Symbolic Model Checking for Probabilistic Timed Automata [⋆]

Marta Kwiatkowska [a], Gethin Norman [a], Jeremy Sproston [b,⋆⋆], Fuzhi Wang [a]

[a]*School of Computer Science, University of Birmingham, Edgbaston, Birmingham B15 2TT, United Kingdom*

[b]*Dipartimento di Informatica, Università di Torino, 10149 Torino, Italy*

Abstract

Probabilistic timed automata are timed automata extended with discrete probability distributions, and can be used to model timed randomised protocols or fault-tolerant systems. We present symbolic model-checking algorithms for probabilistic timed automata to verify both qualitative temporal logic properties, corresponding to satisfaction with probability 0 or 1, and quantitative properties, corresponding to satisfaction with arbitrary probability. The algorithms operate on zones, which represent sets of valuations of the probabilistic timed automaton's clocks. Our method considers only those system behaviours which guarantee the divergence of time with probability 1. The paper presents a symbolic framework for the verification of probabilistic timed automata against the probabilistic, timed temporal logic PTCTL. We also report on a prototype implementation of the algorithms using Difference Bound Matrices, and present the results of its application to the CSMA/CD and FireWire root contention protocol case studies.

1 Introduction

Systems exhibiting both *timed* and *probabilistic* characteristics are widespread, in application contexts as diverse as home entertainment, medicine and business. For example, timing constraints are often vital to the correctness of embedded digital technology, whereas probability exhibits itself commonly in the

[⋆] Supported in part by the EPSRC grants GR/N22960, GR/S46727, FORWARD and the FIRB-Perf project of the Italian Ministry of Research.

[⋆⋆]Corresponding author. *Tel.:* +39-011-6706772; *Fax.:* +39-011-751603.
Email address: sproston@di.unito.it (Jeremy Sproston)

form of statistical estimates regarding the environment in which a system is embedded. Similarly, protocol designers often exploit the combination of time and probability to design correct, efficient protocols, such as the IEEE1394 FireWire root contention protocol. The diffusion of such systems has led to methods for obtaining formal correctness guarantees; for instance, adaptations of model checking [1]. *Symbolic model checking* refers to model-checking techniques in which implicit representations – such as Binary Decision Diagrams [2] – are used to represent both the transition relation of the system model and the state sets that are computed during the execution of the model-checking algorithm.

In this paper, we consider the modelling formalism of *probabilistic timed automata* [3,4,5], an extension of timed automata [6,7] with discrete probability distributions. Probabilistic timed automata have been shown as being suitable for the description of timed, randomized protocols, such as the aforementioned FireWire protocol [8], the backoff strategy of the IEEE802.11 WLAN protocol [9], and the link-local address selection protocol of the IPv4 standard [10]. As a requirement specification language for probabilistic timed automata we consider PTCTL (Probabilistic Timed Computation Tree Logic). The logic PTCTL combines the probabilistic threshold operator of the probabilistic temporal logic PCTL [11] with the timing constraints of the timed temporal logic TCTL [12,7], in order to express properties such as the probabilistic deadline property 'with probability 0.99 or greater, the system reaches a leader-elected state within 1 second'. Model checking of probabilistic timed automata against PTCTL was shown to be decidable in [3] via an adaptation of the classical region-graph construction [6,12].

Unfortunately, the region-graph construction (and the integer-time semantics employed in [8,9,10]) can result in huge state spaces if the maximal constant used in the description of the automaton is large. Instead, the practical success of *symbolic*, *zone*-based techniques for non-probabilistic timed automata, as implemented in the tools Uppaal [13] and Kronos [14], suggests that a similar symbolic approach may also be employed for the verification of probabilistic timed automata. We answer this hypothesis affirmatively in this paper by providing zone-based algorithms for the verification of PTCTL. As is standard in model-checking methods for branching-time logics such as PCTL and TCTL, the algorithms are based on backwards search through the state space by iterating successively predecessor relations which, given a state set Z, return the set of states which can reach states in Z in one transition. This differs from the *forwards reachability* approach employed in [3,15] for verifying probabilistic timed automata, which, unlike the approach presented in this paper, leads to only approximate results and is only applicable to a subset of PTCTL.

Our approach is to consider two classes of PTCTL properties: on the one hand, *qualitative* PTCTL formulae refer to probabilistic thresholds 0 and 1 only, whereas, on the other hand, *quantitative* PTCTL formulae feature arbitrary probability thresholds. The two classes involve different types of algorithms; in particular, the algorithms for qualitative properties require only graph-based analysis, and do not refer to exact transition probabilities, avoiding potentially expensive numerical computation during the model-checking process.

We first consider the subset of PTCTL which requires the computation of maximum reachability probabilities. For qualitative formulae, we show that model checking can be performed using a combination of the algorithm developed for verifying analogous properties of finite-state probabilistic systems [16] and the algorithm for computing the existence of a path satisfying a temporal logic formula in non-probabilistic timed automata [7]. More precisely, our algorithm comprises iteration of timed-predecessor and discrete-predecessor operations. The timed-predecessor operation maps a state set Z to the set of states which can reach Z by letting time elapse; the discrete-predecessor operation maps a state set Z and an edge e of the graph of the probabilistic timed automaton to the set of states which can reach Z by crossing the edge e. The case of quantitative formulae is more complicated, because a simple iteration of timed-predecessor and discrete-predecessor operations does not suffice to compute the probabilities with which a state satisfies a temporal logic formula. Our approach instead is to iterate timed-predecessor, discrete-predecessor and *intersection* operations until a fix-point is reached. The role of the intersection operations is to characterise the set of states from which *multiple edges* within the support of the *same distribution* of the probabilistic timed automaton can be used to reach previously generated state sets. Upon termination of the fix-point algorithm, the set of generated state sets is used to construct a finite-state probabilistic system which has sufficient information to compute the maximum reachability probability of interest using well-established finite-state probabilistic model checking methods [17].

Secondly, we consider algorithms for the subset of PTCTL which requires the computation of minimum reachability probabilities. In order to verify properties of real-world timed behaviour, it is vital that such algorithms incorporate a notion of *time divergence*. For example, to compute the minimum probability of reaching a certain state set F, for any state other than those in F, the probabilistic timed automaton could exhibit behaviour in which the amount of time elapsed converges before F is reached, or even in which no time elapses at all. Clearly, such behaviours are pathological, and should be disregarded during model checking. We present both qualitative and quantitative algorithms for computing minimum reachability probabilities which consider *only time-divergent behaviour*, based on the non-probabilistic precedent of [7]. The algorithms are based on computing maximum probabilities for the dual formula while restricting attention to time-divergent behaviours. Note that letting time converge can only make the reachability of a state set less probable, and therefore we do not need to consider time-divergence explicitly when formulating algorithms for maximum reachability probabilities.

Again following the precedent of [7], we present an algorithm to check that a probabilistic timed automaton does not contain a state in which it is impossible for time to diverge with probability 1. The presence of such a state constitutes a modelling error, and would invalidate the correctness of our model-checking procedure. Finally, we report on a prototype implementation of the techniques of this paper using Difference Bound Matrices (DBMs) [18]. We apply this implementation to two case studies: the first concerns the IEEE802.3 CSMA/CD (Carrier Sense, Multiple Access with Collision Detection) communication protocol [19], whereas the second considers the IEEE1394 FireWire root contention protocol [20].

The paper proceeds as follows. We review a number of preliminary concepts in Section 2, whereas in Section 3 we revisit the definition of probabilistic timed automata and PTCTL. In Section 4, we introduce the algorithms for qualitative and quantitative properties, referring to both the maximum and the minimum probability of satisfaction. Section 5 summarises our prototype implementation and the application of it to the case studies. In Section 6, we conclude the paper. A preliminary version of this work appeared as [21].

2 Preliminaries

We present a number of preliminary concepts, in particular defining three (increasingly general) kinds of probabilistic transition systems, the last of which will be used for the semantics of probabilistic timed automata in Section 3.

2.1 Distributions

A (discrete probability) *distribution* over a finite set Q is a function $\mu : Q \to [0,1]$ such that $\sum_{q \in Q} \mu(q) = 1$. Let $\mathsf{support}(\mu)$ be the subset of Q such that $q \in \mathsf{support}(\mu)$ if and only if $\mu(q) > 0$. Given $Q' \subseteq Q$, we let $\mu(Q') = \sum_{q \in Q'} \mu(q)$. For any $q \in Q$, the *point distribution* μ_q denotes the distribution which assigns probability 1 to q. For a possibly uncountable set Q_∞, let $\mathsf{Dist}(Q_\infty)$ be the set of distributions over finite subsets of Q_∞.

2.2 Discrete-Time Markov Chains

In this section, we recall the definition of discrete-time Markov chains, and the way in which probability measures can be defined over their behaviour.

Let AP be a fixed finite set of atomic propositions.

Definition 1 *A* (labelled) Discrete-Time Markov Chain (DTMC) *is a tuple* $\mathsf{DTMC} = (S, \mathbf{P}, \mathcal{L})$ *where:*

- S *is a (countable) set of* states*;*
- $\mathbf{P} : S{\times}S \to [0, 1]$ *is a* transition probability matrix*, such that* $\sum_{s' \in S} \mathbf{P}(s, s') = 1$ *for all states* $s \in S$*;*
- $\mathcal{L} : S \to 2^{AP}$ *is a* labelling function *which assigns to each state* $s \in S$ *the set* $\mathcal{L}(s)$ *of atomic propositions that are valid in* s*.*

Each element $\mathbf{P}(s, s')$ of the transition probability matrix gives the probability of making a transition from state s to state s'. An execution of a DTMC is represented by a finite or infinite *path* ω. A finite path is a finite, non-empty sequence of states $s_0 s_1 \ldots s_n$ such that $\mathbf{P}(s_i, s_{i+1}){>}0$ for all $0 \leqslant i < n$. Similarly, an infinite path is an infinite sequence of states $s_0 s_1 s_2 \ldots$ such that $\mathbf{P}(s_i, s_{i+1}){>}0$ for all $i \geqslant 0$. The length of a finite path, denoted by $|s_0 s_1 \cdots s_n|$, is n (the number of transitions of the path), whereas the length of an infinite path is ∞. For any path ω and any $i \leqslant |\omega|$, we denote by $\omega(i)$ the $(i{+}1)$th state of ω. The last state of a finite path ω is denoted by $\mathit{last}(\omega)$. We say that a finite path ω_{fin} of length n is a *prefix* of an infinite path ω if $\omega_{\mathit{fin}}(i){=}\omega(i)$ for $0 \leqslant i \leqslant n$. The sets of all finite and infinite paths starting in state s are denoted $\mathit{Path}_{\mathit{fin}}(s)$ and $\mathit{Path}_{\mathit{ful}}(s)$, respectively.

To reason about the probabilistic behaviour of the DTMC, we need to determine the probability with which certain paths are taken. This is achieved by defining, for each state $s \in S$, a probability measure Prob_s over $\mathit{Path}_{\mathit{ful}}(s)$. Below, we give an outline of this construction. For further details, see [22]. The probability measure is induced by the transition probability matrix $\mathbf{P}$ as follows. First, for any finite path $\omega_{\mathit{fin}} \in \mathit{Path}_{\mathit{fin}}(s)$ such that $|\omega_{\mathit{fin}}|{=}n$, we define the probability $\mathbf{P}_s(\omega_{\mathit{fin}})$ as follows:

$$\mathbf{P}_s(\omega_{\mathit{fin}}) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } n = 0 \\ \mathbf{P}(\omega_{\mathit{fin}}(0), \omega_{\mathit{fin}}(1)) \cdots \mathbf{P}(\omega_{\mathit{fin}}(n{-}1), \omega_{\mathit{fin}}(n)) & \text{otherwise} \end{cases}$$

Next, we define the *cylinder* of a finite path ω_{fin} as:

$$C(\omega_{\mathit{fin}}) \stackrel{\text{def}}{=} \{\omega \in \mathit{Path}_{\mathit{ful}}(s) \,|\, \omega_{\mathit{fin}} \text{ is a prefix of } \omega\},$$

and let Σ_s be the smallest σ-algebra on $\mathit{Path}_{\mathit{ful}}(s)$ which contains the cylinders $C(\omega_{\mathit{fin}})$ for $\omega_{\mathit{fin}} \in \mathit{Path}_{\mathit{fin}}(s)$. Finally, we define Prob_s on Σ_s as the unique measure such that $\mathit{Prob}_s(C(\omega_{\mathit{fin}})) = \mathbf{P}_s(\omega_{\mathit{fin}})$ for all $\omega_{\mathit{fin}} \in \mathit{Path}_{\mathit{fin}}(s)$.

Next, we present a form of transition system which combines probabilistic choice, as in Markov chains, with *nondeterministic* choice. We refer to such systems simply as *probabilistic systems*, and note that they are essentially equivalent to Markov decision processes [23] and probabilistic-nondeterministic systems [17].

Definition 2 *A* probabilistic system*,* PS*, is a tuple* $(S, \mathit{Steps}, \mathcal{L})$ *where*

- S *is a set of* states*;*
- $\mathit{Steps} \subseteq S \times \mathsf{Dist}(S)$ *is a* probabilistic transition relation*;*
- $\mathcal{L} : S \to 2^{AP}$ *is a* labelling function *assigning atomic propositions to states.*

We assume that the probabilistic transition relation is *total*; that is, for every state $s \in S$, there exists $(s, \mu) \in \mathit{Steps}$ for some $\mu \in \mathsf{Dist}(S)$. Occasionally we omit the labelling condition from the definition of probabilistic systems, and write (S, Steps).

A *probabilistic transition* $s \xrightarrow{\mu} s'$ is made from a state s by nondeterministically selecting a distribution $\mu \in \mathsf{Dist}(S)$ such that $(s, \mu) \in \mathit{Steps}$, and then making a probabilistic choice of target state s' according to μ, such that $\mu(s'){>}0$.

