the function $g_l : prob(l) \to Zones(\mathcal{X})$ is the *enabling condition* for l. A state of a probabilistic timed automaton PTA is a pair (l, v) where $l \in L$ and $v \in \mathbb{R}^{|\mathcal{X}|}$. If the current state is (l, v), there is a nondeterministic choice of either letting *time pass* while satisfying the invariant condition $inv(l)$, or making a *discrete* transition according to any distribution in $prob(l)$ whose enabling condition $g_l(p)$ is satisfied. If the distribution $p \in prob(l)$ is chosen, then the probability of moving to the location l' and resetting all of the clocks in the set X to 0 is given by $p(l', X)$.

Example 2. Consider the probabilistic timed automaton PTA modelling a simple probabilistic communication protocol given in Figure 1. The nodes represent the locations: II (sender, receiver both idle); DI (sender has data, receiver idle); SI (sender sent data, receiver idle); and SR (sender sent data, receiver received). As soon as data has been received by the sender, the protocol moves to the location DI with probability 1. In DI, after between 1 and 2 time units, the protocol makes a transition either to SR with probability 0.9 (data received), or to SI with probability 0.1 (data lost). In SI, the protocol will attempt to resend the data after 2 to 3 time units, which again can be lost, this time with probability 0.05.

Before we represent a probabilistic timed automaton as a symbolic probabilistic system, we introduce the following definitions. Let $v \in \mathbb{R}^{|\mathcal{X}|}$ be a clock valuation: for any real $\eta \geq 0$, the clock valuation $v + \eta$ is obtained from v by adding η to the values of each of the clocks; and, for any $X \subseteq \mathcal{X}$, the clock valuation $v[X := 0]$ is obtained from v by resetting all of the clocks in X to 0. Now, for zone ζ and $\eta \geq 0$, let $\zeta + \eta$, be the expression in which each clock $x \in \mathcal{X}$ is replaced syntactically by $x + \eta$ in ζ, and let $[X := 0]\zeta$ be the expression in which each clock $x \in X$ is replaced syntactically by 0 in ζ. The set of *edges* of PTA, denoted

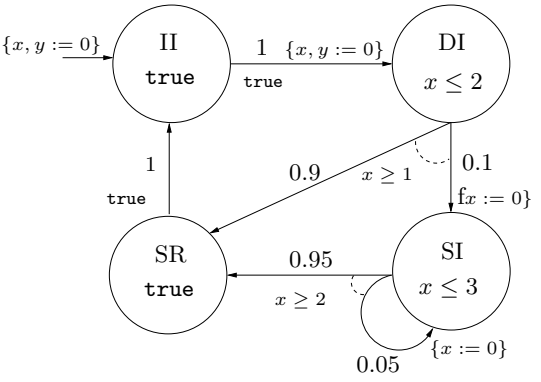

Fig. 1. A probabilistic timed automaton modelling a probabilistic protocol.

by $E_{\mathsf{PTA}} \subseteq L^2 \times 2^{\mathcal{X}} \times \mathit{Zones}(\mathcal{X})$, is defined such that $(l, l', X, \zeta) \in E_{\mathsf{PTA}}$ if and only if there exists $p \in \mathit{prob}(l)$ such that $g_l(p) = \zeta$ and $p(l', X) > 0$.

A probabilistic timed automaton $\mathsf{PTA} = (L, \mathcal{X}, \mathit{inv}, \mathit{prob}, \langle g_l \rangle_{l \in L})$ defines a symbolic probabilistic system $\mathbb{P} = (S, \mathit{Steps}, R, \ulcorner \cdot \urcorner, \mathcal{T}\mathit{ra}, \mathsf{D})$, where:

