Symbolic Computation of Minimal Probabilistic Reachability*

Marta Kwiatkowska[1], Gethin Norman[1] and Jeremy Sproston[2]
[1] School of Computer Science, University of Birmingham, Edgbaston, Birmingham B15 2TT, United Kingdom
[2] Dipartimento di Informatica, Università di Torino, 10149 Torino, Italy

January 13, 2003

Abstract

This paper continues our study of the verification problem for infinite-state systems featuring both nondeterministic and probabilistic choice. In an earlier paper we defined symbolic probabilistic systems, an extension of the framework of symbolic transition systems due to Henzinger et. al., and considered the problem of deciding the maximal probability of reaching a set of target states. A symbolic probabilistic system is an infinite-state system equipped with an algebra of symbolic operators on its state space, additionally extended with a symbolic encoding of probabilistic transitions to obtain a model for infinite-state probabilistic systems. In this paper we generalise the notion of symbolic probabilistic systems and consider the minimal reachability problem, that is, the problem of computing the minimal probability of reaching a given set of target states. An exact answer to this problem is obtained algorithmically via iteration of a refined version of the classical predecessor operation, combined with intersection and set difference operations. As in the previous work on symbolic transition systems, our state space exploration algorithm is semi-decidable for infinite-state systems. Together with the earlier work concerning the maximal reachability problem, the results presented here yield a semi-decidable algorithm for model checking symbolic systems against the probabilistic temporal logic PCTL. We illustrate our approach with the help of probabilistic timed automata, for which previous verification techniques suffered from an unnecessarily fine subdivisions of the state space, or which returned only estimates of the actual probabilities.

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

*Supported in part by the EPSRC grant GR/N22960.

controller) interacting with a finite set of real-valued variables (representing an analogue environment). The standard approach is to express system behaviour purely in terms of nondeterminism. However, in many cases, particularly in the context of fault-tolerant systems, it may be desirable, or behaviourally more faithful, to express the relative likelihood of the system exhibiting certain behaviour. Nondeterminism and probabilistic behaviour is inherent real-world protocols such as the IEEE 1394 FireWire root contention and IEEE 802.11 MAC. It also enables the modelling of asynchronous systems and the underspecification of system behaviour.

This paper continues our study of the verification problem for infinite-state systems featuring both nondeterminism and probabilistic behaviour. In [KNS01] we considered the *maximal reachability probability problem* for this class of system. Here we focus on the *minimal reachability probability problem*, namely the computation of the minimal probability with which a given set of target states is reachable. In the same way that reachability underlies the verification of temporal modalities in the non-probabilistic context, probabilistic reachability is fundamental for probabilistic model checking [BdA95]. The results presented here complete those of [KNS01], and together provide the foundations for model checking infinite-state probabilistic-nondeterministic systems against probabilistic temporal logics.

Symbolic probabilistic systems are infinite-state systems with a probabilistic transition relation and an algebra of operations on implicit, *symbolic* representations of possibly infinite state sets, and are inspired by the (non-probabilistic) symbolic transition systems of [HMR01]. The operations required on the symbolic representations of state sets, both in the non-probabilistic and probabilistic setting, include *boolean* and *predecessor* operations, which together enable model checking of reachability properties by *backwards exploration* of the state space. 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 by the traditional predecessor operation: the *probability* of taking the transition must also be known. Therefore, symbolic probabilistic systems also encode the transitions of a probabilistic system into a number of *transition 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 consists of symbolic encodings of both states *and* probabilistic transitions, together with an algebra of operations including the typed predecessor operations.

Our key contribution concerns the computation of the *minimum* reachability probability for certain classes of symbolic probabilistic systems by reduction to a finite-state problem. This is achieved through an algorithm which successively iterates typed predecessor, intersection and difference operations, starting from the target set. The main difficulty is that during backwards search the transitions that do not lead to the target states will never be encountered, but, in contrast to computing maximal reachability probabilities [KNS01], such transitions are important for the computation of *minimal* reachability probabilities. To include such transitions in our analysis we perform a *pre-computation* algorithm which finds those states that have positive minimum probability of

reaching the target. This necessitates the inclusion of the difference operation in addition to those required in [KNS01].

Related work. Approaches to infinite-state systems with discrete probability distributions include model checking methods for probabilistic lossy channel systems [BS02]. In the case of probabilistic timed automata, methods for computing *exact* reachability probabilities are presented in [KNSS02] and [KNS02] based on the *region graph* [AD94] and *digital clocks* [HMP92] respectively. However, both suffer 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). An alternative in [KNSS02] uses forwards reachability, however it is only able to compute *upper bounds* on the maximal reachability probabilities. We also mention [DJJL01] which uses abstraction and refinement methods to calculate bounds on the minimal and maximal reachability probabilities for probabilistic systems. Verification methodologies for infinite-state systems with *continuous* distributions are given in [BHHK00, DGJP00, KNSS00].

2 Symbolic probabilistic systems

2.1 Preliminaries

A discrete probability *(sub)distribution* over a finite set Q is a function $\mu : Q \to [0,1]$ such that $\sum_{q \in Q} \mu(q) \leq 1$. For a possibly uncountable set Q', let $\mathsf{Dist}(Q')$ be the set of distributions 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* $s \to t$ 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. A *path* of a probabilistic system is a finite or infinite sequence of probabilistic transitions of the form $\omega = s_0 \xrightarrow{\mu_0} s_1 \xrightarrow{\mu_1} \cdots$. 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 [Var85]. Formally, an *adversary* of $\mathbb{S}$ is a function A mapping every finite path ω to a distribution $\mu \in \mathit{Steps}(\mathit{last}(\omega))$. Let $\mathit{Adv}_{\mathbb{S}}$ be the set of adversaries of $\mathbb{S}$. For any $A \in \mathit{Adv}_{\mathbb{S}}$, let $\mathit{Path}^A_{\mathit{ful}}$ denote the set of infinite paths associated with A. Then, in the standard way, we define the measure Prob^A over $\mathit{Path}^A_{\mathit{ful}}$ [KSK76].

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

reaching U from s is defined as

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

The minimal reachability probability can be obtained as the solution to a linear programming problem in the case of finite systems [BdA95], and is useful to verify properties of the form "with probability at least 0.99, a data packet is delivered". In addition, for real-time systems, it can be used to verify time-bounded reachability properties, also known as *soft* deadlines, such as "with probability 0.975 or greater, a leader is elected within 100 time units".

2.2 Symbolic probabilistic systems: definition

Symbolic transition systems were introduced in [HMR01] as (possibly infinite-state) transition systems equipped with *region algebras*, comprising a set of *regions* (each element of which denotes a possibly infinite set of states), boolean, predecessor, emptiness and membership operations on regions. In [HMR01], classes of infinite-state systems for which a finitary structure can be identified by iteration of certain operations of the region algebra are defined, consequently highlighting the decidability of certain verification problems.

To represent probabilistic behaviour as well as nondeterminism in such a symbolic framework and to allow for quantitative reasoning we augment the framework of symbolic transition systems with a *symbolic encoding of probabilistic transitions*. The symbolic representation, first presented in [KNS01], is derived in two steps. First we encode the state transitions induced by the probabilistic transitions of the system within a set of *transition types*. Note that, by introducing such transition types we must also replace the single predecessor operation present in the framework of symbolic (non-probabilistic) transition systems with a *family of predecessor operations* index by the set of transition types. The second step is to 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.

We now give the definition of symbolic probabilistic systems which generalise the symbolic transition systems of [HMR01], and extends the symbolic probabilistic transition systems of [KNS01] to allow for the computation of minimal reachability probabilities in addition to maximal reachability probabilities. The definition of regions 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, the difference being the typed predecessor operations.

Definition 1 (Symbolic Probabilistic Systems) *A* symbolic probabilistic system $\mathbb{P} = (S, Steps, R, \ulcorner\cdot\urcorner, \mathcal{T}ra, \mathsf{D})$ *comprises: a probabilistic system* $(S, Steps)$*; a set of regions* 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* $s, t \in S$*:*

(a) *if* $a \in \mathcal{T}ra$ *and* $t \in \delta_a(s)$*, then there exists* $\mu \in Steps(s)$ *such that* $\mu(t) > 0$*;*

(b) *if* $\mu \in Steps(s)$*, then there exists* $\nu \in \mathsf{D}$ *with* $s \in enabled(\nu)$ *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* $\nu \in \mathsf{D}$*,* $s \in enabled(\nu)$ *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* $\mu \in Steps(s)$ *such that:*

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

where $\mathcal{T}ra(s) = \{a \mid a \in \mathcal{T}ra(s) \wedge \delta_a(s) \neq \varnothing\}$ *and* $s \in enabled(\nu)$ *if and only if* $\delta_a(s) \neq \varnothing$ *for all* $a \in \mathcal{T}ra$ *such that* $\nu(a) > 0$*.*

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 all* $\sigma, \tau \in R$*.*

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 all* $\sigma, \tau \in R$*.*

5. *There is a computable function* $\mathsf{Empty} : R \to \mathbb{B}$ *such that* $\mathsf{Empty}(\sigma)$ *if and only if* $\ulcorner \sigma \urcorner = \varnothing$ *for all* $\sigma \in R$*.*

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 all* $s \in S$ *and* $\sigma \in R$*.*

Note that, using And and Diff operations, we can define a computable function $\mathsf{Or} : R \times R \to R$ such that $\ulcorner \mathsf{Or}(\sigma, \tau) \urcorner = \ulcorner \sigma \urcorner \cup \ulcorner \tau \urcorner$ for all $\sigma, \tau \in R$.