We consider two ways in which a probabilistic system's computation may be represented. A *path*, representing a particular resolution of both nondeterminism *and* probability, is a non-empty finite or infinite sequence of transitions:

$$\omega = s_0 \xrightarrow{\mu_0} s_1 \xrightarrow{\mu_1} s_2 \xrightarrow{\mu_2} \cdots \ .$$

We use the same notation for the length, $(i{+}1)$th state and prefix of paths of probabilistic systems as that used for paths of DTMCs as presented in Section 2.2; in particular, the set of infinite (respectively, finite) paths starting in the state s are denoted by $\mathit{Path}_{\mathit{ful}}(s)$ (respectively, $\mathit{Path}_{\mathit{fin}}(s)$). Furthermore, for the finite path ω_{fin}, the finite or infinite path ω, and the distribution μ such that $(\mathit{last}(\omega_{\mathit{fin}}), \mu) \in \mathit{Steps}$ and $\mu(\omega(0)){>}0$, we write $\omega_{\mathit{fin}} \xrightarrow{\mu} \omega$ for the concatenation of ω_{fin} and ω via the transition $\mathit{last}(\omega_{\mathit{fin}}) \xrightarrow{\mu} \omega(0)$.

In contrast to a path, an *adversary* represents a particular resolution of nondeterminism *only*. Formally, an adversary A is a function mapping every finite path ω_{fin} to a distribution μ such that $(\mathit{last}(\omega_{\mathit{fin}}), \mu) \in \mathit{Steps}$. For any adversary A and state s, we let $\mathit{Path}^A_{\mathit{ful}}(s)$ (respectively, $\mathit{Path}^A_{\mathit{fin}}(s)$) denote the subset of $\mathit{Path}_{\mathit{ful}}(s)$ (respectively, $\mathit{Path}_{\mathit{fin}}(s)$) induced by A. We use $\mathit{Adv}_{\mathsf{PS}}$ to denote the set of adversaries of the probabilistic system PS.

For each adversary $A \in Adv_{\mathsf{PS}}$, we can define the probability measure $Prob_s^A$ over $Path_{ful}^A(s)$. More precisely, for a probabilistic system $\mathsf{PS} = (S, Steps, \mathcal{L})$ and state $s \in S$, under a given adversary A, the behaviour from state s can be described with the (countable) infinite-state DTMC: $\mathsf{DTMC}_s^A = (S_s^A, \mathbf{P}_s^A, \mathcal{L}_s^A)$ where $S_s^A = Path_{fin}^A(s)$, for any finite paths $\omega_{fin}, \omega'_{fin} \in S_s^A$:

$$\mathbf{P}^A(\omega_{fin}, \omega'_{fin}) = \begin{cases} \mu(s') \text{ if } \omega'_{fin} \text{ is of the form } \omega_{fin} \xrightarrow{\mu} s' \text{ and } A(\omega_{fin}) = \mu \\ \;\;0 \quad \text{otherwise,} \end{cases}$$

and $\mathcal{L}_s^A(\omega_{fin}) = \mathcal{L}(last(\omega_{fin}))$ for each $\omega_{fin} \in S_s^A$. There is a one-to-one correspondence between the paths of DTMC_s^A and the paths of $Path_{ful}^A(s)$, and hence using the construction given in Section 2.2 we can define a probability measure $Prob_s^A$ over $Path_{ful}^A(s)$ [24].

We now introduce the following definitions concerning probabilistic systems which are required later in the paper. To begin we introduce the syntax and semantics for the probabilistic temporal logic PCTL [11].

Definition 3 *The syntax of* PCTL *is defined as follows:*

$$\Phi ::= a \;\Big|\; \neg\Phi \;\Big|\; \Phi \vee \Phi \;\Big|\; \mathcal{P}_{\sim\lambda}[\Phi\, \mathcal{U}\, \Phi] \;\Big|\; \mathcal{P}_{\sim\lambda}[\Phi\, \mathcal{V}\, \Phi]$$

where $a \in AP$, $\sim \in \{\leqslant, <, >, \geqslant\}$ *and* $\lambda \in [0, 1]$.

We use use the abbreviations $\Diamond\Phi$ and $\Box\Phi$ for $\mathtt{true}\, \mathcal{U}\, \Phi$ and $\mathtt{false}\, \mathcal{V}\, \Phi$ respectively. In the standard manner, we refer to $\Phi\, \mathcal{U}\, \Psi$, $\Phi\, \mathcal{V}\, \Psi$, $\Diamond\Phi$ and $\Box\Psi$ as *path formulae*.

PCTL can be used to express properties such as:

- 'with probability less than 0.01, an error state is reached', which is represented as the formula $\mathcal{P}_{<0.01}[\Diamond\ error]$, where *error* is an atomic proposition labelling the error locations;
- 'with probability greater than 0.98, the system remains operational', which is represented as the formula $\mathcal{P}_{\geqslant 0.98}[\Box\ operational]$, where *operational* is an atomic proposition labelling the states in which the system is operational.

Below we present the semantics for PCTL followed by a number of lemmas concerning PCTL required in the remainder of the paper.

Definition 4 *Let* $\mathsf{PS} = (S, Steps, \mathcal{L})$ *be a probabilistic system. For any state* $s \in S$ *and* PCTL *formula* Θ*, the satisfaction relation* $s \models \Theta$ *is defined induc-*

tively as follows:

$$\begin{aligned} s \models a &\Leftrightarrow a \in \mathcal{L}(s) \\ s \models \Phi \vee \Psi &\Leftrightarrow s \models \Phi \ \textit{ or } \ s \models \Psi \\ s \models \neg\Phi &\Leftrightarrow s \not\models \Phi \\ s \models \mathcal{P}_{\sim\lambda}[\varphi] &\Leftrightarrow p_s^A(\varphi) \sim \lambda \ \textit{ for all } \ A \in Adv_{\mathsf{PS}} \end{aligned}$$

where $p_s^A(\varphi) = Prob_s^A\{\omega \in Path_{ful}^A(s) \,|\, \omega \models \varphi\}$ *and, for any path* $\omega \in Path_{ful}^A(s)$*:*

$$\begin{aligned} \omega \models \Phi\,\mathcal{U}\,\Psi &\Leftrightarrow \exists i \in \mathbb{N}.\,(\omega(i) \models \Psi \wedge \forall j{<}i.\,\omega(j) \models \Phi) \\ \omega \models \Phi\,\mathcal{V}\,\Psi &\Leftrightarrow \forall i \in \mathbb{N}.\,((\forall j{<}i.\,\omega(j) \not\models \Phi) \rightarrow \omega(i) \models \Psi)\,. \end{aligned}$$

In the lemmas below we require an extension of the logic PCTL that allows more general path formulae, that is those obtained through the negation, conjunction and disjunction of (standard) PCTL path formulae. The semantics for such formulae follows the standard approach for such connectives, for example, for any PCTL path formulae φ, φ' and path ω:

$$\begin{aligned} \omega \models \neg\varphi &\Leftrightarrow \omega \not\models \varphi \\ \omega \models \varphi \vee \varphi' &\Leftrightarrow \omega \models \varphi \ \text{ or } \ \omega \models \varphi'\,. \end{aligned}$$

Lemma 5 *Let* $\mathsf{PS}{=}(S, Steps, \mathcal{L})$ *be a probabilistic system and* Φ *and* Ψ PCTL *formulae. For any state* $s \in S$*:*

$$\sup_{A \in Adv_{\mathsf{PS}}} p_s^A(\Phi\,\mathcal{U}\,\Psi) = \sup_{A \in Adv_{\mathsf{PS}}} p_s^A(\Phi\,\mathcal{U}\,\neg\mathcal{P}_{<1}[\Phi\,\mathcal{U}\,\Psi])\,.$$

Proof. See for example [16]. □

Lemma 6 *Let* $\mathsf{PS}{=}(S, Steps, \mathcal{L})$ *be a probabilistic system and* Φ *and* Ψ *be* PCTL *formulae. For any path* $\omega \in Path_{ful}$*:*

$$\begin{aligned} \omega \models \Phi\,\mathcal{U}\,\Psi &\Leftrightarrow \omega \not\models \neg\Phi\,\mathcal{V}\,\neg\Psi \\ \omega \models \Phi\,\mathcal{U}\,\Psi &\Leftrightarrow \omega \not\models (\neg\Psi\,\mathcal{U}\,(\neg\Psi{\wedge}\neg\Phi)) \vee \Box(\Phi{\wedge}\neg\Psi) \\ \omega \models \Phi\,\mathcal{V}\,\Psi &\Leftrightarrow \omega \models (\Psi\,\mathcal{U}\,(\Psi{\wedge}\Phi)) \vee \Box(\neg\Phi{\wedge}\Psi)\,. \end{aligned}$$

Proof. The lemma is independent of the fact that we consider probabilistic systems, and a proof can be found in for example [25]. □

Lemma 7 *Let* PS=$(S, \mathit{Steps}, \mathcal{L})$ *be a probabilistic system and* Φ, Ψ *and* Θ *be* PCTL *formulae. For any path* $\omega \in \mathit{Path}_{\mathit{ful}}$*:*

$$\omega \models (\Psi\, \mathcal{U}\, (\Psi{\wedge}\Phi)) \vee ((\neg\Phi{\wedge}\Psi)\, \mathcal{U}\, \Theta) \;\;\Leftrightarrow\;\; \omega \models \Psi\, \mathcal{U}\, ((\Phi{\wedge}\Psi) \vee \Theta)\,.$$

Proof. Consider any probabilistic system $\mathsf{PS} = (S, \mathit{Steps}, \mathcal{L})$ and path $\omega \in \mathit{Path}_{\mathit{ful}}$. For the 'if' direction suppose that $\omega \models \Psi\, \mathcal{U}\, ((\Phi{\wedge}\Psi) \vee \Theta)$. Now, by Definition 4 there exists an $i{\geqslant}0$ such that: $\omega(i) \models (\Phi{\wedge}\Psi) \vee \Theta$ and $\omega(j) \models \Psi$ for all $j{<}i$, and hence we have the following two cases to consider:

- $\omega(i) \models \Phi{\wedge}\Psi$ and $\omega(j) \models \Psi$ for all $j{<}i$, then using Definition 4 it follows that $\omega \models \Psi\, \mathcal{U}\, (\Psi{\wedge}\Phi)$.
- $\omega(i) \models \Theta$ and $\omega(j) \models \Psi$ for all $j{<}i$, then either $\omega(j) \models \neg\Phi{\wedge}\Psi$ for all $j{<}i$, and therefore $\omega \models (\neg\Phi{\wedge}\Psi)\, \mathcal{U}\, \Theta$, or $\omega(k) \models \Phi{\wedge}\Psi$ for some $k{<}i$ and since by the hypothesis $\omega(j) \models \Psi$ for all $j{<}i$, we have $\omega \models \Psi\, \mathcal{U}\, (\Psi{\wedge}\Phi)$.

Because these are the only possible cases to consider using Definition 4 the 'if' direction follows.

For the 'only if' direction suppose that $\omega \models (\Psi\, \mathcal{U}\, (\Psi{\wedge}\Phi)) \vee ((\neg\Phi{\wedge}\Psi)\, \mathcal{U}\, \Theta)$, considering the satisfaction of each disjunct separately, it is straightforward to show that $\omega \models \Psi\, \mathcal{U}\, ((\Phi{\wedge}\Psi) \vee \Theta)$ as required. □

Lemma 8 *Let* PS=$(S, \mathit{Steps}, \mathcal{L})$ *be a finite-state probabilistic system, A an adversary of* PS *and* Φ *be a* PCTL *formula. For any state* $s \in S$:

$$p_s^A(\Box\ \Phi) = 1 \;\;\Leftrightarrow\;\; \forall\, \omega \in \mathit{Path}_{\mathit{ful}}^A(s).\, \omega \models \Box\ \Phi\ .$$

Proof. The 'if' direction follows from the definition of p_s^A and the fact that Prob_s^A is a probability measure. For the 'only if' direction suppose for a contradiction that $p_s^A(\Box\ \Phi) = 1$ and there exists a path ω in $\mathit{Path}_{\mathit{ful}}^A(s)$ such that $\omega \not\models \Box\ \Phi$. Now, using Definition 4, it follows that there exists $i \in \mathbb{N}$ such that $\omega(i) \models \neg\Phi$. Letting ω_{fin} be the finite prefix of ω of length i, we have $\omega' \not\models \Box\ \Phi$ for all $\omega' \in \{\omega' \in \mathit{Path}_{\mathit{ful}}^A(s) \,|\, \omega_{\mathit{fin}}$ is a prefix of $\omega'\}$. Furthermore, from the measure construction (see Section 2.2), we have $\mathit{Prob}_s^A\{\omega' \in \mathit{Path}_{\mathit{ful}}^A(s) \,|\, \omega_{\mathit{fin}}$ is a prefix of $\omega'\} > 0$. Finally, combining these two facts we have $p_s^A(\Box\ \Phi) < 1$ which is a contradiction as required. □

Lemma 9 *Let* PS=$(S, \mathit{Steps}, \mathcal{L})$ *be a finite-state probabilistic system, A an adversary of* PS *and* Φ *be a* PCTL *formula. For any state* $s \in S$:

$$p_s^A(\Phi\, \mathcal{U}\, \neg\mathcal{P}_{<1}[\Box\, \Phi]) \geqslant p_s^A(\Box\, \Phi)\,.$$