- (S, Steps) is the infinite-state probabilistic system obtained as a semantical model for probabilistic timed automata in the standard manner [15].
- The set R of symbolic states is given by $L \times \mathit{Zones}(\mathcal{X})$. The extension function $\ulcorner \cdot \urcorner$ is given by $\ulcorner (l, \zeta) \urcorner = \{(l, v) \in S \mid v \models \zeta\}$ for each $(l, \zeta) \in R$.
- The set of transition types $\mathcal{T}\mathit{ra}$ is the set of edges E_{PTA} plus the special type *time* such that, for any edge $(l', l'', X, \zeta') \in E_{\mathsf{PTA}}$, and state $(l, v) \in S$:

$$\delta_{\mathit{time}}(l, v) = \{(l, v + \eta) \mid \eta \geq 0 \wedge \forall\, 0 \leq \eta' \leq \eta \,.\, v + \eta' \models \mathit{inv}(l)\}$$

$$\delta_{(l', l'', X, \zeta)}(l, v) = \begin{cases} \{(l'', v[X := 0])\} & \text{if } l = l' \text{ and } v \models \zeta \\ \emptyset & \text{otherwise.} \end{cases}$$

- The set of distribution templates D is such that $\nu \in \mathsf{D}$ if and only if either:
  1. $\nu(\mathit{time}) = 1$, or
  2. there exists a location $l \in L$ and distribution $p \in \mathit{prob}(l)$ such that, for all transition types $a \in \mathcal{T}\mathit{ra}$:

$$\nu(a) = \begin{cases} p(l', X) & \text{if } a = (l, l', X, g_l(p)) \text{ for some } l' \in L \text{ and } X \subseteq \mathcal{X} \\ 0 & \text{otherwise.} \end{cases}$$

Given $(l, v) \in S$, the set $\delta_{\mathit{time}}(l, v)$ represents the set of states to which a time passage transition can be made, whereas $\delta_{(l', l'', X, \zeta)}(l, v)$ represents the unique state which is reached after crossing the edge denoted by (l', l'', X, ζ), provided that it is available, and the empty symbolic state otherwise. As time passage transitions are always made with probability 1, there exists a distribution template $\nu_{\mathit{time}} \in \mathsf{D}$, such that $\nu_{\mathit{time}}(\mathit{time}) = 1$; each of the other distribution templates in D is derived from a unique distribution of the probabilistic timed automaton.

For any symbolic state $(l, \zeta) \in R$, and any edge $(l', l'', X, \zeta') \in E_{\mathsf{PTA}}$, the typed predecessor operations are defined by:

$$\mathsf{pre}_{time}(l, \zeta) = (l, (\exists \eta \geq 0 \,.\, \zeta + \eta \wedge \forall\, 0 \leq \eta' \leq \eta \,.\, inv(l) + \eta'))$$
$$\mathsf{pre}_{(l', l'', X, \zeta')}(l, \zeta) = \begin{cases} (l', (\zeta' \wedge inv(l') \wedge [X := 0](\zeta \wedge inv(l)))) & \text{if } l = l'' \\ (l, \mathtt{false}) & \text{otherwise.} \end{cases}$$

Observe that these operations are defined in terms of pairs of locations and constraints on clocks. Note that by classical timed automata theory [11], for each $a \in \mathcal{T}ra$, the function pre_a is well defined and computable. Boolean operations, membership and emptiness are also well defined and computable for R. Both of the sets $\mathcal{T}ra$ and D are finite, which follows from the finiteness of L and $prob(l)$ for each $l \in L$.

Points 1(b) and 1(c) of the definition of symbolic probabilistic systems apply to probabilistic timed automata for the following reasons. As explained above, the distribution template ν_{time} encodes time passage transitions of the probabilistic system $(S, Steps)$ and conditions 1(b) and 1(c) follow trivially. The other transitions of PTA consist of choices of enabled distributions. Recall that edges of the probabilistic timed automaton are transition types. First consider condition 1(b): for any $l \in L$ and $p \in prob(l)$, there exists a distribution template $\nu \in \mathsf{D}$ assigning the same probability to the edges induced by p. Then, a probabilistic transition of $(S, Steps)$ corresponding to p will be encoded by this ν. For condition 1(c), recall that each $\nu \in \mathsf{D} \setminus \{\nu_{time}\}$ is derived from a particular $p \in prob(l)$ for some $l \in L$. Then, for the state $(l', v) \in S$, either $l' = l$ and $v \models g_l(p)$, and condition 1(c) follows as in the case of 1(b), or ν assigns probability 0 to all types in $\mathcal{T}ra(s)$, and hence any distribution available in this state will ensure the satisfaction of 1(c).