The difference between the framework presented here and that of [KNS01] lies in the need to introduce *enabledness* (conditions *1(b)* and *1(c)* of Definition 1). For completeness we first recall the intuition for transition types and distribution templates before moving on to describing the amendments to *1(b)* and *1(c)*.

Transition types. Transition types encode a set of state transitions of a symbolic probabilistic system. With each transition type $a \in \mathcal{T}ra$ we associate a transition relation $\delta_a : S \to 2^S$ encoding all 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. The lemma below states that every state transition is represented by a transition encoded with some transition type, and vice versa.

Lemma 2 *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. Distribution templates are used to encode the actual probabilities featured in the system. The condition *1(b)* represents the fact that the probabilistic branching structure of the system is modelled by the distribution templates. Dually, condition *1(c)* expresses the fact that, in all states, any transition encoded by an *enabled* distribution template corresponds to a transition of the system.

Example 1. Consider a system in which the state space takes the form of valuations of a single real-valued variable x, where in any $s \in (0,4)$, the variable x can be reset nondeterministically in (1,3) and (2,4), each with probability 0.5. Consider representing the system as a symbolic probabilistic system, where the set of regions 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 $\delta_b(s) = (2,4)$ for $s \in (0,4)$ and $\delta_a(s) = \delta_b(s) = \varnothing$ for $s \notin (0,4)$, 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 $\mu_{s'} \in \mathit{Steps}(s)$ 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 $t_a = t_b = s'$. Note that, the distribution template ν is only enabled in the region $(0,4)$.

Comparison with [KNS01]. Recall that the framework of [KNS01] was defined specifically for maximal reachability, whereas here we additionally allow for the computation of minimal reachability. In [KNS01] condition *1(c)* is of the form:

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

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

This expresses the fact that, in all states, for any transition encoded by a distribution template, there exists a system transition which assigns an *greater or equal* probability to all target states. This implies that in certain states there may be distribution templates which do not correspond to actual transitions of the system, but is nevertheless sufficient for the computation of the maximal reachability probability. Clearly, this does not suffice for minimum reachability.

We resolve the problem by restricting attention to *enabled* distribution templates. To calculate quantitative probabilistic properties there must exist a correspondence between the transitions encoded by a distribution template and system transitions. However, requiring that for any state and distribution template there is a corresponding system transition would be too restrictive. We therefore restrict attention to enabled distribution template. Where a template is enabled if and only if all the corresponding transitions types of the template are enabled, that is, each transition type which *can occur with a non-zero probability* has a corresponding system transition. Note that alternative approaches to defining enabledness are possible, but we will see below the advantage of this approach.

Enabled distribution templates. Recall that, for any $\nu \in \mathsf{D}$, $\mathit{enabled}(\nu)$ represents the states from which ν has a corresponding system transition. Now, since by definition

$$\mathit{enabled}(\nu) = \cap\{\ulcorner \mathsf{pre}_a(R) \urcorner \,|\, a \in \mathcal{T}\!\mathit{ra} \wedge \nu(a) > 0\}$$

it follows that we can we can define a *computable* function $\mathsf{Enabled}$ from distribution templates to *regions*, as opposed to sets of states, using pre and And.

The only condition we impose on this function is that for any $\sigma \in R$, $\nu \in \mathsf{D}$ and $a \in \mathcal{T}\!\mathit{ra}$ such that $\nu(a) > 0$ we have

$$\ulcorner \mathsf{pre}_a(\sigma) \urcorner \subseteq \ulcorner \mathsf{Enabled}(\nu) \urcorner \,. \tag{1}$$

Note that, even in the cases where this condition does not hold, it can be enforced by straightforward renaming of the transition types. In particular, we can define the new transition types to be pairs of the form (a, ν), where $a \in \mathcal{T}\!\mathit{ra}$ and $\nu \in \mathsf{D}$ such that $\nu(a) > 0$, and then the new predecessor operator for such a transition type can be given by:

$$\mathsf{pre}_{(a,\nu)}(\sigma) \stackrel{\text{def}}{=} \mathsf{And}(\mathsf{pre}_a(\sigma), \mathsf{Enabled}(\nu)) \,.$$

If initially the set of distribution templates and transition types are finite, then the resulting set of transition types are finite (the distribution templates do not change).

We should add that if (1) holds then all the results presented in [KNS01] carry over to this new framework.

Finiteness of transition types and templates. Observe that the sets of transition types and distribution templates may be infinite. However, as in [KNS01], we restrict the analysis to systems with finite sets of distribution templates and transition types. This implies that our method is appropriate for classes of system exhibiting finite *regularity* in probabilistic transitions. For example, probabilistic lossy channels [BS02] cannot be modelled by a finite set of distribution templates, because the probability of message loss varies with the quantity of data in the unbounded buffers.

2.3 Example: Probabilistic Timed Automata

The fact that probabilistic timed automata [KNSS02] can be represented as symbolic probabilistic systems follows from a similar result given in [KNS01]. We assume familiarity with the classical, non-probabilistic timed automaton model [AD94, HNSY94] and for an in-depth introduction to probabilistic timed automata, refer to [KNSS02]. As explained in [KNS01], the translation method can be adapted to classes of *probabilistic hybrid automata* [Spr00, Spr01], which are hybrid automata [ACH$^+$95] augmented with a probabilistic edge relation. similar to that featured in the definition of probabilistic timed automata, given an appropriate set of regions and algebra of operations.

Let $\mathcal{X}$ be a set of real-valued variables called *clocks*. Let $\mathit{Zones}(\mathcal{X})$ be the set of *zones* over $\mathcal{X}$, which are conjunctions of atomic constraints of the form

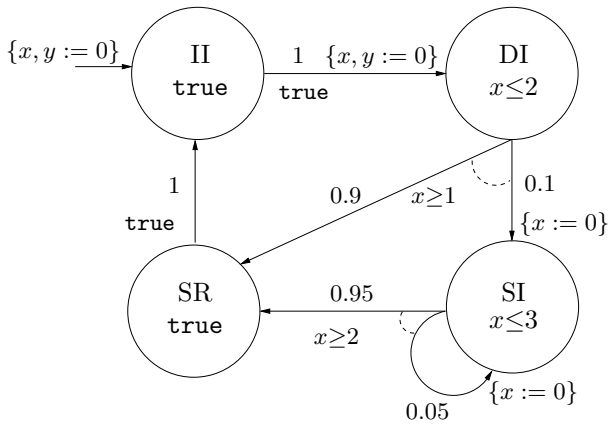

Figure 1: A probabilistic timed automaton modelling a probabilistic protocol.

$x \sim c$ 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 $x \in \mathcal{X}$ with the corresponding value v_x from v. A zone ζ over $\mathcal{X}$ defines a convex polyhedral subset of $\mathbb{R}^{|\mathcal{X}|}$ corresponding to the set of clock valuations which satisfy it.

Definition 3 (Probabilistic Timed Automata) *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 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 l' and resetting all of the clocks in X to 0 is given by $p(l', X)$.

Example 2. Consider the 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 PTA as a symbolic probabilistic system, we introduce the

following definitions. For $v \in \mathbb{R}^{|\mathcal{X}|}$ and $\eta \in \mathbb{R}_{\geq 0}$, the clock valuation $v + \eta$ is obtained from v by adding η to the value of each clock; and, for any $X \subseteq \mathcal{X}$, the clock valuation $v[X := 0]$ is obtained from v by resetting all clocks in X to 0. Now, for zone ζ and $\eta \geq 0$, let $\zeta + \eta$, be the expression in which each $x \in \mathcal{X}$ is replaced syntactically by $x + \eta$ in ζ, and let $[X := 0]\zeta$ be the expression in which each $x \in X$ is replaced syntactically by 0 in ζ. The *edges* of PTA, denoted 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 PTA 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 of PTA [KNSS02].
- The set of regions R 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 any region $(l, \zeta) \in R$.
- The set of transition types $\mathcal{T}\mathit{ra}$ is the set E_{PTA} of edges of PTA plus a special type *time* such that for any state $(l, v) \in S$ and edge $(l', l'', X, \zeta) \in E_{\mathsf{PTA}}$:

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

  and

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

- The set D is such that $\nu \in \mathsf{D}$ if and only if one of the following conditions hold:
  1. there exists $l \in L$ and $p \in \mathit{prob}(l)$ such that, for all $a \in \mathcal{T}\mathit{ra}$:

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

  2. $\nu(\mathit{time}) = 1$.

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

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

$$\mathsf{pre}_{(l',l'',X,\zeta')}(l, \zeta) = \begin{cases} (l', (\zeta' \wedge \mathit{inv}(l') \wedge [X := 0](\zeta \wedge \mathit{inv}(l)))) & \text{if } l{=}l'' \\ (l, \mathtt{false}) & \text{otherwise} \end{cases}$$

$$\mathsf{pre}_{\mathit{time}}(l, \zeta) = (l, (\exists \eta{\geq}0 \,.\, \zeta{+}\eta \wedge \forall\, 0{\leq}\eta'{\leq}\eta \,.\, \mathit{inv}(l){+}\eta')) \,.$$

Observe that these operations are defined in terms of pairs of locations and constraints on clocks. By classical timed automata theory [HNSY94], 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. 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 $(l', v) \in S$, ν is enabled if and only if $l' = l$ and $v \models g_l(p)$ and condition *1(c)* follows as in the case of *1(b)*.

3 Minimal Reachability Algorithm

We now present a *semi-decidable* algorithm (semi-algorithm), solving the minimal reachability probability problem for symbolic probabilistic systems by *backwards exploration* through the state space. As mentioned above, we restrict attention to symbolic probabilistic systems with *finite* sets of transition types and distribution templates. Note that, even for 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 $F \in R$ be the target region for which the minimal reachability probability is to be computed.