Proof. Consider any finite-state probabilistic system $\mathsf{PS} = (S, \mathit{Steps}, \mathcal{L})$, adversary A, state s and PCTL formula Φ. First consider the adversary A' which behaves like A except when a state s' satisfying $\neg\mathcal{P}_{<1}[\Box\,\Phi]$ is reached and, in which case, acts like the adversary for which from s' the probability of satisfying $\Box\,\Phi$ is 1 (the existence of such an adversary follows from the fact that $s' \models \neg\mathcal{P}_{<1}[\Box\,\Phi]$). By the construction of A' it follows that:

$$p^{A}_{s}(\Phi\,\mathcal{U}\,\neg\mathcal{P}_{<1}[\Box\,\Phi]) = p^{A'}_{s}(\Phi\,\mathcal{U}\,\neg\mathcal{P}_{<1}[\Box\,\Phi]) \text{ and } p^{A'}_{s}(\Box\,\Phi) \geqslant p^{A}_{s}(\Box\,\Phi)\,. \quad (1)$$

Now, since any state $s' \in S$, if $s' \models \neg\mathcal{P}_{<1}[\Box\,\Phi]$, by Definition 4 and the construction of A' we have $p^{A'}_{s'}(\Box\Phi) = 1$. From Lemma 8 it follows that any state reachable from s' under the adversary A' satisfies Φ. Using this result we have that for any path ω of $\mathit{Path}^{A'}_{\mathit{ful}}(s)$: $\omega \models \Phi\,\mathcal{U}\,\neg\mathcal{P}_{<1}[\Box\,\Phi]$ implies $\omega \models \Box\,\Phi$, and therefore

$$p^{A'}_{s}(\Box\,\Phi) = p^{A'}_{s}(\Phi\,\mathcal{U}\,\neg\mathcal{P}_{<1}[\Box\,\Phi]) + p^{A'}_{s}\Big((\Box\,\Phi) \wedge \neg(\Phi\,\mathcal{U}\,\neg\mathcal{P}_{<1}[\Box\,\Phi])\Big)\,. \quad (2)$$

Now using Lemma 6 we have:

$$\begin{aligned}
& p^{A'}_{s}\Big((\Box\,\Phi) \wedge \neg(\Phi\,\mathcal{U}\,\neg\mathcal{P}_{<1}[\Box\,\Phi])\Big) \\
&= p^{A'}_{s}\Big((\Box\,\Phi) \wedge \Big((\mathcal{P}_{<1}[\Box\,\Phi]\,\mathcal{U}\,(\mathcal{P}_{<1}[\Box\,\Phi] \wedge \neg\Phi)) \vee \Box(\Phi \wedge \mathcal{P}_{<1}[\Box\,\Phi])\Big)\Big) \\
&= p^{A'}_{s}\Big(\Big((\Box\,\Phi) \wedge (\mathcal{P}_{<1}[\Box\,\Phi]\,\mathcal{U}\,(\mathcal{P}_{<1}[\Box\,\Phi] \wedge \neg\Phi))\Big) \vee \Big((\Box\,\Phi) \wedge \Box(\Phi \wedge \mathcal{P}_{<1}[\Box\,\Phi])\Big)\Big) \\
&= p^{A'}_{s}\Big((\Box\,\Phi) \wedge \Box(\Phi \wedge \mathcal{P}_{<1}[\Box\,\Phi])\Big) \\
&= p^{A'}_{s}\Big(\Box(\Phi \wedge \mathcal{P}_{<1}[\Box\,\Phi])\Big) \\
&= 1 - p^{A'}_{s}\Big(\Diamond(\neg\Phi \vee \neg\mathcal{P}_{<1}[\Box\,\Phi])\Big)
\end{aligned}$$

where the second step follows by the distributivity of conjunction over disjunction, the third step from the fact that, for any PCTL formulae Φ and Ψ, no path can satisfy the formula $(\Box\Phi) \wedge (\mathcal{P}_{<1}[\Box\,\Phi]\,\mathcal{U}\,(\mathcal{P}_{<1}[\Box\,\Phi] \wedge \neg\Phi))$, and the final two steps follow from Definition 4. From (1) and (2) to complete the proof it is sufficient to show that $p^{A'}_{s}((\Box\,\Phi) \wedge \neg(\Phi\,\mathcal{U}\,\neg\mathcal{P}_{<1}[\Box\,\Phi])) = 0$, which from above reduces to demonstrating that $p^{A'}_{s}(\Diamond(\neg\Phi \vee \neg\mathcal{P}_{<1}[\Box\,\Phi])) = 1$.

Now, for any state $s' \in S$, suppose that under A' one cannot reach a state satisfying $\neg\Phi \vee \neg\mathcal{P}_{<1}[\Box\,\Phi]$. Therefore all states reachable from s' under A' satisfy Φ, and hence Lemma 8 implies that $s' \models \neg\mathcal{P}_{<1}[\Box\,\Phi]$ which is a contradiction. Therefore, since the state $s' \in S$ was arbitrary, from any state, under A', one reaches a state satisfying $\neg\Phi \vee \neg\mathcal{P}_{<1}[\Box\,\Phi]$. Now, since S is finite, it follows that under A' the probability of reaching a state satisfying $\neg\Phi \vee \neg\mathcal{P}_{<1}[\Box\,\Phi]$ is 1, and hence $p^{A'}_{s}(\Diamond(\neg\Phi \vee \neg\mathcal{P}_{<1}[\Box\,\Phi])) = 1$ as required. □

Lemma 10 *Let* PS=$(S, \mathit{Steps}, \mathcal{L})$ *be a finite-state probabilistic system and* Φ *and* Ψ *be* PCTL *formulae. For any adversary* $A \in \mathit{Adv}_{\mathsf{PS}}$ *and state* $s \in S$:

$$p^{A}_{s}(\Phi\,\mathcal{V}\,\Psi) \leqslant p^{A}_{s}\Big(\,\Psi\,\mathcal{U}\,((\Phi{\wedge}\Psi) \vee \neg\mathcal{P}_{<1}[\Box\,(\neg\Phi{\wedge}\Psi)])\,\Big)\,.$$

Proof. Consider any finite-state probabilistic system $\mathsf{PS} = (S, Steps, \mathcal{L})$, adversary $A \in Adv_{\mathsf{PS}}$, state $s \in S$ and PCTL formulae Φ and Ψ. Using Lemma 6 we have:

$$
\begin{aligned}
p^A_s(\Phi\,\mathcal{V}\,\Psi) \;=\;& p^A_s\Big(\big(\Psi\,\mathcal{U}\,(\Psi{\wedge}\Phi)\big) \vee \Box(\neg\Phi{\wedge}\Psi)\Big) \\
=\;& p^A_s\big(\Psi\,\mathcal{U}\,(\Psi{\wedge}\Phi)\big) + p^A_s\big(\Box(\neg\Phi{\wedge}\Psi)\big) && \text{rearranging} \\
\leqslant\;& p^A_s\big(\Psi\,\mathcal{U}\,(\Psi{\wedge}\Phi)\big) + p^A_s\big((\neg\Phi{\wedge}\Psi)\,\mathcal{U}\,\neg\mathcal{P}_{<1}[\Box\,(\neg\Phi{\wedge}\Psi)]\big) && \text{by Lemma 9} \\
=\;& p^A_s\Big(\big(\Psi\,\mathcal{U}\,(\Psi{\wedge}\Phi)\big) \vee \big((\neg\Phi{\wedge}\Psi)\,\mathcal{U}\,\neg\mathcal{P}_{<1}[\Box\,(\neg\Phi{\wedge}\Psi)]\big)\Big) && \text{rearranging} \\
=\;& p^A_s\Big(\,\Psi\,\mathcal{U}\,((\Phi{\wedge}\Psi) \vee \neg\mathcal{P}_{<1}[\Box\,(\neg\Phi{\wedge}\Psi)])\,\Big) && \text{by Lemma 7}
\end{aligned}
$$

where the correctness of the rearranging steps in the derivation follow from fact that in both cases the two formulae that form the disjunction are disjoint in the sense that no path can satisfy both formulae. □

2.4 Timed Probabilistic Systems

We now introduce *timed probabilistic systems*, an extension of probabilistic systems and a variant of Segala's probabilistic timed automata [26].

Definition 11 *A* timed probabilistic system*,* TPS*, is a tuple* $(S, TSteps, \mathcal{L})$ *where:*

- S *is a (possibly infinite) set of* states*;*
- $TSteps \subseteq S \times \mathbb{R}_{\geqslant 0} \times \mathsf{Dist}(S)$ *is a* timed probabilistic transition relation*, such that, if* $(s, t, \mu) \in TSteps$ *and* $t{>}0$*, then* μ *is a point distribution;*
- $\mathcal{L} : S \to 2^{AP}$ *is a labelling function assigning atomic propositions to states.*

The component t of a tuple (s, t, μ) is called a *duration*. As for probabilistic systems, we can introduce paths and adversaries for timed probabilistic systems, except transitions are now labelled by duration-distribution pairs and an adversary maps each finite path to a duration-distribution pair.

We restrict attention to *time-divergent adversaries*; a common restriction imposed in real-time systems so that unrealisable behaviour (i.e. corresponding to time not advancing beyond a bound) is disregarded during analysis. For any path

$$\omega = s_0 \xrightarrow{t_0,\mu_0} s_1 \xrightarrow{t_1,\mu_1} s_2 \xrightarrow{t_2,\mu_2} \cdots$$

of a timed probabilistic system, the duration up to the $(n{+}1)$th state of ω, denoted $\mathcal{D}_\omega(n{+}1)$, equals $\sum_{i=0}^{n} t_i$. We say that a path ω is *divergent* if for any $t \in \mathbb{R}_{\geqslant 0}$, there exists $j \in \mathbb{N}$ such that $\mathcal{D}_\omega(j){>}t$.

Definition 12 *An adversary* A *of a timed probabilistic system* TPS *is* divergent *if and only if for each state s of* TPS *the probability under* $Prob_s^A$ *of the divergent paths of* $Path_{ful}^A(s)$ *is* 1*. Let* Adv_{TPS} *be the set of divergent adversaries of* TPS*.*

Our notion of *probabilistic* divergence is less strict than the notion in which an adversary is divergent if and only if all of its paths are divergent, and therefore can avoid needless complications during the system construction process [16,26,3]. A restriction we impose on probabilistic timed systems is that of *non-zenoness*, which stipulates that there does not exist a state from which time cannot diverge, as we consider this situation to be a modelling error.

Definition 13 *A probabilistic timed system* TPS *is* non-zeno *if and only if there exists a divergent adversary of* TPS*.*

3 Probabilistic Timed Automata

In this section we review the definition of probabilistic timed automata [3], a modelling framework for real-time systems exhibiting both nondeterministic and stochastic behaviour. The formalism is derived by extending classical timed automata [6,7] with discrete probability distributions over edges. First, we introduce standard notation for clocks and zones of timed automata, and then we proceed to the definition of probabilistic timed automata. At the end of this section, we introduce PTCTL as a probabilistic timed temporal logic for the specification of properties of probabilistic timed automata.

3.1 Clocks and Zones

Let $\mathcal{X}$ be a finite set of variables called *clocks* which take values from the time domain $\mathbb{R}_{\geqslant 0}$ (non-negative reals). A function $v : \mathcal{X} \to \mathbb{R}_{\geqslant 0}$ is referred to as a *clock valuation*. The set of all clock valuations is denoted by $\mathbb{R}_{\geqslant 0}^{\mathcal{X}}$. For any $v \in \mathbb{R}_{\geqslant 0}^{\mathcal{X}}$ and $t \in \mathbb{R}_{\geqslant 0}$, we use $v{+}t$ to denote the clock valuation defined as $(v{+}t)(x) = v(x){+}t$ for all $x \in \mathcal{X}$. We use $v[X{:=}0]$ to denote the clock valuation obtained from v by resetting all of the clocks in $X \subseteq \mathcal{X}$ to 0, and leaving the values of all other clocks unchanged; formally, $v[X{:=}0](x) = 0$ if $x \in X$ and $v[X{:=}0](x) = v(x)$ otherwise.

The set of *zones* of $\mathcal{X}$, written $Zones(\mathcal{X})$, is defined inductively by the syntax:

$$\zeta ::= x \leqslant d \mid c \leqslant x \mid x{+}c \leqslant y{+}d \mid \neg\zeta \mid \zeta \vee \zeta$$

where $x, y \in \mathcal{X}$ and $c, d \in \mathbb{N}$. As usual, $\zeta_1 \wedge \zeta_2 = \neg(\neg\zeta_1 \vee \neg\zeta_2)$ and strict constraints can be written using negation, for example $x{>}2 = \neg(x{\leqslant}2)$.

The clock valuation v *satisfies* the zone ζ, written $v \triangleright \zeta$, 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. Intuitively, the semantics of a zone is the set of clock valuations (subset of $\mathbb{R}^{\mathcal{X}}_{\geqslant 0}$) which satisfy the zone. Note that more than one zone may represent the same set of clock valuations (for example, $(x{\leqslant}2) \wedge (y{\leqslant}1) \wedge (x{\leqslant}y{+}2)$ and $(x{\leqslant}2) \wedge (y{\leqslant}1) \wedge (x{\leqslant}y{+}3)$). We henceforth consider only canonical zones, which are zones for which the constraints are as 'tight' as possible. For any valid zone $\zeta \in Zones(\mathcal{X})$, there exists a $O(|\mathcal{X}|^3)$ algorithm to compute the (unique) canonical zone of ζ [27]. This enables us to use the above syntax for zones interchangeably with semantic, set-theoretic operations.