The translation method can be adapted to classes of *probabilistic hybrid automata* [18,19], which are hybrid automata [2] augmented with a probabilistic edge relation similar to that featured in the definition of probabilistic timed automata, given an appropriate set of symbolic states and algebra of operations. For example, a translation for *probabilistic linear hybrid automata* is immediate, given the above translation and the translation from non-probabilistic linear hybrid automata to symbolic transition systems of [10].

3 Maximal Reachability Probability Algorithm

We now present a *semi-decidable* algorithm (semi-algorithm) solving the maximal reachability probability problem for symbolic probabilistic systems. As mentioned in the previous section, we restrict attention to those symbolic probabilistic systems with *finite* sets of transition types and distribution templates. Note that, even for symbolic probabilistic systems within this class, the algorithm is not guaranteed to terminate.

Let $\mathbb{P} = (S, Steps, R, \ulcorner \cdot \urcorner, \mathcal{T}ra, \mathsf{D})$ be a symbolic probabilistic system such that the sets $\mathcal{T}ra$ and D are finite, and let $F \subseteq R$ be the target set of symbolic states which for which the maximal reachability probability is to be computed.

Symbolic semi-algorithm ProbReach
input: $(R, \mathcal{T}ra, \{\mathsf{pre}_a\}_{a\in\mathcal{T}ra}, \mathsf{And}, \mathsf{Diff}, \mathsf{Empty}, \mathsf{Member})$
target set $F \subseteq R$
$T_0 := F$;
$E := \emptyset$;
for $i = 0, 1, 2, \ldots$ **do**
$T_{i+1} := T_i$
for all $a \in \mathcal{T}ra \wedge \sigma \in T_i$ **do**
$T_{i+1} := \mathsf{pre}_a(\sigma) \cup T_{i+1}$
$T_{i+1} := \{\mathsf{And}(\mathsf{pre}_a(\sigma), \tau) \mid \tau \in T_{i+1}\} \cup T_{i+1}$ $(*)$
$E := \{(\mathsf{pre}_a(\sigma), a, \sigma)\} \cup E$
end for all
until $\ulcorner T_{i+1} \urcorner \subseteq \ulcorner T_i \urcorner$
$(T, E) := \mathsf{ExtendEdges}(T_i, E)$
return (T, E)

Procedure ExtendEdges
input: graph (T, E)
for all $\sigma \in T \wedge (\sigma', a, \tau) \in E$ **do**
if $\ulcorner \sigma \urcorner \subseteq \ulcorner \sigma' \urcorner$ **then**
$E := \{(\sigma, a, \tau)\} \cup E$
end if
end for all
return (T, E)

Fig. 2. Backwards exploration using predecessor and intersection operations

Our first task is to generate a finite graph (T, E), where $T \subseteq R$ and $E \subseteq T \times \mathcal{T}ra \times T$. The nodes of the graph (T, E) will subsequently form the states of a finite-state probabilistic system, and the edges will be used to define the required probabilistic transitions. The symbolic semi-algorithm ProbReach which generates the graph (T, E) is shown in Figure 2.

The algorithm ProbReach proceeds by successive iteration of predecessor and intersection operations. For each $i \in \mathbb{N}$ and for all currently generated symbolic states in the set T_i, the algorithm constructs the set T_{i+1} of symbolic states by adding to T_i the typed predecessors of the symbolic states in T_i, and the intersections of these predecessors with symbolic states in T_i. Furthermore, the edge relation E is expanded to relate the existing symbolic states to their newly generated typed predecessors. For any two symbolic states $\sigma, \tau \in R$, the test $\ulcorner \sigma \urcorner \subseteq \ulcorner \tau \urcorner$ is decided by checking whether $\mathsf{Empty}(\mathsf{Diff}(\sigma, \tau))$ holds. Then the termination test $\ulcorner T_{i+1} \urcorner \subseteq \ulcorner T_i \urcorner$ denotes the test $\{\ulcorner \sigma \urcorner \mid \sigma \in T_{i+1}\} \subseteq \{\ulcorner \sigma \urcorner \mid \sigma \in T_i\}$, which is decided as follows: for each $\sigma \in T_{i+1}$, check that there exists $\tau \in T_i$ such that both $\ulcorner \sigma \urcorner \subseteq \ulcorner \tau \urcorner$ and $\ulcorner \tau \urcorner \subseteq \ulcorner \sigma \urcorner$ [10].