The approach 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. However, since the semi-algorithm is based on performing a backwards search starting from F the transitions that do not lead to the target set will never be encountered. In the maximal reachability case [KNS01] this was not problematic since in this case we need only concern ourselves with ways of reaching the target set. However, to solve the minimal reachability problem such transitions are important – if there are system transitions which do not lead to the target set then the minimum probability of reaching this set will be zero. We proceed by performing a *precomputation* semi-algorithm which computes the set of states for which the minimum probability of reaching the target set is *greater than zero*. For such states *all* transitions eventually lead to the target set with non-zero probability, and hence performing a backwards exploration on this set will suffice for calculating the minimal

reachability probability.

3.1 Precomputation Algorithm

In this section we present an algorithm for calculating the set of states which have a positive minimum probability of reaching the target set of states. In the finite-state case [BdA95] the algorithm is given by:

input: target set $F \subseteq S$
$T := \varnothing$;
$T' := F$;
while $T \neq T'$
  $T := T'$
  $T' := \{s \mid \forall \mu \in \mathit{Steps}(s).(\exists s' \in T.\mu(s') > 0)\} \cup T$
end while

In order to extend this to our symbolic framework we must express the region corresponding to:

$$\{s \mid \forall \mu \in \mathit{Steps}(s).(\exists s' \in T.\mu(s') > 0)\} ,$$

that is, the set of states from which under any transition one always reach T with positive probability, in terms of the symbolic operations. For a fixed distribution template ν, we first compute the set of states from which, if one chooses to make a transition according to ν, then the probability of reaching T is always positive. A first attempt at defining such a region might be:

$$\mathsf{Or}\{\mathsf{pre}_a(T) \mid a \in \mathcal{T}ra \wedge \nu(a) > 0\} .$$

However, the above does not take into account all the possible transitions corresponding to ν. since in any state performing a transition type can lead to a set of possible successor states. The correct formulation is to consider the region $\mathsf{pre}_\nu(T)$ given by:

$$\mathsf{Or}\{\mathsf{Diff}(\mathsf{pre}_a(T), \mathsf{pre}_a(R \setminus T)) \mid a \in \mathcal{T}ra \wedge \nu(a) > 0\}.$$

The region $\mathsf{Diff}(\mathsf{pre}_a(T), \mathsf{pre}_a(\mathsf{Diff}(R, T)))$ yields all states from which performing a transition of type a always leads to T. Formally, we have the following lemma.

Lemma 4 *For any $\nu \in \mathcal{T}ra$ and $s \in \ulcorner \mathsf{Enabled}(\nu) \urcorner$: $s \in \ulcorner \mathsf{pre}_\nu(T) \urcorner$ if and only if for all $\mu \in \mathit{Steps}(s)$ which can be generated by the distribution template ν there exists $t \in \ulcorner T \urcorner$ such that $\mu(t) > 0$.*

Proof. The "if" direction: suppose that for all $\mu \in \mathit{Steps}(s)$ which can be generated by the distribution template ν there exists $t \in \ulcorner T \urcorner$ such that $\mu(t) > 0$. Now for any distribution $\mu \in \mathit{Steps}(s)$ constructed from ν, by definition there exists a 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:

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

Now suppose that $s \notin \mathsf{pre}_\nu(T)$, that is, there does not exist $a \in \mathcal{T}ra(s)$ with $\nu(a) > 0$ such that $s \in \ulcorner \mathsf{pre}_a(T) \setminus \mathsf{pre}_a(R \setminus \ulcorner T \urcorner) \urcorner$. By definition, if follows that for all $a \in \mathcal{T}ra(s)$ with $\nu(a) > 0$ there exists $t'_a \in \delta_a(s) \setminus T$. Therefore, if the distribution μ' is constructed from the vector $\langle t'_a \rangle_{a \in \mathcal{T}ra(s)} \in \prod_{a \in \mathcal{T}ra(s)} \delta_a(s)$ (where t'_a is arbitrary if $\nu(a) = 0$), then $\mu'(t) = 0$ for all $t \in \ulcorner T \urcorner$. This contradicts the fact that for all $\mu \in \mathit{Steps}(s)$ which can be generated by the distribution template ν there exists $t \in \ulcorner T \urcorner$ such that $\mu(t) > 0$.

The "only if" direction: suppose $s \in \ulcorner \mathsf{pre}_\nu(T) \urcorner$, then $s \in \mathsf{pre}_a(T) \setminus \mathsf{pre}_a(R \setminus T)$ for some $a \in \mathcal{T}ra(s)$ such that $\nu(a) > 0$. It follows that $\delta_a(s) \subseteq \ulcorner T \urcorner$ for some $a \in \mathcal{T}ra$ such that $\nu(a) > 0$, and hence for all $\mu \in \mathit{Steps}(s)$ which can be generated by the distribution template ν there exists $t \in \ulcorner T \urcorner$ such that $\mu(t) > 0$. □

We can now extend this to all transitions via distribution templates. Since the system transitions in a state are related to only the distribution templates *enabled* in the state, we must therefore consider the states where different sets of distribution templates are enabled separately. For any set of distribution templates $D \subseteq \mathsf{D}$, the region

$$\mathsf{Diff}(\mathsf{And}\{\mathsf{pre}_\nu(T) \mid \nu \in D\}, \mathsf{Or}\{\mathsf{Enabled}(\nu) \mid \nu \in \mathsf{D} \setminus D\})$$

yields all states for which the set of distribution templates enabled equals D and under any system transition the probability of reaching T is greater than zero. More formally, we have the following lemma.

Lemma 5 *Let* $D_s = \{\nu \mid \nu \in \mathsf{D} \wedge s \in \ulcorner \mathsf{Enabled}(s) \urcorner\}$*, for any* $s \in S$ *and* $D \subseteq \mathsf{D}$*:*

$$s \notin \ulcorner \mathsf{Diff}(\mathsf{And}\{\mathsf{pre}_\nu(T) \mid \nu \in D\}, \mathsf{Or}\{\mathsf{Enabled}(\nu) \mid \nu \in \mathsf{D} \setminus D\}) \urcorner$$

and if $D = D_s$*:*

$$s \in \ulcorner \mathsf{Diff}(\mathsf{And}\{\mathsf{pre}_\nu(T) \mid \nu \in D\}, \mathsf{Or}\{\mathsf{Enabled}(\nu) \mid \nu \in \mathsf{D} \setminus D\}) \urcorner$$

if and only if for all $\mu \in \mathit{Steps}(s)$ *there exists* $t \in T$ *such that* $\mu(t) > 0$*.*

Proof of Lemma 5. The first part follows from the fact that if $D \subset D_s$, then $s \in \ulcorner \mathsf{Enabled}(\nu) \urcorner$ for some $\nu \in D_s \setminus D$ and if $D \not\subseteq D_s$, then there exists $\nu \in D \setminus D_s$ such that $s \notin \mathsf{pre}_\nu(T)$ (since we require the $\ulcorner \mathsf{pre}_a(T) \urcorner \subseteq \ulcorner \mathsf{Enabled}(\nu) \urcorner$ for all $a \in \mathcal{T}ra$ such that $\nu(a) > 0$ to hold).

For the second part suppose that $D = D_s$. Now, by definition of D_s we have $s \notin \ulcorner \mathsf{Or}\{\mathsf{Enabled}(\nu) \mid \nu \in \mathsf{D} \setminus D\} \urcorner$, and hence the proof reduces to showing that: $s \in \ulcorner \mathsf{And}\{\mathsf{pre}_\nu(T) \mid \nu \in D\} \urcorner$ if and only if for all $\mu \in \mathit{Steps}(s)$ there exists $t \in T$ such that $\mu(t) > 0$. The result then follows from Lemma 14 and since all the distributions in $\mathit{Steps}(s)$ are generated by some $\nu \in D$. □

Using the above results, the symbolic version of the precomputation algorithm is given in Figure 2. Note that the termination test $\ulcorner T' \urcorner \subseteq \ulcorner T \urcorner$ denotes the test $\{\ulcorner \sigma \urcorner \mid \sigma \in T'\} \subseteq \{\ulcorner \sigma \urcorner \mid \sigma \in T\}$, and is computable [HMR01]. From Lemma 5, under the assumption that the algorithm terminates, it follows that $s \in \ulcorner T_{\min} \urcorner$ if and only if the minimum probability of reaching T is positive. Formally, we have the following proposition.

Symbolic semi-algorithm PreMinReach
input: $(R, \mathcal{T}ra, \{\mathsf{pre}_a\}_{a\in\mathcal{T}ra}, \mathsf{And}, \mathsf{Diff}, \mathsf{Empty}, \mathsf{Member})$
target set $F \in R$
$T := \varnothing$;
$T' := F$;
while $\ulcorner T' \urcorner \subseteq \ulcorner T \urcorner$
$T := T'$
for all $D \subseteq \mathsf{D}$ **do**
$T' := \mathsf{Diff}(\mathsf{And}\{\mathsf{pre}_\nu(T) \mid \nu \in D\}, \mathsf{Or}\{\mathsf{Enabled}(\nu) \mid \nu \in \mathsf{D} \setminus D\}) \cup T$
end for all
end while
return $T_{\min} := T$

Figure 2: Precomputation algorithm

Proposition 6 *If with input given by the symbolic probabilistic system* $\mathbb{P} = (S, \mathit{Steps}, R, \ulcorner\cdot\urcorner, \mathcal{T}ra, \mathsf{D})$ *and target set* $F \in R$ *the algorithm* PreMinReach *generates the region* $T_{\min}$*, then for any* $s \in S$*:* $s \in \ulcorner T_{\min} \urcorner$ *if and only if the minimum probability of reaching* F *is greater than zero.*

Proof of Proposition 6. The proof follows from Lemma 5 similarly to the finite state case [BdA95]. □

3.2 Main Algorithm

In this section we present our main algorithm for calculating the minimum probability of reaching a target set of states. The algorithm relies on first invoking the precomputation algorithm, given in Figure 2, for finding the set of states for which the minimum probability of reaching the target set of states is positive. In particular, we intersect all the predecessor operations with the region generated by the precomputation algorithm. More formally, supposing $T_{\min}$ is the set of regions generated by the precomputation algorithm, for any $\sigma \in R$ and $a \in \mathcal{T}ra$ we let $\mathsf{pre}_a(\sigma, T_{\min}) = \mathsf{And}(\mathsf{pre}_a(\sigma), T_{\min})$.