We require the following classical operations on zones [7,28]. For zones $\zeta, \zeta' \in Zones(\mathcal{X})$ and subset a of clocks $X \subseteq \mathcal{X}$, let:

$$\begin{aligned}
\swarrow_{\zeta'} \zeta &\overset{\text{def}}{=} \Big\{ v \mid \exists t{\geqslant}0.\, \Big(v{+}t \triangleright \zeta \;\wedge\; \forall t'{\leqslant}t.\, (v{+}t' \triangleright \zeta{\vee}\zeta') \Big) \Big\} \\
[X{:=}0]\zeta &\overset{\text{def}}{=} \{ v \mid v[X{:=}0] \triangleright \zeta \} \\
\zeta[X{:=}0] &\overset{\text{def}}{=} \{ v[X{:=}0] \mid v \triangleright \zeta \}\,.
\end{aligned}$$

The zone $\swarrow_{\zeta'} \zeta$ contains the clock valuations that can, by letting time pass, reach a clock valuation in ζ and remain in ζ' until ζ is reached. The zone $[X{:=}0]\zeta$ contains the clock valuations which result in a clock valuation in ζ when the clocks in X are reset to 0. The zone $\zeta[X{:=}0]$ contains the clock valuations which are obtained from clock valuations in ζ by resetting the clocks in X to 0.

3.2 Syntax and Semantics of Probabilistic Timed Automata

We now present the formal syntax of probabilistic timed automata.

Definition 14 *A* probabilistic timed automaton *is a tuple* $(L, \mathcal{X}, inv, prob, \mathcal{L})$ *where:*

- L *is a finite set of* locations;
- $\mathcal{X}$ *is a finite set of* clocks;
- $inv : L \to Zones(\mathcal{X})$ *is a function called the* invariant condition;
- $prob \subseteq L{\times}Zones(\mathcal{X}){\times}\mathsf{Dist}(2^{\mathcal{X}}{\times}L)$ *is a finite set called the* probabilistic edge relation;
- $\mathcal{L} : L \to 2^{AP}$ *is a labelling function assigning atomic propositions to locations.*

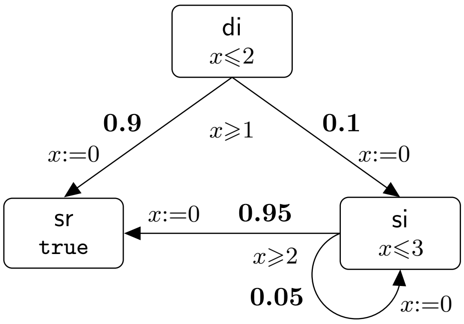

Fig. 1. A probabilistic timed automaton modelling a probabilistic protocol.

A *state* of a probabilistic timed automaton PTA is a pair $(l,v) \in L \times \mathbb{R}^{\mathcal{X}}_{\geqslant 0}$ such that $v \triangleright inv(l)$. Informally, the behaviour of a probabilistic timed automaton can be understood as follows. In any state (l,v), there is a nondeterministic choice of either (1) making a *discrete transition* or (2) letting *time pass*. In case (1), a discrete transition can be made according to any $(l,g,p) \in prob$ with source location l which is *enabled*; that is, zone g is satisfied by the current clock valuation v. Then the probability of moving to the location l' and resetting all of the clocks in the set X to 0 is given by $p(X,l')$. In case (2), the option of letting time pass is available only if the invariant condition $inv(l)$ is continuously satisfied while time elapses.

Definition 15 *An* edge *of* PTA *generated by* $(l,g,p) \in prob$ *is a tuple of the form* (l,g,p,X,l') *such that* $p(X,l') > 0$. *Let* $\mathsf{edges}(l,g,p)$ *be the set of edges generated by* (l,g,p), *and let* $\mathsf{edges} = \{\mathsf{edges}(l,g,p) \mid (l,g,p) \in prob\}$.

Example 16 *Consider the probabilistic timed automaton modelling a simple probabilistic communication protocol given in Figure 1. The nodes represent the locations, namely* di *(sender has data, receiver idle),* si *(sender sent data, receiver idle), and* sr *(sender sent data, receiver received). The automaton starts in location* di *in which data has been received by the sender. 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 *after 2 to 3 time units, the protocol will attempt to resend the data, which again can be lost, this time with probability 0.05.*

We now give the semantics of probabilistic timed automata defined in terms of timed probabilistic systems.

Definition 17 *Let* $\mathsf{PTA} = (L, \mathcal{X}, inv, prob, \mathcal{L})$ *be a probabilistic timed automaton. The* semantics *of* PTA *is defined as the timed probabilistic system* $\mathsf{TPS}_{\mathsf{PTA}} = (S, \mathit{TSteps}, \mathcal{L}')$ *where:*

- $S \subseteq L \times \mathbb{R}^{\mathcal{X}}_{\geqslant 0}$ *and* $(l,v) \in S$ *if and only if* $v \triangleright inv(l)$;
- $((l,v),t,\mu) \in \mathit{TSteps}$ *if and only if one of the following conditions holds:* ***time transitions:*** $t \geqslant 0$, $\mu = \mu_{(l,v+t)}$ *and* $v+t' \triangleright inv(l)$ *for all* $0 \leqslant t' \leqslant t$

discrete transitions: *$t{=}0$ and there exists $(l,g,p) \in prob$ such that $v \triangleright g$, $v[X{:=}0] \triangleright inv(l')$ for all $(X,l') \in \mathsf{support}(p)$, and for any $(l',v') \in S$:*

$$\mu(l',v') = \sum_{\substack{X \subseteq \mathcal{X}\ \&\\ v'=v[X:=0]}} p(X,l');$$

- *$\mathcal{L}'(l,v) = \mathcal{L}(l)$ for any $(l,v) \in S$.*

We say that PTA is non-zeno if and only if $\mathsf{TPS}_{\mathsf{PTA}}$ is non-zeno. When clear from the context, we omit the PTA subscript of $\mathsf{TPS}_{\mathsf{PTA}}$.

We say that a probabilistic timed automaton is *well-formed* if whenever a probabilistic edge is enabled it can be taken. Formally, a probabilistic timed automaton $\mathsf{PTA} = (L, \mathcal{X}, inv, prob, \mathcal{L})$ is said to be well-formed if:

$$\forall (l,g,p) \in prob.\ \forall v \in \mathbb{R}_{\geqslant 0}^{\mathcal{X}}.\ (v \triangleright g) \to \Big(\forall (X,l') \in \mathsf{support}(p).\ v[X{:=}0] \triangleright inv(l') \Big)\ .$$

A probabilistic timed automaton can be transformed into a well-formed probabilistic timed automaton by simply replacing the guard g in each probabilistic edge $(l,g,p) \in prob$ with

$$\Big(\wedge_{(X,l') \in \mathsf{support}(p)} [X{:=}0] inv(l') \Big) \wedge g\ .$$

Since this transformation has no effect on the semantics of the automaton, for the remainder of the paper we assume all probabilistic timed automata we consider are well-formed.

3.3 Probabilistic Timed Computation Tree Logic

We now describe Probabilistic Timed Computation Tree Logic (PTCTL) which can be used to specify properties of probabilistic timed automata. This logic is a combination of two extensions of the temporal logic CTL [29], the timed logic TCTL [12,7] and the probabilistic logic PCTL [11,17]. The logic TCTL employs a set of *formula clocks*, $\mathcal{Z}$, disjoint from the clocks $\mathcal{X}$ of the probabilistic timed automaton under study. Formula clocks are assigned values by *formula clock valuations* $\mathcal{E} \in \mathbb{R}_{\geqslant 0}^{\mathcal{Z}}$. The logic TCTL can express timing constraints and includes the *reset quantifier* $z.\phi$, used to reset the formula clock z so that the formula ϕ is evaluated from a state at which $z{=}0$. PTCTL is obtained by enhancing TCTL with the probabilistic quantifier $\mathcal{P}_{\sim\lambda}[\cdot]$ from PCTL and removing the path quantifiers $\exists$ and $\forall$.

Definition 18 *The syntax of* PTCTL *is defined as follows:*

$$\phi ::= a \,\Big|\, \zeta \,\Big|\, \neg\phi \,\Big|\, \phi \vee \phi \,\Big|\, z.\phi \,\Big|\, \mathcal{P}_{\sim\lambda}[\phi\, \mathcal{U}\, \phi] \,\Big|\, \mathcal{P}_{\sim\lambda}[\phi\, \mathcal{V}\, \phi]$$

where $a \in AP$*,* $\zeta \in Zones(\mathcal{X} \cup \mathcal{Z})$*,* $z \in \mathcal{Z}$*,* $\sim \in \{\leqslant, <, >, \geqslant\}$ *and* $\lambda \in [0, 1]$.

We use the abbreviations $\Diamond\phi$ and $\Box\phi$ for $\texttt{true}\,\mathcal{U}\,\phi$ and $\texttt{false}\,\mathcal{V}\,\phi$ respectively.

In PTCTL we can express properties such as:

- 'with probability strictly greater than 0.99, the system delivers packet 1 within 5 time units and does not try to send packet 2 in the meantime', which is represented by $z.\mathcal{P}_{>0.99}[\mathit{packet2unsent}\,\mathcal{U}\,(\mathit{packet1delivered}\wedge(z<5))]$;
- 'with probability at least 0.95, the system clock x does not exceed 3 before 8 time units elapse', which is represented as $z.\mathcal{P}_{\geqslant 0.95}[(x\leqslant 3)\,\mathcal{U}\,(z=8)]$;
- 'the system remains up after the first 60 time units have elapsed with probability greater than 0.99', represented as $z.\mathcal{P}_{\geqslant 0.99}[\Box\,(\mathit{system\_up} \vee (z\leqslant 60))]$.

Next, we define the semantics of PTCTL. We write $v, \mathcal{E}$ to denote the composite clock valuation in $\mathbb{R}_{\geqslant 0}^{(\mathcal{X}\cup\mathcal{Z})}$ obtained from $v \in \mathbb{R}_{\geqslant 0}^{\mathcal{X}}$ and $\mathcal{E} \in \mathbb{R}_{\geqslant 0}^{\mathcal{Z}}$. Given a state and formula clock valuation pair $(l, v), \mathcal{E}$, zone ζ and duration t, by abuse of notation we let $(l, v), \mathcal{E} \triangleright \zeta$ denote $v, \mathcal{E} \triangleright \zeta$, and $(l, v)+t$ denote $(l, v+t)$.

Definition 19 *Let* $\mathsf{TPS} = (S, \mathit{TSteps}, \mathcal{L}')$ *be the timed probabilistic system associated with the probabilistic timed automaton* PTA*. For any state* $s \in S$*, formula clock valuation* $\mathcal{E} \in \mathbb{R}_{\geqslant 0}^{\mathcal{Z}}$ *and* PTCTL *formula* θ*, we say that* $s, \mathcal{E}$ *satisfies* θ*, written* $s, \mathcal{E} \models \theta$*, where the relation* $\models$ *is defined inductively as follows:*

$$\begin{aligned}
s, \mathcal{E} \models a &\;\Leftrightarrow\; a \in \mathcal{L}'(s)\\
s, \mathcal{E} \models \zeta &\;\Leftrightarrow\; s, \mathcal{E} \triangleright \zeta\\
s, \mathcal{E} \models \phi \vee \psi &\;\Leftrightarrow\; s, \mathcal{E} \models \phi \;\text{ or }\; s, \mathcal{E} \models \psi\\
s, \mathcal{E} \models \neg\phi &\;\Leftrightarrow\; s, \mathcal{E} \not\models \phi\\
s, \mathcal{E} \models z.\phi &\;\Leftrightarrow\; s, \mathcal{E}[z{:=}0] \models \phi\\
s, \mathcal{E} \models \mathcal{P}_{\sim\lambda}[\varphi] &\;\Leftrightarrow\; p^{A}_{s,\mathcal{E}}(\varphi) \sim \lambda \;\text{ for all }\; A \in \mathit{Adv}_{\mathsf{TPS}}
\end{aligned}$$

where $p^{A}_{s,\mathcal{E}}(\varphi) = \mathit{Prob}^{A}_{s}\{\omega \in \mathit{Path}^{A}_{\mathit{ful}}(s) \,|\, \omega, \mathcal{E} \models \varphi\}$ *and for any* $\omega \in \mathit{Path}_{\mathit{ful}}(s)$*:*

- $\omega, \mathcal{E} \models \phi\,\mathcal{U}\,\psi$ *if and only if there exists* $i \in \mathbb{N}$ *and* $t \leqslant \mathcal{D}_{\omega}(i{+}1){-}\mathcal{D}_{\omega}(i)$ *such that*

  – $\omega(i){+}t, \mathcal{E}{+}\mathcal{D}_{\omega}(i){+}t \models \psi$
  – $\forall t'{<}t.\, (\omega(i){+}t', \mathcal{E}{+}\mathcal{D}_{\omega}(i){+}t' \models \phi{\vee}\psi)$
  – $\forall j{<}i.\, \forall t'{\leqslant}\mathcal{D}_{\omega}(j{+}1){-}\mathcal{D}_{\omega}(j).\, (\omega(j){+}t', \mathcal{E}{+}\mathcal{D}_{\omega}(j){+}t' \models \phi{\vee}\psi)$

- $\omega, \mathcal{E} \models \phi \, \mathcal{V} \, \psi$ *if and only if for all* $i \in \mathbb{N}$ *and* $t \leqslant \mathcal{D}_\omega(i{+}1){-}\mathcal{D}_\omega(i)$*, if*

  – $\forall t'{<}t. \, (\omega(i){+}t', \mathcal{E}{+}\mathcal{D}_\omega(i){+}t' \not\models \phi{\wedge}\psi)$
  – $\forall j{<}i. \, \forall t'{\leqslant}\mathcal{D}_\omega(j{+}1){-}\mathcal{D}_\omega(j). \, (\omega(j){+}t', \mathcal{E} + \mathcal{D}_\omega(j){+}t' \not\models \phi{\wedge}\psi)$

  then $\omega(i){+}t, \mathcal{E}{+}\mathcal{D}_\omega(i){+}t \models \psi$.

For any PTCTL formula ϕ we denote by $Sat(\phi)$ the set of state and formula clock valuation pairs which satisfy ϕ, that is: $Sat(\phi) = \{s, \mathcal{E} \in S{\times}\mathbb{R}_{\geqslant 0}^{\mathcal{Z}} \mid s, \mathcal{E} \models \phi\}$.