If the outer **for** loop of the symbolic semi-algorithm ProbReach terminates, then we call the procedure ExtendEdges on the graph (T, E). Intuitively, for a

particular edge $(\sigma, a, \tau) \in E$, the procedure constructs edges with the transition type a and target symbolic state τ for all subset symbolic states of σ in T. Finally, observe that the set T is closed under typed predecessor and intersection operations. However, in a practical implementation of ProbReach, symbolic states encoding empty sets of states, and their associated edges, do not need to be added to the sets T and E respectively.

Remark 1 (termination of ProbReach*).* Termination of ProbReach is reliant on the termination of the outer **for** loop, because, if this terminates, T and E are finite, and hence the procedure ExtendEdges will also terminate. Observe that the inner **for** loop of the algorithm will not terminate if the set $\mathcal{T}ra$ is not finite. Now let $\preceq$ be a binary relation on the state space S of $\mathbb{P}$ such that $s \preceq t$ implies, for all $a \in \mathcal{T}ra$ and $s' \in \delta_a(s)$, there exists $t' \in \delta_a(t)$ such that $s' \preceq t'$. We call such a relation a *typed simulation*. Let $\approx$ be an equivalence relation on the state space S such that $s \approx t$ if there exists typed simulations $\preceq, \preceq'$ such that $s \preceq t$ and $t \preceq' s$. We call a relation such as $\approx$ a *typed mutual simulation*, and say $\approx$ has *finite index* if there are finitely many equivalence classes of $\approx$.

The arguments of [10] are adapted to show that ProbReach will terminate for any symbolic probabilistic system for which there exists a typed mutual simulation $\approx$ with finite index, given that the target set F is a set of equivalence classes of $\approx$. That is, we show that for all $\sigma \in T$, the set $\ulcorner\sigma\urcorner$ is a union of equivalence classes of $\approx$. This is achieved by proving by induction on $i \in \mathbb{N}$ that, for all $s, t \in S$ such that $s \preceq t$ for some typed simulation $\preceq$, if $\sigma \in T_i$ and $s \in \ulcorner\sigma\urcorner$, then $t \in \ulcorner\sigma\urcorner$. Probabilistic timed automata and probabilistic rectangular automata with two continuous variables exhibit such a relation, as indicated by [3] and [9] respectively.

If the semi-algorithm ProbReach terminates, the graph (T, E) is such that each symbolic state $\sigma \in T$ encodes a set of states of the symbolic probabilistic system $\mathbb{P}$, all of which can reach the target set F with positive probability. The following lemma asserts that the states encoded by the source of an edge in E are encoded by the appropriately typed predecessor of the edge's target symbolic state.

Lemma 2. *Let $\mathbb{P} = (S, Steps, R, \ulcorner\cdot\urcorner, \mathcal{T}ra, \mathsf{D})$ be a symbolic probabilistic system and let (T, E) be the graph constructed using the semi-algorithm* ProbReach*. For any transition type $a \in \mathcal{T}ra$, if $(\sigma, a, \tau) \in E$, then $\ulcorner\sigma\urcorner \subseteq \ulcorner\mathsf{pre}_a(\tau)\urcorner$.*