The algorithm MinReach proceeds by successive iteration of predecessor, intersection and set difference operations. For each $i \in \mathbb{N}$ and for all currently generated regions in the set T_i, the algorithm constructs the set T_{i+1} of regions by adding to T_i the typed predecessors of the regions in T_i, taking the intersection and difference of these predecessors with regions in T_i. The algorithm is given in Figure 3.

The vital component in the above algorithm is that the set of regions T_i is kept pairwise disjoint at each iteration. More formally, for each new region we first add the part which we have not encountered yet (disjoint from T_{i+1}),

$$\mathsf{Diff}(\mathsf{pre}_a(\sigma, T_{\min}), \mathsf{Or}\{\tau \mid \tau \in T_{i+1}\}),$$

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

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

Figure 3: The main algorithm

and then split everything else we have found so far ($\tau \in T_{i+1}$) into two regions:

$$\mathsf{And}(\mathsf{pre}_a(\sigma, T_{\min}), \tau) \text{ and } \mathsf{Diff}(\tau, \mathsf{pre}_a(\sigma, T_{\min}))\,.$$

In the maximal reachability case [KNS01], we did not need to keep the sets disjoint, and hence the only required operations were predecessor and intersection. This is because, in the maximal case, we only need to know when transitions are enabled, not if they are the *only* transitions enabled. However, this is not the case for calculating minimal probabilities as will be illustrated in *Example 3* below, after we explain the role of the ExtendEdges procedure.

At each step the edge relation E is expanded to relate the existing regions to their newly generated typed predecessors. Then, if the outer **for** loop of the symbolic semi-algorithm MinReach terminates, then we call the procedure ExtendEdges on the graph (T, E). Intuitively, for a particular edge $(\sigma, a, \tau) \in E$ where $\tau \in T$, the procedure constructs edges with the transition type a and target region τ for all subset regions of σ in T.

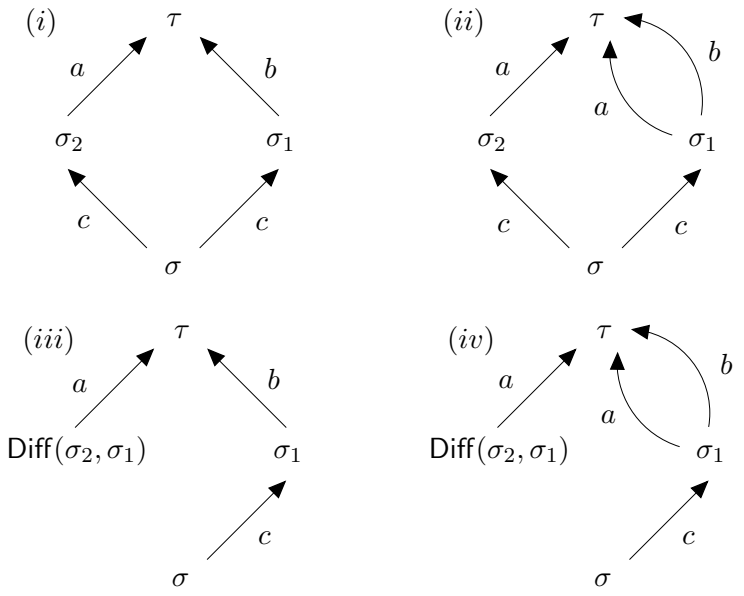

Figure 4: Example demonstrating the need for disjoint regions.

Example 3. Suppose during a backwards search regions are not kept disjoint and the resulting graph is given by graph (i) of Figure 4. If $\ulcorner \sigma_1 \urcorner \subseteq \ulcorner \sigma_2 \urcorner$ and all other regions are disjoint, then the procedure ExtendEdges will insert an extra edge (σ_1, a, τ) shown in graph (ii) of Figure 4. However, suppose that $\mathsf{Diff}(\sigma_1, \sigma_2)$ is unreachable, and hence $\ulcorner \mathsf{pre}_a(\mathsf{Diff}(\sigma', \sigma)) \urcorner = \varnothing$ for all $a \in \mathcal{T}ra$. If we perform a backwards search keeping sets disjoint, the resulting graph is given by (iii) of Figure 4 (and graph (iv) of Figure 4 after performing the procedure ExtendEdges).

Therefore, if regions are not kept disjoint, it is possible to reach τ from σ by first performing an c transition type in such a way that in the resulting state the only transition type which reaches τ is a (choosing the left branch of graph (ii) of Figure 4). On the other hand, by our algorithm, to reach τ from σ, after performing a transition of type c, both the transition types a and b reach τ. Now, suppose all distribution templates enabled in the region σ_1 choose both a and b with positive probability, then incorrect results are obtained from (ii).

Note that in the ExtendEdges procedure all edges whose destination regions do not appear in T are ignored. For example, we ignore the edge (σ, c, σ_2), that is, we do not add an edge of the form $(\sigma, c, \mathsf{Diff}(\sigma_2, \sigma_1))$ to graph (iv) of Figure 4 even though $\ulcorner \mathsf{Diff}(\sigma_2, \sigma_1) \urcorner \subseteq \ulcorner \sigma_2 \urcorner$.

3.2.1 Termination of PreMinReach and MinReach.

Termination of PreMinReach is reliant on the termination of the outer (**while**) loop. Termination of MinReach is dependent on the termination of its outer **for** loop, since, when this terminates, both T and E are finite, and hence the procedure ExtendEdges will also terminate. Observe that the inner **for** loop of PreMinReach will not terminate if the set D is not finite, and the inner **for** loop

of MinReach 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*. We say that two states $s, t \in S$ are *typed-bisimilar*, denoted $s \cong t$ if there exists a symmetric typed simulation on S such that $s \preceq t$. We call the equivalence relation $\cong$ a *typed bisimilarity*, and say $\cong$ has *finite index* if there are finitely many equivalence classes of $\cong$.

The arguments of [HMR01] are adapted to show that both PreMinReach and MinReach will terminate for any symbolic probabilistic system for which the typed bisimilarity relation $\cong$ has finite index, given that the target set F is a set of equivalence classes of $\cong$. That is, we show that for all $\sigma \in T$, the set $\ulcorner\sigma\urcorner$ is a union of equivalence classes of $\cong$. Probabilistic timed automata and probabilistic singular automata exhibit such a relation.

3.2.2 Encoding the Probabilistic Behaviour

As we have seen the algorithm uses only the transition types and not the distribution templates to construct the graph (T, E). In this section we use the distribution templates to construct a finite-state probabilistic system, the states of which are the regions generated by the algorithm MinReach, and the transitions of which are induced by the set of edges E and the 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 region 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 regions and the transition types of the edges in the identified group. Formally, we define a *probabilistic system* $\mathbb{Q} = (T, \mathit{Steps}_{\mathbb{Q}})$, where $\mathit{Steps}_{\mathbb{Q}} : T \to 2^{\mathsf{Dist}(T)}$ is the probabilistic transition relation $\mathit{Steps}_{\mathbb{Q}}$ constructed as follows. For any region $\sigma \in T$, 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, then $a \neq a'$;

3. the set E_π is maximal with respect to set inclusion;

4. for all regions $\tau \in T$:

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

For any $\sigma \in T$, 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.

Symbolic semi-algorithm PTAPreMinReach
input: $(R, \mathcal{T}ra \cup \{time\}, \{\mathsf{pre}_a\}_{a \in \mathcal{T}ra \cup \{time\}}, \mathsf{And}, \mathsf{Diff}, \mathsf{Empty}, \mathsf{Member})$
target set $F \in R$
$T := \varnothing$;
$T' := \mathsf{Diff}(\mathsf{Diff}(\mathsf{pre}_{time}(F), F), \mathsf{Or}\{\mathsf{pre}_{time}(\mathsf{Enabled}(\nu)) \mid \nu \in \mathsf{D}\}) \cup \{F\}$;
while $\ulcorner T' \urcorner \subseteq \ulcorner T \urcorner$
$T := T'$
for all $D \subseteq \mathsf{D}$ **do**
$T' := \mathsf{Diff}(\mathsf{And}\{\mathsf{pre}_\nu^{\mathsf{PTA}}(T) \mid \nu \in D\}, \mathsf{Or}\{\mathsf{pre}_{time}(\mathsf{Enabled}(\nu)) \mid \nu \in \mathsf{D} \setminus D\}) \cup T$
end for all
end while
return $T_{\min} := T$

where $\mathsf{pre}_\nu^{\mathsf{PTA}}(T) \stackrel{\text{def}}{=} \mathsf{Diff}(\mathsf{pre}_{time}(\mathsf{pre}_\nu(T)), \mathsf{pre}_{time}(\mathsf{Diff}(\mathsf{Enabled}(\nu), \mathsf{pre}_\nu(T))))$

Figure 5: Precomputation algorithm for probabilistic timed automata

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.