We now present a number of definitions and lemmas concerning the satisfaction of PTCTL formulae that we will require in the remainder of the paper.

Lemma 20 *Let* PTA *be a probabilistic timed automaton,* $\mathsf{TPS}{=}(S, \mathit{TSteps}, \mathcal{L}')$ *be the corresponding timed probabilistic system and* ϕ *and* ψ *be* PTCTL *formulae. For any state and formula clock valuation pair* $\omega, \mathcal{E} \in \mathit{Path}_{\mathit{ful}} {\times} \mathbb{R}_{\geqslant 0}^{\mathcal{Z}}$*:*

$$\omega, \mathcal{E} \models \phi \, \mathcal{U} \, \psi \quad \Leftrightarrow \quad \omega, \mathcal{E} \not\models \neg\phi \, \mathcal{V} \, \neg\psi \, .$$

Proof. The proof follows from the semantics of PTCTL (Definition 19). □

Proposition 21 *Let* $\mathsf{PTA} = (L, \mathcal{X}, \mathit{inv}, \mathit{prob}, \mathcal{L})$ *be a probabilistic timed automaton,* $\mathsf{TPS}{=}(S, \mathit{TSteps}, \mathcal{L}')$ *be the corresponding timed probabilistic system and* θ *be a* PTCTL *formula. There is a finite probabilistic system (the region graph)* R *such that for any* PTCTL *formulae* ϕ *and* ψ*, adversary* $A \in \mathit{Adv}_{\mathsf{TPS}}$ *and state-formula clock valuation pair* $s, \mathcal{E} \in S \times 2^{\mathcal{Z}}$*, there exists* PCTL *formulae* Φ *and* Ψ*, adversary* B *of* R *and state* r *of* R *such that:*

- *if* $\mathcal{P}_{\sim\lambda}[\phi \, \mathcal{U} \, \psi]$ *is a subformula of* θ*, then* $p_s^A(\phi \, \mathcal{U} \, \psi) = p_r^B(\Phi \, \mathcal{U} \, \Psi)$*;*
- *if* $\mathcal{P}_{\sim\lambda}[\phi \, \mathcal{V} \, \psi]$ *is a subformula of* θ*, then* $p_s^A(\phi \, \mathcal{V} \, \psi) = p_r^B(\Phi \, \mathcal{V} \, \Psi)$*.*

Proof. The proof follows from the region graph construction [6] applied to probabilistic timed automata [3]. In particular, the state r of R corresponds to the unique region to which $s, \mathcal{E}$ belongs. □

Lemma 22 *Let* $\mathsf{PTA} = (L, \mathcal{X}, \mathit{inv}, \mathit{prob}, \mathcal{L})$ *be a probabilistic timed automaton and* $\mathsf{TPS}{=}(S, \mathit{TSteps}, \mathcal{L}')$ *be the corresponding timed probabilistic system. For any state-formula clock valuation pair* $s, \mathcal{E} \in S \times 2^{\mathcal{Z}}$ *and* PTCTL *path formula* φ*, there exists adversaries* A_1 *and* A_2 *such that:*

$$p_{s,\mathcal{E}}^{A_1}(\varphi) = \inf_{A \in \mathit{Adv}_{\mathsf{TPS}}} p_{s,\mathcal{E}}^{A}(\varphi) \;\; \text{and} \;\; p_{s,\mathcal{E}}^{A_2}(\varphi) = \sup_{A \in \mathit{Adv}_{\mathsf{TPS}}} p_{s,\mathcal{E}}^{A}(\varphi) \, .$$

```
**algorithm** $\mathsf{PTCTLModelCheck}(\mathsf{PTA}, \theta)$

**output:** set of symbolic states $\llbracket\theta\rrbracket$ **such that**

    $\llbracket a\rrbracket := \{(l, \mathit{inv}(l)) \,|\, l \in L \text{ and } l \in \mathcal{L}(a)\}$
    $\llbracket\zeta\rrbracket := \{(l, \mathit{inv}(l) \wedge \zeta) \,|\, l \in L\}$
    $\llbracket\neg\phi\rrbracket := \{(l, \mathit{inv}(l) \wedge \neg \bigvee_{(l,\zeta)\in\llbracket\phi\rrbracket} \zeta) \,|\, l \in L\}$
    $\llbracket\phi \vee \psi\rrbracket := \llbracket\phi\rrbracket \vee \llbracket\psi\rrbracket$
    $\llbracket z.\phi\rrbracket := \{(l, [\{z\}{:=}0]\zeta) \,|\, (l,\zeta) \in \llbracket\phi\rrbracket\}$
    $\llbracket\mathcal{P}_{\sim\lambda}[\phi\,\mathcal{U}\,\psi]\rrbracket := \mathsf{Until}(\llbracket\phi\rrbracket, \llbracket\psi\rrbracket, \sim\lambda)$
    $\llbracket\mathcal{P}_{\sim\lambda}[\phi\,\mathcal{V}\,\psi]\rrbracket := \mathsf{Release}(\llbracket\phi\rrbracket, \llbracket\psi\rrbracket, \sim\lambda)$
```

Fig. 2. Symbolic PTCTL model checking algorithm

Proof. Employing Proposition 21 we can reduce the problem to finite state probabilistic systems and since we have used a probabilistic version of divergence the result is a simple adaptation of the approach used for probabilistic systems under probabilistic notions of fairness, see for example in the case of supremum [30, Lemma 9.5.15 (page 243)]. □

4 Symbolic PTCTL Model Checking

In this section, we present a method for model checking a probabilistic timed automaton against PTCTL formulae. Our algorithm relies on an implicit, symbolic representation of the clock-valuation space (and also avoids explicit construction of the probabilistic timed automaton's region graph, as utilised in [3]). In order to represent symbolically the state sets computed during the model checking process, we use the concept of *symbolic state*: a symbolic state is a pair (l, ζ) comprising a location and a zone over $\mathcal{X} \cup \mathcal{Z}$. The set of state and formula clock valuation pairs corresponding to a symbolic state (l, ζ) is $\{(l, v), \mathcal{E} \,|\, v, \mathcal{E} \triangleright \zeta\}$, while the state set corresponding to a set of symbolic states is the union of those corresponding to each individual symbolic state. In the manner standard for model checking, we progress up the parse tree of a PTCTL formula, from the leaves to the root, recursively calling the algorithm $\mathsf{PTCTLModelCheck}$, shown in Figure 2, to compute the set of symbolic states which satisfy each subformula. Handling observables and Boolean operations is classical, and we therefore reduce our problem to computing $\mathsf{Until}(\llbracket\phi_1\rrbracket, \llbracket\phi_2\rrbracket, {\sim}\lambda)$ and $\mathsf{Release}(\llbracket\phi_1\rrbracket, \llbracket\phi_2\rrbracket, {\sim}\lambda)$, which arises when we check a probabilistically quantified formula.

As in the cases for (non-probabilistic) timed automata and (finite-state) probabilistic systems with fairness constraints, when considering properties which have universal quantification over paths or require the computation of minimum probabilities, the standard algorithm can no longer be applied. For example, for any formula clock $z \in \mathcal{Z}$, under divergent adversaries the minimum probability of reaching $z{>}1$ is 1; however, if we remove the restriction to time-divergent adversaries this minimum probability becomes 0.

The techniques we introduce here are based on those for non-probabilistic timed automata [7], which we now recall. For the discussion below, to simplify presentation, we will use a TCTL formula to represent its corresponding satisfaction set, i.e. use ϕ to denote $\mathit{Sat}(\phi)$, and dually allow a set of state and formula clock valuation pairs to represent a TCTL formula, i.e. use Y to denote a formula where $s, \mathcal{E} \models Y$ if and only if $s, \mathcal{E} \in Y$.

In [7], it is shown that verifying $\phi \ \forall\mathcal{U} \ \psi$ ('all divergent paths satisfy $\phi \, \mathcal{U} \, \psi$') reduces to computing the fixpoint:

$$\mathit{lfp}\, Y. \Big(\psi \vee \neg z. \Big(\neg Y \ \exists\mathcal{U} \ (\neg(\phi \vee Y) \vee (z{>}c)) \Big) \Big) \tag{3}$$

for any $c \in \mathbb{N}$ greater than 0. The important point is that the universal quantification over paths has been replaced by an existential quantification, combined with a constraint enforcing that more than c time units must elapse repeatedly.

For the analysis of probabilistic timed automata it is convenient to consider, instead of until, the dual, release formula $\phi \ \exists\mathcal{V} \ \psi$ ('there exists a divergent path satisfying $\phi \, \mathcal{V} \, \psi$'). Using (3) and the duality between $\mathcal{U}$ and $\mathcal{V}$, for any $c \in \mathbb{N}$ greater than 0, $\phi \ \exists\mathcal{V} \ \psi$ reduces to computing:

$$\begin{aligned}
& \neg \mathit{lfp}\, Y. \Big(\neg\psi \vee \neg z. \Big((\neg Y) \ \exists\mathcal{U} \ (\neg(\neg\phi \vee Y) \vee (z{>}c)) \Big) \Big) \\
&= \neg \mathit{lfp}\, Y. \Big(\neg(\psi \wedge \neg\neg z. \Big((\neg Y) \ \exists\mathcal{U} \ (\neg(\neg\phi \vee Y) \vee (z{>}c)) \Big) \Big) \\
&= \neg \mathit{lfp}\, Y. \Big(\neg(\psi \wedge z. \Big((\neg Y) \ \exists\mathcal{U} \ (\neg(\neg\phi \vee Y) \vee (z{>}c)) \Big) \Big) \\
&= \neg \mathit{lfp}\, Y. \Big(\neg(\psi \wedge z. \Big((\neg Y) \ \exists\mathcal{U} \ ((\neg\neg\phi \wedge \neg Y) \vee (z{>}c)) \Big) \Big) \\
&= \neg \mathit{lfp}\, Y. \Big(\neg(\psi \wedge z. \Big((\neg Y) \ \exists\mathcal{U} \ ((\phi \wedge \neg Y) \vee (z{>}c)) \Big) \Big) \\
&= \mathit{gfp}\, Y. \Big(\psi \wedge z. \Big(Y \ \exists\mathcal{U} \ ((\phi \wedge Y) \vee (z{>}c)) \Big) \Big) \, .
\end{aligned}$$

The validity of the above reduction steps correspond to standard logical equivalences (either $\neg\neg\theta \equiv \theta$, $\theta \vee \theta' \equiv \neg(\neg\theta \wedge \neg\theta')$ or $\neg(\theta \vee \theta') \equiv \neg\theta \wedge \neg\theta'$) except the final reduction which follows from the duality between the least and greatest fixpoint ($\neg \mathit{lfp}\, X. (\neg\theta\{X{:=}\neg X\}) \equiv \mathit{gfp}\, X. \theta$). Therefore, verifying the formula $\phi \ \exists\mathcal{V} \ \psi$ can be performed by computing the fixpoint:

$$\mathit{gfp}\, Y. \Big(\psi \wedge z. \Big(Y \ \exists\mathcal{U} \ ((\phi \wedge Y) \vee (z{>}c)) \Big) \Big) \, . \tag{4}$$

Now, letting[1]

$$p^{\max}_{s,\mathcal{E}}(\varphi) \stackrel{\text{def}}{=} \sup_{A \in Adv_{\mathsf{TPS}}} p^{A}_{s,\mathcal{E}}(\varphi) \quad \text{and} \quad p^{\min}_{s,\mathcal{E}}(\varphi) \stackrel{\text{def}}{=} \inf_{A \in Adv_{\mathsf{TPS}}} p^{A}_{s,\mathcal{E}}(\varphi)\,,$$

we have, for any state and formula clock valuation pair $s, \mathcal{E}$:

$$\begin{aligned} p^{\min}_{s,\mathcal{E}}(\phi\,\mathcal{U}\,\psi) &= \inf_{A \in Adv_{\mathsf{TPS}}} p^{A}_{s,\mathcal{E}}(\phi\,\mathcal{U}\,\psi) \\ &= \inf_{A \in Adv_{\mathsf{TPS}}} \Big(1 - p^{A}_{s,\mathcal{E}}(\neg\phi\,\mathcal{V}\,\neg\psi)\Big) && \text{by Lemma 20} \\ &= 1 - \sup_{A \in Adv_{\mathsf{TPS}}} p^{A}_{s,\mathcal{E}}(\neg\phi\,\mathcal{V}\,\neg\psi) && \text{rearranging} \\ &= 1 - p^{\max}_{s,\mathcal{E}}(\neg\phi\,\mathcal{V}\,\neg\psi)\,. \end{aligned}$$

Substituting this equality into the semantics of PTCTL (Definition 19) we have:

$$\{s,\mathcal{E} \mid s,\mathcal{E} \models \mathcal{P}_{\lesssim\lambda}[\phi\,\mathcal{U}\,\psi]\} = \{s,\mathcal{E} \mid p^{\max}_{s,\mathcal{E}}(\phi\,\mathcal{U}\,\psi) \lesssim \lambda\} \tag{5}$$

$$\{s,\mathcal{E} \mid s,\mathcal{E} \models \mathcal{P}_{\lesssim\lambda}[\phi\,\mathcal{V}\,\psi]\} = \{s,\mathcal{E} \mid p^{\max}_{s,\mathcal{E}}(\phi\,\mathcal{V}\,\psi) \lesssim \lambda\} \tag{6}$$

$$\{s,\mathcal{E} \mid s,\mathcal{E} \models \mathcal{P}_{\gtrsim\lambda}[\phi\,\mathcal{U}\,\psi]\} = \{s,\mathcal{E} \mid 1{-}\lambda \gtrsim p^{\max}_{s,\mathcal{E}}(\neg\phi\,\mathcal{V}\,\neg\psi)\} \tag{7}$$

$$\{s,\mathcal{E} \mid s,\mathcal{E} \models \mathcal{P}_{\gtrsim\lambda}[\phi\,\mathcal{V}\,\psi]\} = \{s,\mathcal{E} \mid 1{-}\lambda \gtrsim p^{\max}_{s,\mathcal{E}}(\neg\phi\,\mathcal{U}\,\neg\psi)\} \tag{8}$$

that is we have reduced the model checking problem to the computation of maximum probabilities for until and release formulae.