Next, we construct a finite-state probabilistic system, the states of which are the symbolic states generated by ProbReach, and the transitions of which are induced by the set of edges E and the finite set of distribution templates D. That is, we lift the identification of state transitions encoded in E to probabilistic transitions. We achieve this by grouping edges which have the *same* source symbolic state and which correspond to *different* transition types. Then a probabilistic transition of $\mathbb{Q}$ is derived from a distribution template by using the association between target symbolic states and the transition types of the edges in the identified group. Formally, we define a *sub-probabilistic system* $\mathbb{Q} = (T, Steps_{\mathbb{Q}})$, where $Steps_{\mathbb{Q}} : T \to 2^{\mathsf{SubDist}(T)}$ is the sub-probabilistic transition relation $Steps_{\mathbb{Q}}$

constructed as follows. For any symbolic state $\sigma \in T$, let $\pi \in \mathit{Steps}_{\mathbb{Q}}(\sigma)$ if and only if there exists a subset of edges $E_\pi \subseteq E$ and a distribution template $\nu \in \mathsf{D}$ such that:

1. if $(\sigma', a, \tau') \in E_\pi$, then $\sigma' = \sigma$;
2. if $(\sigma, a, \tau), (\sigma, a', \tau') \in E_\pi$ are distinct edges, then $a \neq a'$;
3. the set E_π is maximal;
4. for all symbolic states $\tau \in T$:

$$\pi(\tau) = \sum_{a \in \mathcal{T}ra \wedge (\sigma, a, \tau) \in E_\pi} \nu(a).$$

For any symbolic state $\sigma \in T$, any $\pi \in \mathit{Steps}_{\mathbb{Q}}(\sigma)$ may be a *sub-distribution*, as it is not necessarily the case that all of the transition types assigned positive probability by the distribution template associated with π are featured in the edges in E_π: some transition types may lead to states which cannot reach the target F. Note that the finiteness of the set D of distribution templates is required for the construction of the sub-probabilistic system $\mathbb{Q}$ to be feasible.

We now state the formal correctness of our algorithm (the proof can be found in [16]); that is, for any state $s \in S$ and symbolic state $\sigma \in T$ such that $s \in \ulcorner \sigma \urcorner$, the maximal reachability probability of $\mathbb{P}$ reaching the set $\ulcorner F \urcorner$ of states from the state s equals that of $\mathbb{Q}$ reaching the set F from σ.

Theorem 1. *If $\mathbb{Q} = (T, \mathit{Steps}_{\mathbb{Q}})$ is the sub-probabilistic system constructed using the algorithm* ProbReach*, with input given by the symbolic probabilistic system $\mathbb{P} = (S, \mathit{Steps}_{\mathbb{P}}, R, \ulcorner \cdot \urcorner, \mathcal{T}ra, \mathsf{D})$ and target set $F \subseteq R$, then for any state $s \in S$:*

$$\mathit{ProbReach}(s, \ulcorner F \urcorner) = \max_{\sigma \in T \wedge s \in \ulcorner \sigma \urcorner} \mathit{ProbReach}(\sigma, F) \ .$$

Recall from Section 2.1 that the maximal reachability probability for finite probabilistic systems can be computed using established methods [6].

We now describe a method which removes information from $\mathbb{Q}$ which is redundant to the computation of the maximal reachability probability.

Remark 2 (redundant conjunction operations). The purpose of the conjunction operation And in the algorithm ProbReach is to generate symbolic states for which multiple transition types are available. However, taking the conjunction of predecessors of transition types which are never *both* assigned positive probability by any distribution template does not add information concerning the probabilistic branching of the symbolic probabilistic system to $\mathbb{Q}$, and hence does not affect in the computation of the maximal reachability probability. To avoid taking such redundant conjunctions of state sets, we can replace the line marked $(*)$ in the semi-algorithm ProbReach with the following:

for all $\nu \in \mathsf{D}$ such that $\nu(a) > 0$ **do**
$T_{i+1} := \{\mathsf{And}(\mathsf{pre}_a(\sigma), \sigma') \mid (\sigma', b, \tau) \in \mathsf{relevant}(a, \nu, E)\} \cup T_{i+1}$
$E := \{(\mathsf{And}(\mathsf{pre}_a(\sigma), \sigma'), c, \tau) \mid (\sigma', b, \tau) \in \mathsf{relevant}(a, \nu, E) \ \wedge c \in \{a, b\}\} \cup E$
end for all

where $(\sigma, b, \tau) \in \mathsf{relevant}(a, \nu, E)$ if and only if $b \neq a$, $\nu(b) > 0$ and $(\sigma, b, \tau) \in E$.