3.2.3 Correctness

We now state the formal correctness of our algorithm.

Theorem 7 *If $\mathbb{Q} = (T, Steps_{\mathbb{Q}})$ is the probabilistic system constructed through the algorithm* MinReach *with input given by the symbolic probabilistic system $\mathbb{P} = (S, Steps_{\mathbb{P}}, R, \ulcorner \cdot \urcorner)$ and target set $F \in R$, then for any state $s \in S$ with $MinReach(s, symbF) > 0$:*

$$MinReach(s, \ulcorner F \urcorner) = MinReach(\sigma, F)$$

where $\sigma \in T$ is such that $s \in \ulcorner \sigma \urcorner$.

Proof. See appendix. □

Recall from Section 2.1 that the minimal reachability probability for finite probabilistic systems can be computed using established methods [BdA95].

3.2.4 Probabilistic Timed Automata.

In the case of probabilistic timed automata an additional requirement is necessary to compute the minimal reachability probability due to *time divergence*. We admit only *time divergent adversaries*, which necessitates modifications to the algorithm and consequent restrictions on the probabilistic timed automata. First, we assume that probabilistic timed automata satisfy the condition: for any adversary which makes discrete transitions infinitely often there exists an equivalent divergent adversary (i.e. making the same discrete choices).

To restrict attention to divergent adversaries, we remove the locations where time can diverge (i.e. have unbounded invariants). Then, the behaviour, for any remaining location and divergent adversary, corresponds to letting time elapse (possibly 0 time units) and then performing a discrete transition.

Furthermore, since time transitions (transitions of type *time*) must be treated as a special case, we modify the precomputation algorithm to that given in Figure 5. First, add to F its time predecessors, which must reach F before a discrete transition can be performed. Second, in our backwards search we use the function $\ulcorner \mathsf{pre}_\nu^{\mathsf{PTA}}(\cdot) \urcorner$, which, for any region T, returns the set of states where, if time elapses (possibly 0 time units) and a state transition which can be generated by ν is performed, then a state in T is reached with positive probability. Also, note that the region $\mathsf{Diff}(\mathsf{And}\{\mathsf{pre}_\nu^{\mathsf{PTA}}(T) \mid \nu \in D\}, \mathsf{Or}\{\mathsf{pre}_{time}(\mathsf{Enabled}(\nu)) \mid \nu \in \mathsf{D} \setminus D\})$ corresponds to the set of states which, by letting time elapse one can only perform transitions represented by a distribution template in D, and, for any such a transition T is reached with positive probability.

The prove the correctness of $\mathsf{PTAPreMinReach}$, that is, the fact that the algorithm returns exactly those states for which, under the set of time-divergent adversaries, the minimum probability of reaching the target set of states is greater then zero, w first require the following two lemmas.

Lemma 8 *For any $\nu \in \mathcal{T}ra$ and $s \in S$: $s \in \ulcorner \mathsf{pre}_\nu^{\mathsf{PTA}}(T) \urcorner$ if and only if from s it is possible to let time elapse (possibly 0 time units) and reach a state where a discrete transition which can be generated from the distribution template ν can be performed and, for any such state, all discrete transitions from this state which can be generated by the distribution template ν lead to T with positive probability.*

Proof. The "if" direction: suppose from s it is possible to let time elapse and reach a state where a discrete transition which can be generated from the distribution template ν can be performed and, for any such state, all discrete transitions from this state which can be generated by the distribution template ν lead to T with positive probability. Using Lemma 14 it follows that the set of states which can be reached from s by letting time elapse and can perform a transition generated from the distribution template ν is a subset of $\ulcorner \mathsf{pre}_\nu(T) \urcorner$, and hence from s by letting time elapse, one cannot reach a state in $\ulcorner \mathsf{Diff}(\mathsf{Enabled}(\nu), \mathsf{pre}_\nu(T)) \urcorner$. Moreover, since s can by letting time elapse reach a state where such a transition is possible we have:

$$s \in \ulcorner \mathsf{Diff}(\mathsf{pre}_{time}(\mathsf{pre}_\nu(T)), \mathsf{pre}_{time}(\mathsf{Diff}(\mathsf{Enabled}(\nu), \mathsf{pre}_\nu(T)))) \urcorner$$

as required.

The "only if" direction: suppose $s \in \ulcorner \mathsf{pre}_\nu^{\mathsf{PTA}}(T) \urcorner$, then $s \in \ulcorner \mathsf{pre}_{time}(\mathsf{pre}_\nu(T)) \urcorner$, and hence s can reach a state in $\mathsf{pre}_\nu(T)$ by letting time advance. Moreover, $s \notin \ulcorner \mathsf{pre}_{time}(\mathsf{Diff}(\mathsf{Enabled}(\nu), \mathsf{pre}_\nu(T))) \urcorner$, and thus the only states where a transition which can be generated by ν can be performed that s can reach by letting time advance are in $\mathsf{pre}_\nu(T)$. The result then follows from Lemma 14. □

Lemma 9 *Letting* $D_s^{\mathsf{PTA}} = \{\nu \,|\, \nu \in \mathsf{D} \wedge t \in \ulcorner \mathsf{Enabled}(s) \urcorner$ *for some* $t \in \delta_{time}\}$, *for any* $s \in S$ *and* $D \subseteq \mathsf{D}$*: if* $D \neq D_s^{\mathsf{PTA}}$, *then*

$$s \notin \ulcorner \mathsf{Diff}(\mathsf{And}\{\mathsf{pre}_\nu^{\mathsf{PTA}}(T) \,|\, \nu \in D\}, \mathsf{Or}\{\mathsf{pre}_{time}(\mathit{Enabled}(\nu)) \,|\, \nu \in \mathsf{D} \setminus D\}) \urcorner$$

and if $D = D_s^{\mathsf{PTA}}$*:*

$$s \in \ulcorner \mathsf{Diff}(\mathsf{And}\{\mathsf{pre}_\nu^{\mathsf{PTA}}(T) \,|\, \nu \in D\}, \mathsf{Or}\{\mathsf{pre}_{time}(\mathit{Enabled}(\nu)) \,|\, \nu \in \mathsf{D} \setminus D\}) \urcorner$$

if and only if from s *all discrete transitions, which can be performed after letting time elapse (possibly 0 time units), reach* T *with positive probability.*

Proof. The first part is a direct result of the following two observations.

1. If $D \subset D_s^{\mathsf{PTA}}$, then there exists $\nu \in D_s^{\mathsf{PTA}} \setminus D$ and state $t \in S$ such that $t \in \mathsf{Enabled}(\nu)$ and $t \in \delta_{\mathsf{time}}(s)$, and hence $s \in \mathsf{pre}_{time}(\mathsf{Enabled}(\nu))$.
2. If $D \not\subseteq D_s^{\mathsf{PTA}}$, then there exists $\nu \in D \setminus D_s^{\mathsf{PTA}}$ such that $s \notin \mathsf{pre}_\nu^{\mathsf{PTA}}(T)$.

Now for the second part assume that $D = D_s^{\mathsf{PTA}}$. The result then follows using Lemma 8 and noting from s, after letting time elapse, a transition which can be generated by the distribution template ν can be performed if and only if $\nu \in D_s^{\mathsf{PTA}}$. □

Proposition 10 *If for the target region* F *the algorithm* PTAPreMinReach *generates the set of region* $T_{\min}$ *, then* $s \in \ulcorner T_{\min} \urcorner$ *if and only if the minimum probability of reaching* F*, under the set of divergent adversaries, is greater than zero.*

Proof. First, by Lemma 9 and the fact that we have removed the locations in which time can diverge, it follows that: $s \in \ulcorner T_{\min} \urcorner$ if and only if under the set of adversaries which in any location where time cannot diverge eventually makes a discrete transition the minimum probability of reaching F is positive. Now since we assume that the probabilistic timed automata that for any adversary which makes discrete transitions infinitely often there exists an equivalent divergent adversary, if follows that this probability is equal to the minimum under the set of divergent adversaries. □

For the main algorithm (given in Figure 3), the only modification is that we remove from E any edge of the form $(\sigma, \mathit{time}, \sigma)$, since removing loops only affects the probability when they are taken infinitely often and such behaviour is time convergent. Therefore, using Theorem 7 and Proposition 10, it follows that the minimum probability of reaching F for the probabilistic system constructed through the algorithm MinReach equals, in the probabilistic timed automata, the minimum probability of reaching $\ulcorner F \urcorner$ under the set of adversaries which in any location where time cannot diverge eventually make a discrete transition. Then, similarly to the proof of Proposition 10, it follows that it also equals the minimum probability under the set of divergent adversaries.

3.2.5 Example

If we now return to *Example 2* of the PTA given in Figure 1 to find the minimal probability of reaching location SR (correct receipt of a message) within 4 time units of the data arriving at the sender (F equals $\langle \mathrm{SR}, y{<}4 \rangle$).

Applying the precomputation algorithm, given in Figure 5, yields the regions $\langle \mathrm{DI}, x{\leq}2 \wedge y{<}x{+}2 \rangle$, $\langle \mathrm{SI}, x{\leq}3 \wedge y{<}x{+}1 \rangle$, $\langle \mathrm{SR}, y{<}4 \rangle$ (the node II does not appear since the protocol can remain in this location indefinitely). The application of MinReach returns the probabilistic system given in Figure 6. By classical probabilistic reachability analysis on this system, the minimum probability of reaching SR within 4 time units of the data arriving at the sender, that is, the minimum probability of reaching $\langle \mathrm{SR}, y{<}4 \rangle$ from the region containing the state $\langle \mathrm{DI}, x{=}0 \wedge y{=}0 \rangle$ (which is given by $\langle \mathrm{DI}, x{\leq}1 \wedge x{\leq}y{<}x{+}2 \rangle$), is 0.9.