We begin in Section 4.1 by introducing operations on symbolic states. In Section 4.2, we introduce algorithms for calculating the maximum until probabilities, while in Section 4.3 we present algorithms for calculating the maximum release probabilities. In each case we include specialised algorithms for qualitative formulae ($\lambda \in \{0,1\}$), as, for such formulae, verification can be performed through non-numerical analysis [31,32]. Then in Section 4.4 we show how to ensure that a probabilistic timed automaton is non-zeno, Section 4.5 discusses the termination of the algorithms introduced.

Note that the cases $\mathcal{P}_{\geqslant 0}[\cdot]$ and $\mathcal{P}_{\leqslant 1}[\cdot]$ are trivially satisfied, while the cases $\mathcal{P}_{<0}[\cdot]$ and $\mathcal{P}_{>1}[\cdot]$ are trivially not satisfied, and therefore we omit these cases in our analysis.

4.1 Operations on Symbolic States

In this section we extend the *time predecessor* and *discrete predecessor* functions tpre and dpre of [7,28] to probabilistic timed automata. First, for any set

[1] Note that, Lemma 22 implies that 'minimum' and 'maximum' can be used instead of 'infimum' and 'supremum'.

of symbolic states $\mathtt{U}$, let $\zeta^l_{\mathtt{U}} = \bigvee\{\zeta \mid (l,\zeta) \in \mathtt{U}\}$; that is, $\zeta^l_{\mathtt{U}}$ is the zone such that $v, \mathcal{E} \triangleright \zeta^l_{\mathtt{U}}$ if and only if $(l,v), \mathcal{E} \in \mathtt{u}$ for some $\mathtt{u} \in \mathtt{U}$. For any sets of symbolic states $\mathtt{U}, \mathtt{V} \subseteq L \times \mathit{Zones}(\mathcal{X} \cup \mathcal{Z})$, clock $z \in \mathcal{Z}$ and edge (l,g,p,X,l'):

$$\begin{aligned} z.\mathtt{U} &\overset{\text{def}}{=} \Big\{ (l, [\{z\}{:=}0]\zeta^l_{\mathtt{U}}) \,\Big|\, l \in L \Big\} \\ \mathsf{tpre}_{\mathtt{U}}(\mathtt{V}) &\overset{\text{def}}{=} \Big\{ (l, \swarrow_{\zeta^l_{\mathtt{U}} \wedge inv(l)} (\zeta^l_{\mathtt{V}} \wedge inv(l))) \,\Big|\, l \in L \Big\} \\ \mathsf{dpre}((l,g,p,X,l'),\mathtt{U}) &\overset{\text{def}}{=} \Big\{ (l, g \wedge inv(l) \wedge ([X{:=}0]\zeta^{l'}_{\mathtt{U}})) \Big\}. \end{aligned}$$

Furthermore, we define the conjunction and disjunction of sets of symbolic states as follows:

$$\mathtt{U} \wedge \mathtt{V} \overset{\text{def}}{=} \Big\{ (l, \zeta^l_{\mathtt{U}} \wedge \zeta^l_{\mathtt{V}}) \,\Big|\, l \in L \Big\} \quad \text{and} \quad \mathtt{U} \vee \mathtt{V} \overset{\text{def}}{=} \Big\{ (l, \zeta^l_{\mathtt{U}} \vee \zeta^l_{\mathtt{V}}) \,\Big|\, l \in L \Big\}.$$

Finally, let $[\![\mathtt{false}]\!] \overset{\text{def}}{=} \emptyset$ and $[\![\mathtt{true}]\!] \overset{\text{def}}{=} \{(l, inv(l)) \mid l \in L\}$, the sets of symbolic states representing the empty and full state sets respectively.

Informally $z.\mathtt{U}$ denotes the set of symbolic states describing those state and formula clock valuation pairs which, when clock z is reset to 0, belong to the set of states and formula clock valuation pairs encoded by $\mathtt{U}$. We denote by $\mathsf{tpre}_{\mathtt{U}}(\mathtt{V})$ the set of symbolic states describing those state and formula clock valuation pairs which belong to the set encoded by $\mathtt{V}$ by letting time elapse, remaining at all intermediate times in the set encoded by $\mathtt{U}$. Finally, we denote by $\mathsf{dpre}((l,g,p,X,l'),\mathtt{U})$ the set of symbolic states describing those state and formula clock valuation pairs which, when the edge (l,g,p,X,l') is traversed, belong to the set encoded by $\mathtt{U}$.

4.2 Computing Maximum Until Probabilities

In this section we present methods for calculating the set of states satisfying a formula of the form $\mathcal{P}_{\precsim\lambda}[\phi\,\mathcal{U}\,\psi]$ and $\mathcal{P}_{\succsim\lambda}[\phi\,\mathcal{V}\,\psi]$ which, from (5) and (8), reduce to the computation of $p^{\max}_{s,\mathcal{E}}(\phi\,\mathcal{U}\,\psi)$ or $p^{\max}_{s,\mathcal{E}}(\neg\phi\,\mathcal{U}\,\neg\psi)$ for all state and formula clock valuation pairs $s, \mathcal{E}$. Note that, since we consider only non-zeno automata, when calculating these sets we can ignore the restriction to divergent adversaries; intuitively, letting time converge cannot make the event of reaching a ψ-satisfying state more probable. This is analogous to the fact that verifying the same type of properties against (finite-state) probabilistic systems does not need to take *fairness* constraints into account [24], and that verifying (non-probabilistic) non-zeno timed automata against formulae of the form $\phi\,\exists\mathcal{U}\,\psi$ ('there exists a divergent path which satisfies $\phi\,\mathcal{U}\,\psi$') does not need to take divergence of paths into account [7].

4.2.1 The Qualitative Case

We first concentrate on the *qualitative* case, that is compute the set of states satisfying $\phi\,\mathcal{U}\,\psi$ with maximum probability equal to 1, or maximum probability strictly greater than 0, respectively. Our approach is inspired by the methods for computing the associated properties on *finite-state* probabilistic systems [16], which we now recall.

Theorem 23 ([16,33]) *Let* $\mathsf{PS} = (S, \mathit{Steps}, \mathcal{L})$ *be a finite-state probabilistic system and* $\Phi\,\mathcal{U}\,\Psi$ *a* PCTL *path formula.*

- *The set* $\{s \in S \,|\, p_s^{\max}(\Phi\,\mathcal{U}\,\Psi){>}0\}$ *is given by the fixpoint*

$$\mathit{lfp}\,Y.\,\Big(\,\mathit{Sat}(\Psi) \cup \Big(\,\mathit{Sat}(\Phi) \cap \mathit{pre}_0^{\mathsf{PS}}(Y)\,\Big)\,\Big)$$

where $\mathit{pre}_0^{\mathsf{PS}}(Y) = \{s \,|\, \exists(s,\mu){\in}\mathit{Steps}.\,\mu(Y){>}0\}$ *for* $Y \subseteq S$.

- *The set* $\{s \in S \,|\, p_s^{\max}(\Phi\,\mathcal{U}\,\Psi){\geqslant}1\}$ *is given by the fixpoint*

$$\mathit{gfp}\,Y.\,\mathit{lfp}\,Y'.\,\Big(\,\mathit{Sat}(\Psi) \cup \Big(\,\mathit{Sat}(\Phi) \cap \mathit{pre}_1^{\mathsf{PS}}(Y,Y')\,\Big)\,\Big)$$

where $\mathit{pre}_1^{\mathsf{PS}}(Y,Y') = \{s \,|\, \exists(s,\mu){\in}\mathit{Steps}.\,(\mu(Y){=}1{\wedge}\mu(Y'){>}0)\}$ *for* $Y, Y' \subseteq S$.

Intuitively, $s \in \mathit{pre}_0^{\mathsf{PS}}(Y)$ if one can go from s to a state in Y with positive probability, and $s \in \mathit{pre}_1^{\mathsf{PS}}(Y,Y')$ if one can go from s to a state in Y' with positive probability and with probability 1 reach a state in Y.

In contrast to verifying a PCTL until formula against probabilistic systems, when checking the satisfaction of a PTCTL until formula $\phi\,\mathcal{U}\,\psi$ against a timed probabilistic system, one must check that, as time passes, the system remains in the set of states satisfying $\phi{\vee}\psi$. Therefore, in our context, the functions *pre0* and *pre1* are parameterised by a state set Y and require the continuous evolution through Y during a time-passage transition.

We now introduce the functions *pre0* and *pre1*, which operate on states of a probabilistic timed automaton, and are analogous to $\mathit{pre}_0^{\mathsf{PS}}$ and $\mathit{pre}_1^{\mathsf{PS}}$, respectively.

Definition 24 *Let* PTA *be a probabilistic timed automaton with corresponding timed probabilistic system* $\mathsf{TPS} = (S, \mathit{TSteps}, \mathcal{L}')$, *and* $Y, Y', Y_0, Y_1 \subseteq S$ *be sets of states of* TPS. *Then:*

$$\begin{aligned}\mathit{pre0}_Y(Y') = \Big\{ & s,\mathcal{E} \in S{\times}\mathbb{R}_{\geqslant 0}^{\mathcal{Z}} \,|\, \exists(s,t,\mu) \in \mathit{TSteps}.\\ & \exists s' \in S.\,\Big(s',\mathcal{E}{+}t \in Y' \wedge \mu(s'){>}0\Big) \wedge \forall t'{\leqslant}t.\,\Big(s{+}t',\mathcal{E}{+}t' \in Y{\cup}Y'\Big)\Big\}\end{aligned}$$

```
algorithm dpre0(V)

$\mathtt{Y} := [\![\mathtt{false}]\!]$
for $e \in$ edges
    $\mathtt{Y} := \mathtt{Y} \vee \mathsf{dpre}(e, \mathtt{V})$
end
return Y
```

```
algorithm dpre1(U, V)

$\mathtt{Y} := [\![\mathtt{false}]\!]$
for $(l, g, p) \in \mathit{prob}$
    $\mathtt{Y}_0 := [\![\mathtt{true}]\!]$
    $\mathtt{Y}_1 := [\![\mathtt{false}]\!]$
    for $e \in \mathsf{edges}(l, g, p)$
        $\mathtt{Y}_0 := \mathsf{dpre}(e, \mathtt{U}) \wedge \mathtt{Y}_0$
        $\mathtt{Y}_1 := \mathsf{dpre}(e, \mathtt{V}) \vee \mathtt{Y}_1$
    end
    $\mathtt{Y} := (\mathtt{Y}_0 \wedge \mathtt{Y}_1) \vee \mathtt{Y}$
end
return Y
```

Fig. 3. The functions dpre0 and dpre1

and $\mathit{pre1}_Y(Y_0, Y_1)$ *equals*

$$\Big\{ s, \mathcal{E} \in S \times \mathbb{R}^{\mathcal{Z}}_{\geqslant 0} \,|\, \exists (s, t, \mu) \in \mathit{TSteps}. \forall s' \in S. \Big(\mu(s') > 0 \rightarrow s', \mathcal{E} + t \in Y_0 \Big)$$
$$\wedge \exists s' \in S. \Big(s', \mathcal{E} + t \in Y_1 \wedge \mu(s') > 0 \Big) \wedge \forall t' \leqslant t. \Big(s + t', \mathcal{E} + t' \in Y \cup (Y_0 \cap Y_1) \Big) \Big\} .$$

Similarly to the finite-state case (Theorem 23), these functions can be embedded in fixpoint expressions which correspond to the complements of the state sets satisfying $\mathcal{P}_{\leqslant 0}[\phi\, \mathcal{U}\, \psi]$ or $\mathcal{P}_{<1}[\phi\, \mathcal{U}\, \psi]$. Note that the fixpoint expression given in Proposition 25 corresponds to finding those states and formula clock valuation pairs from which there exists a path satisfying $\phi\, \mathcal{U}\, \psi$, and therefore has the same structure as that used in [7] for verifying timed automata against the formula $\phi\, \exists \mathcal{U}\, \psi$.

Proposition 25 *Let* PTA *be a probabilistic timed automaton with corresponding timed probabilistic system* TPS $= (S, \mathit{TSteps}, \mathcal{L}')$*, and* ϕ, ψ *be* PTCTL *formulae. The set* $\{s, \mathcal{E} \in S \times \mathbb{R}^{\mathcal{Z}}_{\geqslant 0} \,|\, p^{\max}_{s,\mathcal{E}}(\phi\, \mathcal{U}\, \psi) > 0\}$ *is given by the fixpoint*

$$\mathit{lfp}\, Y. \Big(\mathit{Sat}(\psi) \cup \mathit{pre0}_{\mathit{Sat}(\phi \vee \psi)}(Y) \Big) .$$

Proof. To ease notation, we use $p^{\max}_{>0}(\phi\, \mathcal{U}\, \psi)$ to denote the set of state and formula clock valuation pairs $\{s, \mathcal{E} \in S \times \mathbb{R}^{\mathcal{Z}}_{\geqslant 0} \,|\, p^{\max}_{s,\mathcal{E}}(\phi\, \mathcal{U}\, \psi) > 0\}$. Our aim is to show that:

$$p^{\max}_{>0}(\phi\, \mathcal{U}\, \psi) = \mathit{lfp}\, Y. \Big(\mathit{Sat}(\psi) \cup \mathit{pre0}_{\mathit{Sat}(\phi \vee \psi)}(Y) \Big) .$$

We split the proof into two parts: first we show that $p^{\max}_{>0}(\phi\, \mathcal{U}\, \psi)$ is a fixpoint and second we show that it is the least fixpoint.