Example 2 (continued). Say that we want to find the maximal probability of the probabilistic timed automaton of Figure 1 reaching the location SR, corresponding to correct receipt of a message, within 4 time units of the data arriving at the sender. Given that the target set F equals $\{(\mathrm{SR}, y < 4)\}$, application of ProbReach on the symbolic probabilistic system of this automaton results in the construction of the sub-probabilistic system in Figure 3. As suggested above, we do not consider symbolic states corresponding to empty sets of states. By classical probabilistic reachability analysis on this system, the maximal probability of reaching SR within 4 time units of the data arriving at the sender is 0.995.

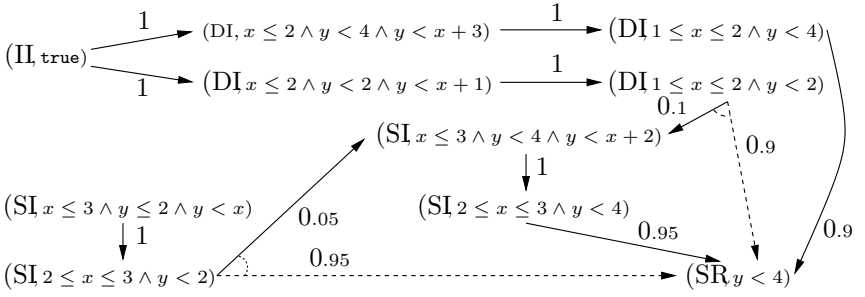

Fig. 3. The probabilistic system generated by ProbReach for the PTA in Figure 1.

The symbolic states and the solid edges are generated by the main loop of the algorithm ProbReach, while the dashed lines are added by the procedure ExtendEdges. For example, there is a solid edge corresponding to a particular transition type from the symbolic state $(\mathrm{DI}, 1 \leq x \leq 2 \wedge y < 4)$ to the symbolic state $(\mathrm{SR}, y < 4)$. Then, as $(\mathrm{DI}, 1 \leq x \leq 2 \wedge y < 2)$ is a subset of $(\mathrm{DI}, 1 \leq x \leq 2 \wedge y < 4)$, the procedure ExtendEdges adds an extra edge from $(\mathrm{DI}, 1 \leq x \leq 2 \wedge y < 2)$ to $(\mathrm{SR}, y < 4)$ of the same transition type. On inspection of Figure 1, and by the definition of the translation method for probabilistic timed automata to symbolic probabilistic systems, there exists a distribution template which assigns probability 0.9 and 0.1 to the transition types of the edges from $(\mathrm{DI}, 1 \leq x \leq 2 \wedge y < 2)$ to $(\mathrm{SR}, y < 4)$, and to $(\mathrm{SI}, x \leq 3 \wedge y < 4 \wedge y < x + 2)$, respectively. Therefore, the distribution associated with the symbolic state $(\mathrm{DI}, 1 \leq x \leq 2 \wedge y < 2)$ shown in Figure 3 is constructed.

4 Conclusions

Recall that the state space exploration algorithm presented in Section 3 iterates predecessor and intersection operations; unlike a partition refinement algorithm, it does not perform *difference* operations. Our motivation is that state sets of many infinite-state systems, including timed and hybrid automata, are described by constraints. If difference operations are used when intersecting state sets, then constraints representing the states within the intersection, *and* the negation of these constraints, are represented, rather than just the former.

Note that the algorithm could be applied only to the portion of the state space which is reachable from initial states, thereby avoiding analysis of unreachable states. Furthermore, the practical implementation of our approach can be tailored to the model in question. For probabilistic timed automata, state sets and transitions resulting from time transitions do not need to be represented; instead, typed predecessors are redefined to reflect both time passage and edge transitions. Observe that the state space exploration technique presented here will only generate *convex* zones; non-convex zones are notoriously expensive in terms of space.