The symbolic states and edges

$$(\langle \mathrm{DI}, x{<}\ 1 \wedge y{<}x \rangle, \mathit{time}, \langle \mathrm{DI}, 1{\leq}x{\leq}2 \wedge x{-}1{\leq}y{<}1 \rangle)$$

and

$$(\langle \mathrm{SI}, x{<}\ 2 \wedge y{<}x{-}1 \rangle, \mathit{time}, \langle \mathrm{SI}, 2{\leq}x{\leq}3 \wedge x{-}2{\leq}y{<}1 \rangle)$$

are generated by the main loop of the algorithm ProbReach, while the remaining edges are added by the procedure ExtendEdges. For example, from the main algorithm there is an edge

$$(\langle \mathrm{SI}, x{\leq}3 \wedge x{-}2{\leq}y{<}x{+}1 \rangle, \mathit{time}, \langle \mathrm{SI}, 2{\leq}x{\leq}3 \wedge 1{\leq}y{<}x{+}1 \rangle)$$

and since

$$\ulcorner \langle \mathrm{SI}, x{<}2 \wedge x{-}1{\leq}y{<}x{+}1 \rangle \urcorner \subset \ulcorner \langle \mathrm{SI}, x{\leq}3 \wedge x{-}2{\leq}y{<}x{+}1 \rangle \urcorner$$

we add the edge

$$(\langle \mathrm{SI}, x{<}2 \wedge x{-}1{\leq}y{<}x{+}1 \rangle, \mathit{time}, \langle \mathrm{SI}, 2{\leq}x{\leq}3 \wedge 1{\leq}y{<}x{+}1 \rangle)\ .$$

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 $\langle \mathrm{DI}, 1{\leq}x{\leq}2 \wedge x{-}1{\leq}y{<}1 \rangle$ to $\langle \mathrm{SR}, y{<}4 \rangle$, and to $\langle \mathrm{SI}, x{<}2 \wedge x{-}1{\leq}y{<}x{+}1 \rangle$, respectively. Therefore, the distribution associated with the symbolic state $\langle \mathrm{DI}, 1{\leq}x{\leq}2 \wedge x{-}1{\leq}y{<}1 \rangle$ shown in Figure 6 is constructed.

4 Conclusions

The state space exploration algorithm presented in Section 3 for calculating the minimal reachability probability iterates predecessor, intersection and set difference operations. This differs from the algorithm presented in [KNS01] for calculating maximal reachability probabilities where only predecessor and intersection operations are required. Note that the algorithm could be applied

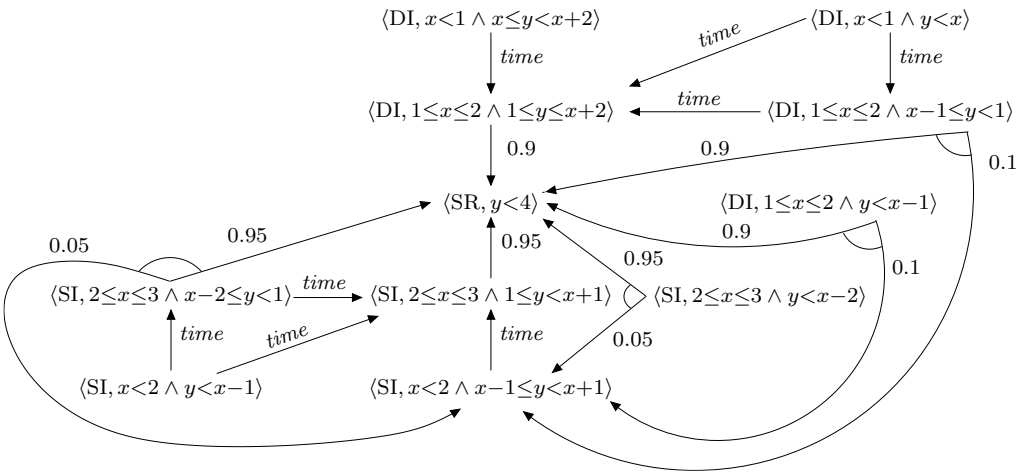

Figure 6: The system generated by MinReach for the PTA in Figure 1.

only to the reachable portion of the state space, thereby avoiding analysis of unreachable states. Furthermore, the practical implementation of our approach can be tailored to the model in question.

Together with [KNS01] our method extends to enable the verification of symbolic probabilistic systems against probabilistic temporal logics such as PCTL [BdA95, BK98].

References

[ACH+95] 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.

[AD94] R. Alur and D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[BdA95] A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In P. Thiagarajan, editor, *Proc. 15th Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 1026 of *LNCS*, pages 499–513. Springer, 1995.

[BHHK00] C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen. Model checking continuous-time Markov chains by transient analysis. In E. Emerson and A. Sistla, editors, *Proc. 12th International Conference on Computer Aided Verification (CAV'00)*, volume 1855 of *LNCS*, pages 358–372. Springer, 2000.

[BK98] C. Baier and M. Z. Kwiatkowska. Model checking for a probabilistic branching time logic with fairness. *Distributed Computing*, 11(3):125–155, 1998.

[BS02] N. Bertrand and Ph. Schnoebelen. Model checking lossy channels systems is probably decidable. Research Report LSV-02-16, Lab. Specification and Verification, ENS de Cachan, November 2002.

[DGJP00] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden. Approximating labeled Markov processes. In *Proc. 15th Annual IEEE Symposium on Logic in Computer Science*, pages 95–106. IEEE Computer Society Press, 2000.

[DJJL01] P. D'Argenio, B. Jeannet, H. Jensen, and K. Larsen. Reachability analysis of probabilistic systems by successive refinements. In L. de Alfaro and S. Gilmore, editors, *Proc. 1st Joint International Workshop on Process Algebra and Probabilistic Methods, Performance Modeling and Verification (PAPM/PROBMIV'01)*, volume 2165 of *LNCS*, pages 39–56. Springer, 2001.

[HJ94] H. Hansson and B. Jonsson. A logic for reasoning about time and probability. *Formal Aspects of Computing*, 6(5):512–535, 1994.

[HMP92] T. Henzinger, Z. Manna, and A. Puneli. What good are digital clocks? In W. Kuich, editor, *Proc. 19th International Colloquium, Automata, Languages and Programming (ICALP'92)*, volume 623 of *LNCS*, pages 545–558. Springer, 1992.

[HMR01] 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.

[HNSY94] 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.

[KNS01] M. Kwiatkowska, G. Norman, and J. Sproston. Symbolic computation of maximal probabilistic reachability. In K. Larsen and M. Nielsen, editors, *Proc. 13th International Conference on Concurrency Theory (CONCUR'01)*, volume 2154 of *LNCS*, pages 169–183. Springer, 2001.

[KNS02] M. Kwiatkowska, G. Norman, and J. Sproston. Probabilistic model checking of deadline properties in the IEEE 1394 FireWire root contention protocol. *Special Issue of Formal Aspects of Computing*, 2002. To appear.

[KNSS00] M. Kwiatkowska, G. Norman, R. Segala, and J. Sproston. Verifying quantitative properties of continuous probabilistic timed automata. In C. Palamidessi, editor, *Proc. CONCUR 2000 - Concurrency Theory*, volume 1877 of *Lecture Notes in Computer Science*, pages 123–137. Springer, 2000.

[KNSS02] M. Kwiatkowska, G. Norman, R. Segala, and J. Sproston. Automatic verification of real-time systems with discrete probability distributions. *Theoretical Computer Science*, 282:101–150, 2002.

[KSK76] J. G. Kemeny, J. L. Snell, and A. W Knapp. *Denumerable Markov Chains*. Graduate Texts in Mathematics. Springer, 2nd edition, 1976.

[Spr00] J. Sproston. Decidable model checking of probabilistic hybrid automata. In M. Joseph, editor, *Proc. Int. Symp. on Formal Techniques in Real-Time and Fault Tolerant Systems (FTRTFT'00)*, volume 1926 of *LNCS*, pages 31–45. Springer, 2000.

[Spr01] J. Sproston. *Model Checking of Probabilistic Timed and Hybrid Systems*. PhD thesis, University of Birmingham, 2001.

[Var85] M. Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proc. 26th Annual Symposium on Foundations of Computer Science (FOCS'85)*, pages 327–338. IEEE Computer Society Press, 1985.

Appendix: Proof of Theorem 7.

In this section we fix a symbolic probabilistic system $\mathbb{P}$ and target region F and show that the minimal reachability probabilities for the symbolic probabilistic system $\mathbb{P}$ and the finite sub-probabilistic system $\mathbb{Q}$ agree. First we require the following definitions and lemmas.

Definition 11 *For any adversary $A \in Adv$, $F \in R$ and finite path ω let:*

$$\mathbf{P}_0^A(\omega, F) = \begin{cases} 1 & \text{if } last(\omega) \in \ulcorner F \urcorner \\ 0 & \text{otherwise} \end{cases}$$

and for any $n \geq 0$, if $A(\pi) = p$:

$$\mathbf{P}_{n+1}^A(\omega, F) = \begin{cases} 1 & \text{if } last(\omega) \in \ulcorner F \urcorner \\ \sum_{s' \in S} p(s') \cdot \mathbf{P}_n^A(\omega \xrightarrow{p} s', F) & \text{otherwise.} \end{cases}$$

Note, in the case when $\mathbb{S}$ corresponds to $\mathbb{Q}$ we replace $last(\omega) \in \ulcorner F \urcorner$ with $last(\omega) \subseteq F$.