- To establish that $p^{\max}_{>0}(\phi\,\mathcal{U}\,\psi)$ is a fixpoint, that is $p^{\max}_{>0}(\phi\,\mathcal{U}\,\psi)$ equals $\mathit{Sat}(\psi)\cup \mathit{pre0}_{\mathit{Sat}(\phi\vee\psi)}(p^{\max}_{>0}(\phi\,\mathcal{U}\,\psi))$, we show that:

$$p^{\max}_{>0}(\phi\,\mathcal{U}\,\psi) \subseteq \mathit{Sat}(\psi) \cup \mathit{pre0}_{\mathit{Sat}(\phi\vee\psi)}(p^{\max}_{>0}(\phi\,\mathcal{U}\,\psi)) \qquad (9)$$

$$p^{\max}_{>0}(\phi\,\mathcal{U}\,\psi) \supseteq \mathit{Sat}(\psi) \cup \mathit{pre0}_{\mathit{Sat}(\phi\vee\psi)}(p^{\max}_{>0}(\phi\,\mathcal{U}\,\psi))\ . \qquad (10)$$

In the case of (9), for any $s,\mathcal{E} \in p^{\max}_{>0}(\phi\,\mathcal{U}\,\psi)$, by Definition 17, the zero-duration time transition $(s,0,\mu_s)$ is an element of TSteps, and from Definition 19 we have that $s,\mathcal{E} \models \phi\vee\psi$. Combining these two facts with Definition 24 it follows that $s,\mathcal{E} \in \mathit{pre0}_{\mathit{Sat}(\phi\vee\psi)}(p^{\max}_{>0}(\phi\,\mathcal{U}\,\psi))$. Hence, since $s,\mathcal{E} \in p^{\max}_{>0}(\phi\,\mathcal{U}\,\psi)$ was arbitrary, $p^{\max}_{>0}(\phi\,\mathcal{U}\,\psi) \subseteq \mathit{pre0}_{\mathit{Sat}(\phi\vee\psi)}(p^{\max}_{>0}(\phi\,\mathcal{U}\,\psi))$ from which (9) follows.

It therefore remains to show that (10) holds. From Definition 19 it follows that $p^{\max}_{>0}(\phi\,\mathcal{U}\,\psi) \supseteq \mathit{Sat}(\psi)$, and hence the problem reduces to demonstrating that:

$$p^{\max}_{>0}(\phi\,\mathcal{U}\,\psi) \supseteq \mathit{pre0}_{\mathit{Sat}(\phi\vee\psi)}(p^{\max}_{>0}(\phi\,\mathcal{U}\,\psi))\ .$$

Recall that, for any $s,\mathcal{E} \in S\times\mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$, we have $s,\mathcal{E} \in p^{\max}_{>0}(\phi\,\mathcal{U}\,\psi)$ if and only if there exists an adversary A such that $p^{A}_{s,\mathcal{E}}(\phi\,\mathcal{U}\,\psi){>}0$, and observe that $p^{A}_{s,\mathcal{E}}(\phi\,\mathcal{U}\,\psi){>}0$ if and only if there exists a path $\omega \in \mathit{Path}^{A}_{\mathit{ful}}(s)$ such that $\omega,\mathcal{E} \models \phi\,\mathcal{U}\,\psi$. Combining these properties we have:

$$s,\mathcal{E} \in p^{\max}_{>0}(\phi\,\mathcal{U}\,\psi) \quad\Leftrightarrow\quad \omega,\mathcal{E} \models \phi\,\mathcal{U}\,\psi \text{ for some } \omega \in \mathit{Path}_{\mathit{ful}}(s)\ . \qquad (11)$$

Now for any $s,\mathcal{E} \in \mathit{pre0}_{\mathit{Sat}(\phi\vee\psi)}(p^{\max}_{>0}(\phi\,\mathcal{U}\,\psi))$, from Definition 24 there exists a transition $(s,t,\mu) \in \mathit{TSteps}$ and state $s' \in S$ such that $\mu(s'){>}0$ and $s',\mathcal{E}{+}t \in p^{\max}_{>0}(\phi\,\mathcal{U}\,\psi)$. Since $s',\mathcal{E}{+}t \in p^{\max}_{>0}(\phi\,\mathcal{U}\,\psi)$, from (11) there exists a path $\omega \in \mathit{Path}_{\mathit{ful}}(s')$ such that $\omega,\mathcal{E}{+}t \models \phi\,\mathcal{U}\,\psi$. Letting $\omega' = s \xrightarrow{t,\mu} \omega$, from Definition 24 it follows that $s{+}t',\mathcal{E}{+}t' \in \mathit{Sat}(\phi\vee\psi)$ for all $t'{\leqslant}t$, which, in combination with the fact that $\omega,\mathcal{E}{+}t \models \phi\,\mathcal{U}\,\psi$, guarantees that $\omega',\mathcal{E} \models \phi\,\mathcal{U}\,\psi$. Given that $\omega' \in \mathit{Path}_{\mathit{ful}}(s)$, from (11) we have $s,\mathcal{E} \in p^{\max}_{>0}(\phi\,\mathcal{U}\,\psi)$, and hence since $s,\mathcal{E} \in \mathit{pre0}_{\mathit{Sat}(\phi\vee\psi)}(p^{\max}_{>0}(\phi\,\mathcal{U}\,\psi))$ was arbitrary, (10) follows.

- We next demonstrate that $p^{\max}_{>0}(\phi\,\mathcal{U}\,\psi)$ is the least fixpoint, that is, for any $Y \subseteq S\times\mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$, if $Y = \mathit{Sat}(\psi) \cup \mathit{pre0}_{\mathit{Sat}(\phi\vee\psi)}(Y)$, then $p^{\max}_{>0}(\phi\,\mathcal{U}\,\psi) \subseteq Y$. The proof is by contradiction: assume that there exists $Y \subseteq S\times\mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$ such that $Y = \mathit{Sat}(\psi) \cup \mathit{pre0}_{\mathit{Sat}(\phi\vee\psi)}(Y)$ and $p^{\max}_{>0}(\phi\,\mathcal{U}\,\psi) \setminus Y \neq \emptyset$. Now for any $s,\mathcal{E} \in p^{\max}_{>0}(\phi\,\mathcal{U}\,\psi) \setminus Y$, by construction $s,\mathcal{E} \in p^{\max}_{>0}(\phi\,\mathcal{U}\,\psi)$, and therefore from (11) there exists $\omega \in \mathit{Path}_{\mathit{ful}}(s)$ such that $\omega,\mathcal{E} \models \phi\,\mathcal{U}\,\psi$. Now using Definition 19, we have that there exists $i \in \mathbb{N}$ and $t \leqslant \mathcal{D}_\omega(i{+}1){-}\mathcal{D}_\omega(i)$ such that

- $\omega(i){+}t, \mathcal{E}{+}\mathcal{D}_\omega(i){+}t \models \psi$
- $\forall t'{<}t.\ \Big(\omega(i){+}t', \mathcal{E}{+}\mathcal{D}_\omega(i){+}t' \models \phi{\vee}\psi\Big)$
- $\forall j{<}i.\, \forall t'{\leqslant}\mathcal{D}_\omega(j{+}1){-}\mathcal{D}_\omega(j).\ \Big(\omega(j){+}t', \mathcal{E}{+}\mathcal{D}_\omega(j){+}t' \models \phi{\vee}\psi\Big)$.

Since $\mathit{Sat}(\psi) \subseteq Y$ and Y is a fixpoint, from Definition 24:

$$\forall t'{\leqslant}\ t.\ \Big(\omega(i){+}t', \mathcal{E}{+}\mathcal{D}_\omega(i){+}t' \in \mathit{pre0}_{\,\mathit{Sat}(\phi\vee\psi)}(Y)\Big)$$

and, since Y is a fixpoint, we have $\mathit{pre0}_{\,\mathit{Sat}(\phi\vee\psi)}(Y) \subseteq Y$. Repeatedly applying this fact together with Definition 24 it follows that:

$$\forall j{<}i.\, \forall t'{\leqslant}\mathcal{D}_\omega(j{+}1){-}\mathcal{D}_\omega(j).\ (\omega(j){+}t', \mathcal{E}{+}\mathcal{D}_\omega(j){+}t' \in \mathit{pre0}_{\,\mathit{Sat}(\phi\vee\psi)}(Y))\ ,$$

and therefore, since $\omega \in \mathit{Path}_{\mathit{ful}}(s)$, we have $s, \mathcal{E} \in Y$, which is a contradiction.

We conclude that $p^{\max}_{>0}(\phi\,\mathcal{U}\,\psi) = \mathit{lfp}\,Y.\,(\mathit{Sat}(\psi){\cup}\mathit{pre0}_{\,\mathit{Sat}(\phi\vee\psi)}(Y))$ as required. □

Proposition 26 *Let* PTA *be a probabilistic timed automaton with corresponding timed probabilistic system* TPS $= (S, \mathit{TSteps}, \mathcal{L}')$*, and* ϕ, ψ *be* PTCTL *formulae. The set* $\{s, \mathcal{E} \in S{\times}\mathbb{R}^{\mathcal{Z}}_{\geqslant 0} \mid p^{\max}_{s,\mathcal{E}}(\phi\,\mathcal{U}\,\psi){\geqslant}1\}$ *is given by the fixpoint*

$$\mathit{gfp}\,Y_0.\,\mathit{lfp}\,Y_1.\,\Big(\ \mathit{Sat}(\psi) \cup \mathit{pre1}_{\,\mathit{Sat}(\phi\vee\psi)}(Y_0, Y_1)\ \Big)\ .$$

Proof. We use $p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi)$ to denote the set of state and formula clock valuation pairs $\{s, \mathcal{E} \in S{\times}\mathbb{R}^{\mathcal{Z}}_{\geqslant 0} \mid p^{\max}_{s,\mathcal{E}}(\phi\,\mathcal{U}\,\psi){\geqslant}1\}$, and let

$$Z = \mathit{gfp}\,Y_0.\,\mathit{lfp}\,Y_1.\,\Big(\ \mathit{Sat}(\psi) \cup \mathit{pre1}_{\,\mathit{Sat}(\phi\vee\psi)}(Y_0, Y_1)\ \Big)\ .$$

Hence, our aim is to show that $p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi) = Z$. We split the proof into two parts: demonstrating that $Z \subseteq p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi)$ and that $Z \supseteq p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi)$.

Our first task is to show that $Z \subseteq p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi)$. Note that, because Z is a fixpoint, we have $Z = \mathit{lfp}\,Y_1.\,(\mathit{Sat}(\psi) \cup \mathit{pre1}_{\,\mathit{Sat}(\phi\vee\psi)}(Z, Y_1))$. Therefore we consider the sequence of sets of state and formula clock valuation pairs defined by $Y_1^0 = \emptyset$ and $Y_1^{i+1} = \mathit{Sat}(\psi) \cup \mathit{pre1}_{\,\mathit{Sat}(\phi\vee\psi)}(Z, Y_1^i)$, and let $i^\star = \min\{i \mid Y_1^i = Y_1^{i+1}\}$. The existence of $i^\star$ follows from the fact that $Y_1^i \subseteq Y_1^{i+1}$ and only finitely many symbolic states can be generated (see Section 4.5). Observe that $Z = Y_1^{i^\star}$ and note that because $i^\star = 0$ implies that $Z = \emptyset$ and is therefore not of interest, we henceforth assume that $i^\star{\geqslant}1$.

Clearly $\mathit{Sat}(\psi) \subseteq p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi)$ and the sets $(Y_1^{i+1} \setminus Y_1^i)$ for $1{\leqslant}i{<}i^\star$ form a partition of Z. Note that $s, \mathcal{E} \in (Y_1^{i+1}{\setminus}Y_1^i)$ implies that $s, \mathcal{E} \in \mathit{pre1}_{\,\mathit{Sat}(\phi\vee\psi)}(Z, Y_1^i)$.

This fact allows us to construct a (memoryless) adversary A in the following way, for any finite path ω_{fin}:

- if $\mathit{last}(\omega_{\mathit{fin}}), \mathcal{E} \in (Y_1^{i+1} \setminus Y_1^i)$ for some $1 \leqslant i < i^\star$ and formula clock valuation $\mathcal{E}$, let $A(\omega_{\mathit{fin}}) = (s, t, \mu)$, where (s, t, μ) is any transition satisfying the condition in the definition of $\mathit{pre1}_{\mathit{Sat}(\phi \vee \psi)}(Z, Y_1^i)$ (see Definition 24);
- if $\mathit{last}(\omega_{\mathit{fin}}), \mathcal{E} \notin (Y_1^{i+1} \setminus Y_1^i)$ for any formula clock valuation $\mathcal{E}$ let $A(\omega_{\mathit{fin}})$ be arbitrary.

Let $\lambda_{\min}$ be the minimum probability referred to in the description of the probabilistic timed automaton; that is, $\lambda_{\min} = \min_{(l,g,p,X,l') \in \mathsf{edges}} p(X, l')$. We will now show that for any $0 \leqslant i < i^\star$, if $s, \mathcal{E} \in (Y_1^{i+1} \setminus Y_1^i)$, then $p^A_{s,\mathcal{E}}(\phi \,\mathcal{U}\, \psi) \geqslant (\lambda_{\min})^i$. We proceed by induction on i.

Base case. Consider a state and formula clock valuation pair $s, \mathcal{E} \in (Y_1^1 \setminus Y_1^0)$. As stated above, $Y_1^1 = \mathit{Sat}(\psi)$ and $Y_1^0 = \emptyset$, and hence it follows from Definition 19 that $p^A_{s,\mathcal{E}}(\phi \,\mathcal{U}\, \psi) = 1 = \lambda^0_{\min}$.