Our method extends to enable the verification of symbolic probabilistic systems against the existential fragments of probabilistic temporal logics such as PCTL [6,5], though at a cost of adding union and difference operations in order to cater for disjunction and negation. However, to enable the verification of full PCTL a solution to the *minimum* reachability probability problem is required.

Finally, we conjecture that the methods presented in this paper have significance for the verification of probabilistic hybrid and parameterised systems.

References

1. P. A. Abdulla, K. Čerāns, B. Jonsson, and Y.-K. Tsay. General decidability theorems for infinite-state systems. In *Proc. LICS'96*, pages 313–321. IEEE Computer Society Press, 1996.
2. R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138(1):3–34, 1995.
3. R. Alur and D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.
4. C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen. Model checking continuous-time Markov chains by transient analysis. In *Proc. CAV 2000*, volume 1855 of *LNCS*, pages 358–372. Springer, 2000.
5. C. Baier and M. Z. Kwiatkowska. Model checking for a probabilistic branching time logic with fairness. *Distributed Computing*, 11(3):125–155, 1998.
6. A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In *Proc. FSTTCS'95*, volume 1026 of *LNCS*, pages 499–513. Springer, 1995.
7. J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden. Approximating labeled Markov processes. In *Proc. LICS 2000*, pages 95–106. IEEE Computer Society Press, 2000.
8. B. Haverkort. *Performance of Computer Communication Systems: A Model-Based Approach*. John Wiley and Sons, 1998.
9. M. R. Henzinger, T. A. Henzinger, and P. W. Kopke. Computing simulations on finite and infinite graphs. In *Proc. FOCS'95*, pages 453–462. IEEE Computer Society Press, 1995.
10. T. A. Henzinger, R. Majumdar, and J.-F. Raskin. A classification of symbolic transition systems, 2001. Preliminary version appeared in *Proc. STACS 2000*, volume 1770 of LNCS, pages 13–34, Springer, 2000.
11. T. A. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine. Symbolic model checking for real-time systems. *Information and Computation*, 111(2):193–244, 1994.

12. P. Iyer and M. Narasimha. Probabilistic lossy channel systems. In *Proc. TAPSOFT'97*, volume 1214 of *LNCS*, pages 667–681. Springer, 1997.
13. J. G. Kemeny, J. L. Snell, and A. W. Knapp. *Denumerable Markov Chains*. Graduate Texts in Mathematics. Springer, 2nd edition, 1976.
14. M. Z. Kwiatkowska, G. Norman, R. Segala, and J. Sproston. Verifying quantitative properties of continuous probabilistic timed automata. In *Proc. CONCUR 2000*, volume 1877 of *LNCS*, pages 123–137. Springer, 2000.
15. M. Z. Kwiatkowska, G. Norman, R. Segala, and J. Sproston. Automatic verification of real-time systems with discrete probability distributions. *Theoretical Computer Science*, 2001. Special issue on ARTS'99. To appear.
16. M. Z. Kwiatkowska, G. Norman, and J. Sproston. Symbolic computation of maximal probabilistic reachability. Technical Report CSR-01-5, School of Computer Science, University of Birmingham, 2001.
17. P. Pettersson and K. G. Larsen. UPPAAL2k. *Bulletin of the European Association for Theoretical Computer Science*, 70:40–44, 2000.
18. J. Sproston. Decidable model checking of probabilistic hybrid automata. In *Proc. FTRTFT 2000*, volume 1926 of *LNCS*, pages 31–45. Springer, 2000.
19. J. Sproston. *Model Checking of Probabilistic Timed and Hybrid Systems*. PhD thesis, University of Birmingham, 2001.
20. M. I. A. Stoelinga and F. Vaandrager. Root contention in IEEE1394. In *Proc. ARTS'99*, volume 1601 of *LNCS*, pages 53–74. Springer, 1999.
21. M. Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proc. FOCS'85*, pages 327–338. IEEE Computer Society Press, 1985.
22. P. Wolper. Expressing interesting properties of programs in propositional temporal logic. In *Proc. POPL'86*, pages 184–193. ACM, 1986.