Lemma 12 *For any $s \in S$:*

$$MinReach(s, F) = \inf_{A \in Adv} \lim_{n \to \infty} \mathbf{P}_n^A(s, F).$$

Proof. The lemma is proved by showing for any $s \in S$ and $C \in Adv_{\mathbb{S}}$:

$$Prob\{\omega \,|\, \omega \in Pathful^C(s) \text{ and } \omega(i) \in U \text{ for some } i \geq 0\} = \lim_{n \to \infty} \mathbf{P}_n^C(s, U)$$

which follows from the fact that we can associate with C a Markov chain whose states are finite paths of $\mathbb{S}$ and the iterative method for PCTL until formulas for Markov chains [HJ94]. □

Lemma 13 *For all $\sigma \in T \setminus F$, $B \in Adv_{\mathbb{Q}}$, if $B(\sigma) = \pi$ and, E_π and $\nu \in \mathsf{D}$ are the set of edges and distribution template used to construct π, then for all $n \in \mathbb{N}$:*

$$\mathbf{P}_{n+1}^B(\sigma, F) = \sum_{(\sigma, a, \tau) \in E_\mu} \nu(a) \cdot \mathbf{P}_n^B(\sigma \xrightarrow{\pi} \tau, F).$$

Proof. Consider any $\sigma \in T$ and $B \in Adv_{\mathbb{Q}}$, if $B(\sigma) = \pi$ and π is constructed from is the set of edges E_π and distribution template ν, then by definition for any $\tau \in T$:

$$\pi(\tau) = \sum_{a \in \mathcal{T}ra \wedge (\sigma, a, \tau) \in E_\pi} \nu(a). \tag{2}$$

Now by Definition 11 we have:

$$\begin{aligned}
\mathbf{P}^B_{n+1}(\sigma, F) &= \sum_{\tau \in T} \pi(\tau) \cdot \mathbf{P}^B_n(\sigma \xrightarrow{\pi} \tau, F) \\
&= \sum_{\tau \in T} \left(\sum_{a \in \mathcal{T}ra \wedge (\sigma,a,\tau) \in E_\pi} \nu(a) \right) \cdot \mathbf{P}^B_n(\sigma \xrightarrow{\pi} \tau, F) \quad \text{by (2)} \\
&= \sum_{\tau \in T} \left(\sum_{a \in \mathcal{T}ra \wedge (\sigma,a,\tau) \in E_\pi} \nu(a) \cdot \mathbf{P}^B_n(\sigma \xrightarrow{\pi} \tau, F) \right) \quad \text{rearranging} \\
&= \sum_{(\sigma,a,\tau) \in E_\pi} \nu(a) \cdot \mathbf{P}^B_n(\sigma \xrightarrow{\pi} \tau, F)
\end{aligned}$$

as required.

Lemma 14 *For any transition type $a \in \mathcal{T}ra$, if $(\sigma, a, \tau) \in E$, then $\ulcorner \sigma \urcorner \subseteq \ulcorner \mathsf{pre}_a(\tau) \urcorner$.*

Proof. The proof follows from the construction of the edges E in the algorithm MinReach. □

Lemma 15 *For any $s \in S$, the minimum probability of reaching $\ulcorner F \urcorner$ is greater than zero if and only if there exists a unique $\sigma \in T$ such that $s \in \ulcorner \sigma \urcorner$.*

Proof. For the "if" direction: suppose there exists $\sigma \in T$ such that $s \in \ulcorner \sigma \urcorner$ (by construction σ is unique). The proof follows by induction on the length of the shortest path from s to reach $\ulcorner F \urcorner$ which only passes through states whose minimum probability of reaching $\ulcorner F \urcorner$ is greater than zero.

For the "only if" direction: by Proposition 6 it follows that for any $\sigma \in T$: if $s \in \ulcorner \sigma \urcorner$, then the minimum probability of reaching $\ulcorner F \urcorner$ is greater than zero. The fact that σ is unique follows from the construction. □

Lemma 16 *If $s, t \in S$ and $a \in \mathcal{T}ra$ such that $t \in \delta_a(s)$, and $s \in \ulcorner \sigma \urcorner$ and $t \in \ulcorner \tau \urcorner$ for some $\sigma, \tau \in T$, then $(\sigma, a, \tau) \in E$.*

Proof. Consider any $s, t \in S$ and $a \in \mathcal{T}ra$ such that $t \in \delta_a(s)$, and $s \in \ulcorner \sigma \urcorner$ and $t \in \ulcorner \tau \urcorner$ for some $\sigma, \tau \in T$. First by Lemma 15 the minimum probability of reaching $\ulcorner F \urcorner$ from s is greater than zero, and hence, $s \in \ulcorner T_{\min} \urcorner$. Therefore, since $t \in \delta_a(s)$ we have $s \in \ulcorner \mathsf{pre}_a(\tau) \cap T_{\min} \urcorner = \ulcorner \mathsf{pre}_a(\tau, T_{\min}) \urcorner$, and it follows that $(\sigma, a, \tau) \in E$ as required. □

Proof of Theorem 7. The main step in the proof involves showing a correspondence between the probability values of $\mathbf{P}^A_n$ for adversaries A of the infinite state probabilistic system $\mathbb{P}$ and $\mathbf{P}^B_n$ for adversaries B of the constructed probabilistic system $\mathbb{Q}$. After which the result follows from Lemma 12.

Formally, we will show the following correspondence: For all $n \in \mathbb{N}$, $s \in S$ such that $\mathit{MinReach}(s, \ulcorner F \urcorner) > 0$:

(a) if $B \in \mathit{Adv}_{\mathbb{Q}}$, $\sigma \in T$ and $s \in \ulcorner \sigma \urcorner$, then there exists $A \in \mathit{Adv}_{\mathbb{P}}$ such that $\mathbf{P}^A_n(s, \ulcorner F \urcorner) \leq \mathbf{P}^B_n(\sigma, F)$

(b) if $A \in \mathit{Adv}_{\mathbb{P}}$, then there exists $\sigma \in T$ and $B \in \mathit{Adv}_{\mathbb{Q}}$ such that $s \in \ulcorner \sigma \urcorner$ and $\mathbf{P}_n^B(\sigma, F) \leq \mathbf{P}_n^A(s, \ulcorner F \urcorner)$.

We now prove **(a)** and **(b)** by induction on $n \in \mathbb{N}$. In the case for $n = 0$ for both **(a)** and **(b)** follow from Definition 11.

Next, suppose **(a)** and **(b)** hold for some $n \in \mathbb{N}$ and consider any $s \in S$ such that $\mathit{MinReach}(s, \ulcorner F \urcorner) > 0$. If $s \in \ulcorner F \urcorner$, then the properties **(a)** and **(b)** follow from Definition 11. Therefore we are left with the case when $s \notin \ulcorner F \urcorner$.

(a) Consider any $B \in \mathit{Adv}_{\mathbb{Q}}$ and $\sigma \in T$ such that $s \in \ulcorner \sigma \urcorner$. Now $B(\sigma) = \pi$ for some $\pi \in \mathit{Steps}_{\mathbb{Q}}(\sigma)$ and by construction of $\mathbb{Q}$ there exists $\nu \in \mathsf{D}$ and $E_\pi \subseteq E$ such that

- if $(\sigma', a, \tau') \in E_\pi$, then $\sigma' = \sigma$
- if $(\sigma, a, \tau), (\sigma, a', \tau') \in E_\pi$ are distinct, then $a \neq a'$
- if $(\sigma, a, \tau) \in E$, then $(\sigma, a, \tau') \in E_\pi$ for some $\tau' \in T$
- for all $\tau \in T$:

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

Let $\mathcal{T}\mathit{ra}(E_\pi) = \{a \,|\, (\sigma, a, \tau) \in E_\pi \text{ for some } \tau \in T\}$. We will now construct a vector of states $\langle t_a \rangle_{a \in \mathcal{T}\mathit{ra}(s)} \in \prod_{a \in \mathcal{T}\mathit{ra}(s)} \delta_a(s)$. Considering any $a \in \mathcal{T}\mathit{ra}(s)$, we have the following two cases to consider.

- $a \in \mathcal{T}\mathit{ra}(E_\pi)$, then by definition $(\sigma, a, \tau) \in E_\pi$ for some $\tau \in T$, and hence $\sigma \subseteq \mathsf{pre}_a(\tau)$ by Lemma 14. Now, since $s \in \ulcorner \sigma \urcorner$, it follows by definition of pre_a that $\ulcorner \tau \urcorner \subseteq \delta_a(s)$. Letting t_a be any $t \in \ulcorner \tau \urcorner$, by induction (since $\tau \in T$) there exists an adversary A_a such that:

$$\mathbf{P}_n^{A_a}(t_a, \ulcorner F \urcorner) \leq \mathbf{P}_n^{B'}(\tau, F) = \mathbf{P}_n^B(\sigma \xrightarrow{\pi} \tau, F) \tag{3}$$

  where $B' \in \mathit{Adv}_{\mathbb{Q}}$ is such that $B'(\omega) = B(\sigma \xrightarrow{\mu} \omega)$ for any path ω.

- $a \notin \mathcal{T}\mathit{ra}(E_\pi)$, in this case we will prove that $\mathit{MinReach}(t, \ulcorner F \urcorner) = 0$ for all $t \in \delta_a(s)$. Therefore, suppose, for a contradiction, that there exists $t \in \delta_a(s)$ such that $\mathit{MinReach}(t, \ulcorner F \urcorner) > 0$, by Lemma 15 there exists $\tau \in T$ such that $t \in \ulcorner \tau \urcorner$, and hence applying Lemma 16 there exists an edge $(\sigma, a, \tau) \in E$ which contradicts the fact that $(\sigma, a, \tau') \notin E_\pi$ for any $\tau' \in T$. Therefore $\mathit{MinReach}(t, \ulcorner F \urcorner) = 0$ for all $t \in \delta_a(s)$, and letting $t_a \in \delta_a(s)$ be arbitrary, we have for any adversary A_a:

$$\mathbf{P}_n^{A_a}(t_a, \ulcorner F \urcorner) = 0 \; . \tag{4}$$

Note that, for any $a, b \in \mathcal{T}\mathit{ra}$ such that $a \in \mathcal{T}\mathit{ra}(E_\pi)$ and $b \notin \mathcal{T}\mathit{ra}(E_\pi)$ we have $t_a \neq t_b$. Now, since $t_a \in \delta_a(s)$ for all $a \in \mathcal{T}\mathit{ra}(s)$, by definition of the distribution templates, there exists $\mu \in \mathit{Steps}_{\mathbb{P}}(s)$ such that for all $t \in S$:

$$\mu(t) = \sum_{a \in \mathcal{T}\mathit{ra}(s) \wedge t = t_a} \nu(a). \tag{5}$$

Now suppose $A \in Adv_{\mathbb{P}}$ is the adversary that chooses μ in state s and then behaves[1] like A_a once it reaches the state t_a, by Definition 11 we have:

$$\begin{aligned}
\mathbf{P}^A_{n+1}(s, \ulcorner F \urcorner) &= \sum_{s' \in S} \mu(s') \cdot \mathbf{P}^A_n(s \xrightarrow{\mu} s', \ulcorner F \urcorner) \\
&= \sum_{t \in \{t_a \mid a \in \mathcal{T}ra(s)\}} \left(\sum_{\substack{a \in \mathcal{T}ra(E_\pi) \\ \wedge t_a = t}} \nu(a) \right) \cdot \mathbf{P}^A_n(s \xrightarrow{\mu} t, \ulcorner F \urcorner) && \text{by (5)} \\
&= \sum_{t \in \{t_a \mid a \in \mathcal{T}ra(E_\pi)\}} \left(\sum_{\substack{a \in \mathcal{T}ra(E_\pi) \\ \wedge t_a = t}} \nu(a) \right) \cdot \mathbf{P}^A_n(s \xrightarrow{\mu} t, \ulcorner F \urcorner) && \text{by (4)} \\
&= \sum_{t \in \{t_a \mid a \in \mathcal{T}ra(E_\pi)\}} \left(\sum_{\substack{a \in \mathcal{T}ra(E_\pi) \\ \wedge t_a = t}} \nu(a) \cdot \mathbf{P}^A_n(s \xrightarrow{\mu} t, \ulcorner F \urcorner) \right) && \text{rearranging} \\
&= \sum_{(\sigma, a, \tau) \in E_\pi} \nu(a) \cdot \mathbf{P}^A_n(s \xrightarrow{\mu} t_a, \ulcorner F \urcorner) && \text{by definition of } \mathcal{T}ra(E_\pi) \\
&\leq \sum_{(\sigma, a, \tau) \in E_\pi} \nu(a) \cdot \mathbf{P}^B_n(\sigma \xrightarrow{\pi} \tau, F) && \text{by (3)} \\
&= \mathbf{P}^B_{n+1}(\sigma, F) && \text{by Lemma 13}
\end{aligned}$$

and since $B \in Adv_{\mathbb{Q}}$ and $\sigma \in T$ are arbitrary, **(a)** holds by induction.

(b) Consider any adversary $A \in Adv_{\mathbb{P}}$, then $A(s) = \mu$ for some $\mu \in Steps_{\mathbb{P}}(s)$. By definition there exists a distribution template $\nu \in \mathsf{D}$ and 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 for all $s' \in S$:

$$\sum_{a \in \mathcal{T}ra(s) \wedge s' = t_a} \nu(a) = \mu(s'). \tag{6}$$

Now, for any $t \in S$ such that $\mu(t) > 0$ we have the following cases to consider.

- $MinReach(t, F) > 0$ (and hence $\mathbf{P}^A_n(a \xrightarrow{\mu} t, \ulcorner F \urcorner) > 0$) then by induction, there exists $\tau_t \in T$ and an adversary $B_t \in Adv_{\mathbb{Q}}$ such that $t \in \ulcorner \tau_t \urcorner$ and

  $$\mathbf{P}^{B_t}_n(\tau_t, F) \leq \mathbf{P}^{A'}_n(t, \ulcorner F \urcorner) = \mathbf{P}^A_n(s \xrightarrow{\mu} t, \ulcorner F \urcorner) \tag{7}$$

  where $A' \in Adv_{\mathbb{P}}$ is the adversary such that $A'(\omega) = A(s \xrightarrow{\mu} \omega)$. Next, let $\mathcal{T}ra_t(s) \subseteq \mathcal{T}ra(s)$ be the set of transition types such that $a \in \mathcal{T}ra_t(s)$ if and only if $\nu(a) > 0$ and $t_a = t$. Note that, for any distinct $t, t' \in S$: $\mathcal{T}ra_t(s) \cap \mathcal{T}ra_{t'}(s) = \varnothing$.

  Now for each $a \in \mathcal{T}ra_t(s)$ by definition $t \in \delta_a(s)$, and hence by Lemma 16 we have $(\sigma, a, \tau_t) \in E$.

[1] If $t_a = t_b$ for $a \neq b$, then let A behave like A_a if $\mathbf{P}^{A_a}_n(t_a, \ulcorner F \urcorner) \leq \mathbf{P}^{A_b}_n(t_b, \ulcorner F \urcorner)$ and A_b otherwise.

- $\mathit{MinReach}(t, \ulcorner F \urcorner) = 0$ in this case we show that there does not exist an edge (σ, a, τ) for any $a \in \mathcal{T}ra$ or $\tau \in T$ such that $t \in \ulcorner \tau \urcorner$. This follows by Lemma 15, since if there exists $\tau \in T$ such that $t \in \ulcorner \tau \urcorner$ then $\mathit{MinReach}(t, \ulcorner F \urcorner) > 0$.

Now suppose that $E_\mu = \{(\sigma, a, \tau_t) \,|\, t \in S,\ \mu(s) > 0 \text{ and } a \in \mathcal{T}ra_t(s)\}$. From above it follows that:

- if $(\sigma', \tau') \in E_\mu$, then $\sigma' = \sigma$;
- if $(\sigma, a, \tau), (\sigma, a', \tau') \in E_\mu$ are distinct, then $a \neq a'$;
- for any $a \in \mathcal{T}ra$ if $(\sigma, a, \tau) \notin E_\mu$ for any τ', then $(\sigma, a, \tau) \notin E$ for any τ'.

It then follows from the construction of $\mathbb{Q}$ that there exists $\pi \in \mathit{Steps}_{\mathbb{Q}}(\sigma)$ such that for all $\tau \in T$:

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

Now suppose that B is the adversary which chooses π in σ and for all $t \in S$ such that $\mu(t) > 0$ and $\mathbf{P}^A_n(a \xrightarrow{\mu} t, \ulcorner F \urcorner) > 0$ behaves[2] like B_t when it reaches the state τ_t, then by Lemma 13 and construction of π:

$$\begin{aligned}
\mathbf{P}^B_{n+1}(\sigma, F) &= \sum_{(\sigma,a,\tau) \in E_\mu} \nu(a) \cdot \mathbf{P}^B_n(\sigma \xrightarrow{\pi} \tau, F) \\
&= \sum_{\substack{t \in S \wedge \mu(t) > 0 \wedge \\ \mathit{MinReach}(t, \ulcorner F \urcorner) > 0}} \left(\sum_{a \in \mathcal{T}ra_t(s)} \nu(a) \cdot \mathbf{P}^B_n(\sigma \xrightarrow{\pi} \tau_t, F) \right) && \text{by construction of } E_\mu \\
&\leq \sum_{\substack{t \in S \wedge \mu(t) > 0 \wedge \\ \mathit{MinReach}(t, \ulcorner F \urcorner) > 0}} \left(\sum_{a \in \mathcal{T}ra_t(s)} \nu(a) \cdot \mathbf{P}^A_n(s \xrightarrow{\mu} t, \ulcorner F \urcorner) \right) && \text{by (7)} \\
&\leq \sum_{t \in S \wedge \mu(t) > 0} \left(\sum_{a \in \mathcal{T}ra_t(s)} \nu(a) \right) \cdot \mathbf{P}^A_n(s \xrightarrow{\mu} t, \ulcorner F \urcorner) && \text{rearranging} \\
&= \sum_{t \in S \wedge \mu(t) > 0 \wedge} \left(\sum_{\substack{a \in \mathcal{T}ra(s) \wedge t = t_a \\ \wedge \nu(a) > 0}} \nu(a) \right) \cdot \mathbf{P}^A_n(s \xrightarrow{\mu} t, \ulcorner F \urcorner) && \text{by construction of } \mathcal{T}ra_t(s) \\
&= \sum_{t \in S \wedge \mu(t) > 0} \mu(t) \cdot \mathbf{P}^A_n(s \xrightarrow{\mu} t, \ulcorner F \urcorner) && \text{by (6)} \\
&= \mathbf{P}^A_{n+1}(s, \ulcorner F \urcorner) && \text{by Definition 11}
\end{aligned}$$

as required. □

[2] If $\tau_t = \tau_{t'}$ for $t \neq t'$, then let A behave like A_t if $\mathbf{P}^{B_t}_n(\tau_t, F) \geq \mathbf{P}^{B_{t'}}_n(\tau_{t'}, F)$ and $B_{t'}$ otherwise.