Induction step. Consider any $1 \leqslant i < i^\star - 1$ and $s, \mathcal{E} \in (Y_1^{i+2} \setminus Y_1^{i+1})$ and suppose that $p^A_{s',\mathcal{E}'}(\phi \,\mathcal{U}\, \psi) \geqslant (\lambda_{\min})^i$ for all $s', \mathcal{E}' \in (Y_1^{i+1} \setminus Y_1^i)$. Since $s, \mathcal{E} \in (Y_1^{i+2} \setminus Y_1^{i+1})$, we have $s, \mathcal{E} \in \mathit{pre1}_{\mathit{Sat}(\phi \vee \psi)}(Z, Y_1^{i+1})$. By construction $A(s) = (s, t, \mu)$ such that there exists $s' \in S$ with $s', \mathcal{E}+t \in Y_1^{i+1}$, $\mu(s') > 0$ and $s+t', \mathcal{E}+t' \in \mathit{Sat}(\phi \vee \psi)$ for all $t' \leqslant t$. It then follows from the definition of $\phi \,\mathcal{U}\, \psi$ and the probability measure $\mathit{Prob}^A_{s,\mathcal{E}}$ that:

$$\begin{array}{rcll} p^A_{s,\mathcal{E}}(\phi \,\mathcal{U}\, \psi) & \geqslant & \mu(s') \cdot p^A_{s',\mathcal{E}+t}(\phi \,\mathcal{U}\, \psi) & \\ & \geqslant & \mu(s') \cdot (\lambda_{\min})^i & \text{by induction} \\ & \geqslant & \lambda_{\min} \cdot (\lambda_{\min})^i & \text{by definition of } \lambda_{\min} \\ & = & (\lambda_{\min})^{i+1} & \text{as required.} \end{array}$$

Therefore , if $0 \leqslant i < i^\star$ and $s, \mathcal{E} \in (Y_1^{i+1} \setminus Y_1^i)$, then $p^A_{s,\mathcal{E}}(\phi \,\mathcal{U}\, \psi) \geqslant (\lambda_{\min})^i$, and in particular:

$$p^A_{s,\mathcal{E}}(\phi \,\mathcal{U}\, \psi) \geqslant (\lambda_{\min})^{i^\star} \text{ for all } s, \mathcal{E} \in Z \,. \tag{12}$$

Next, observe that, for each finite path ω_{fin} ending in $Z \setminus \mathit{Sat}(\psi)$, the adversary A selects a transition $(\mathit{last}(\omega_{\mathit{fin}}), t, \mu)$ such that $\mu(Z) = 1$. Hence, unless a state satisfying ψ is reached, the adversary A chooses to remain in Z with probability 1. Therefore, for each $s, \mathcal{E} \in Z$, combing this result with (12) since $\lambda_{\min} > 0$ it follows that $p^A_{s,\mathcal{E}}(\phi \,\mathcal{U}\, \psi) = 1$, and therefore $s, \mathcal{E} \in p^{\max}_{\geqslant 1}(\phi \,\mathcal{U}\, \psi)$ as required.

It remains to show that $Z \supseteq p^{\max}_{\geqslant 1}(\phi \,\mathcal{U}\, \psi)$ and, since Z is the greatest fixpoint, it suffices to show that $p^{\max}_{\geqslant 1}(\phi \,\mathcal{U}\, \psi)$ is a fixpoint, that is:

$$p^{\max}_{\geqslant 1}(\phi \,\mathcal{U}\, \psi) = \mathit{lfp}\, Y_1 .\, \Big(\mathit{Sat}(\psi) \cup \mathit{pre1}_{\mathit{Sat}(\phi \vee \psi)}(p^{\max}_{\geqslant 1}(\phi \,\mathcal{U}\, \psi), Y_1) \Big)$$

which we prove by demonstrating that:

(a) $p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi) = \mathit{Sat}(\psi) \cup \mathit{pre1}_{\mathit{Sat}(\phi\vee\psi)}(p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi), p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi))$;

(b) for any $Y \subseteq S\times\mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$, if $Y = \mathit{Sat}(\psi) \cup \mathit{pre1}_{\mathit{Sat}(\phi\vee\psi)}(p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi), Y)$, then $p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi) \subseteq Y$.

To prove part (a) we show that:

$$p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi) \subseteq \mathit{Sat}(\psi) \cup \mathit{pre1}_{\mathit{Sat}(\phi\vee\psi)}(p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi), p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi)) \tag{13}$$
$$p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi) \supseteq \mathit{Sat}(\psi) \cup \mathit{pre1}_{\mathit{Sat}(\phi\vee\psi)}(p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi), p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi))\ . \tag{14}$$

Consider any $s,\mathcal{E} \in p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi)$. If $s,\mathcal{E} \in \mathit{Sat}(\psi)$, then $s,\mathcal{E} \in \mathit{Sat}(\psi) \cup \mathit{pre1}_{\mathit{Sat}(\phi\vee\psi)}(p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi), p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi))$. On the other hand, if $s,\mathcal{E} \notin \mathit{Sat}(\psi)$, then we must show that $s,\mathcal{E} \in \mathit{pre1}_{\mathit{Sat}(\phi\vee\psi)}(p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi), p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi))$. Let A be an adversary for which $p^{A}_{s,\mathcal{E}}(\phi\,\mathcal{U}\,\psi) = 1$, and let $A(s) = (s,t,\mu)$. By the construction of the of the probability measure for the adversary A, it follows that $\mu(p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi)) = 1$ and by Definition 19 we have $s{+}t',\mathcal{E}{+}t' \models \phi\vee\psi$ for all $t'{\leqslant}t$. Combining these two facts with Definition 24 it follows that $s,\mathcal{E} \in \mathit{pre1}_{\mathit{Sat}(\phi\vee\psi)}(p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi), p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi))$. Since these are the only cases to consider (13) holds.

Next, consider any $s,\mathcal{E} \in \mathit{Sat}(\psi) \cup \mathit{pre1}_{\mathit{Sat}(\phi\vee\psi)}(p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi), p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi))$, if $s,\mathcal{E} \in \mathit{Sat}(\psi)$ then $p^{A}_{s,\mathcal{E}}(\phi\,\mathcal{U}\,\psi) = 1$ for all adversaries A, and hence $s,\mathcal{E} \in p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi)$. On the other hand, if $s,\mathcal{E} \notin \mathit{Sat}(\psi)$, then

$$s,\mathcal{E} \in \mathit{pre1}_{\mathit{Sat}(\phi\vee\psi)}(p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi), p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi))\ ,$$

which by Definition 24 establishes the existence of a transition (s,t,μ) such that $\mu(p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi)) = 1$ and $s{+}t',\mathcal{E}{+}t' \in \mathit{Sat}(\phi\vee\psi)$ for all $t'{\leqslant}t$. Now let A be the adversary such that $A(s) = (s,t,\mu)$ and for any $s' \in \mathsf{support}(\mu)$ and $\omega_{\mathit{fin}} \in \mathit{Path}^{A'_s}_{\mathit{fin}}(s')$ we have $A(s \xrightarrow{t,\mu} \omega_{\mathit{fin}}) = A_{s'}(\omega_{\mathit{fin}})$ for all $s' \in \mathsf{support}(\mu)$ for some adversary $A_{s'}$ such that $p^{A_{s'}}_{s',\mathcal{E}+t}(\phi\,\mathcal{U}\,\psi) = 1$, and A behaves arbitrarily on all other paths. Note that, from Definition 19 and the fact that $s{+}t',\mathcal{E}{+}t' \models \phi\vee\psi$ for all $t'{\leqslant}t$, if a path $\omega = s \xrightarrow{t,\mu} \omega'$ is such that $\omega',\mathcal{E}{+}t \models \phi\,\mathcal{U}\,\psi$, then $\omega,\mathcal{E} \models \phi\,\mathcal{U}\,\psi$. By the definition of the probability measure $\mathit{Prob}^{A}_{s,\mathcal{E}}$, we have

$$p^{A}_{s,\mathcal{E}}(\phi\,\mathcal{U}\,\psi) = \sum_{s' \in \mathsf{support}(\mu)} \mu(s') \cdot p^{A^{s'}}_{s',\mathcal{E}+t}(\phi\,\mathcal{U}\,\psi)\ ,$$

since $p^{A^{s'}}_{s',\mathcal{E}+t}(\phi\,\mathcal{U}\,\psi) = 1$ for all $s' \in \mathsf{support}(\mu)$, we conclude that $p^{A}_{s,\mathcal{E}}(\phi\,\mathcal{U}\,\psi) = 1$, and hence $s,\mathcal{E} \in p^{\max}_{\geqslant 1}(\phi\,\mathcal{U}\,\psi)$. Therefore, since these are the only cases to consider, (14) follows.

We now consider part (b). The proof is by contradiction: suppose that there exists $Y \subseteq S \times \mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$ such that

$$Y = \mathit{Sat}(\psi) \cup \mathit{pre1}_{\mathit{Sat}(\phi\vee\psi)}(p^{\max}_{\geqslant 1}(\phi\, \mathcal{U}\, \psi), Y) \;\text{ and }\; p^{\max}_{\geqslant 1}(\phi\, \mathcal{U}\, \psi) \setminus Y \neq \emptyset.$$

Now, for any $s,\mathcal{E} \in p^{\max}_{\geqslant 1}(\phi\, \mathcal{U}\, \psi) \setminus Y$, since $s,\mathcal{E} \in p^{\max}_{\geqslant 1}(\phi\, \mathcal{U}\, \psi)$, there exists an adversary A such that $p^{A}_{s,\mathcal{E}}(\phi\, \mathcal{U}\, \psi) = 1$ and by Definition 19, there exists $\omega \in \mathit{Path}_{\mathit{ful}}(s)$, $i \in \mathbb{N}$ and $t \leqslant \mathcal{D}_\omega(i{+}1){-}\mathcal{D}_\omega(i)$ such that

- $\omega(i){+}t, \mathcal{E}{+}\mathcal{D}_\omega(i){+}t \models \psi$
- $\forall t'{<}t.\, \Big(\omega(i){+}t', \mathcal{E}{+}\mathcal{D}_\omega(i){+}t' \models \phi{\vee}\psi\Big)$
- $\forall j{<}i.\, \forall t'{\leqslant}\mathcal{D}_\omega(j{+}1){-}\mathcal{D}_\omega(j).\, \Big(\omega(j){+}t', \mathcal{E}{+}\mathcal{D}_\omega(j){+}t' \models \phi{\vee}\psi\Big)$.

Moreover, since $p^{A}_{s,\mathcal{E}}(\phi\, \mathcal{U}\, \psi){=}1$, by construction of the probability measure Prob^{A}_{s} it follows that:

- $\forall t'{<}t.\, \Big(\omega(i){+}t', \mathcal{E}{+}\mathcal{D}_\omega(i){+}t' \in p^{\max}_{\geqslant 1}(\phi\, \mathcal{U}\, \psi)\Big)$
- $\forall j{<}i.\, \forall t'{\leqslant}\mathcal{D}_\omega(j{+}1){-}\mathcal{D}_\omega(j).\, \Big(\omega(j){+}t', \mathcal{E}{+}\mathcal{D}_\omega(j){+}t' \in p^{\max}_{\geqslant 1}(\phi\, \mathcal{U}\, \psi)\Big)$.

Finally, since $\mathit{Sat}(\psi) \subseteq Y$ and $\mathit{pre1}_{\mathit{Sat}(\phi\vee\psi)}(p^{\max}_{\geqslant 1}(\phi\, \mathcal{U}\, \psi), Y) \subseteq Y$, using Definition 24 we have that:

- $\omega(i){+}t, \mathcal{E}{+}\mathcal{D}_\omega(i){+}t \in Y$
- $\forall t'{<}t.\, \Big(\omega(i){+}t', \mathcal{E}{+}\mathcal{D}_\omega(i){+}t' \in Y\Big)$
- $\forall j{<}i.\, \forall t'{\leqslant}\mathcal{D}_\omega(j{+}1){-}\mathcal{D}_\omega(j).\, \Big(\omega(j){+}t', \mathcal{E}{+}\mathcal{D}_\omega(j){+}t' \in Y\Big)$

and in particular, $s,\mathcal{E} \in Y$ which is a contradiction which completes the proof. □

It now remains to show that we can encode *pre0* and *pre1* using operations on symbolic states. Our approach is to first construct sub-expressions which refer to the discrete transitions of a probabilistic timed automaton only (see Definition 15 and Definition 17). Given the sets $\mathsf{Z}, \mathsf{Z}_0, \mathsf{Z}_1$ of symbolic states, let:

$$\mathsf{dpre0}(\mathsf{Z}) \overset{\text{def}}{=} \bigvee_{e \in \mathsf{edges}} \mathsf{dpre}(e, \mathsf{Z})$$

$$\mathsf{dpre1}(\mathsf{Z}_0, \mathsf{Z}_1) \overset{\text{def}}{=} \bigvee_{(l,g,p) \in \mathit{prob}} \left[\left(\bigwedge_{e \in \mathsf{edges}(l,g,p)} \mathsf{dpre}(e, \mathsf{Z}_0) \right) \wedge \left(\bigvee_{e \in \mathsf{edges}(l,g,p)} \mathsf{dpre}(e, \mathsf{Z}_1) \right) \right] .$$

Intuitively, the expression $\mathsf{dpre0}(\mathsf{Z})$ returns the symbolic states containing states which can reach a state in Z in a single discrete transition. The expression $\mathsf{dpre1}(\mathsf{Z}_0, \mathsf{Z}_1)$ returns the symbolic states containing states for which there exists an outgoing discrete transition derived from a probabilistic edge (l, g, p) for which $\mathtt{Z}_0$ is reached with probability 1 and $\mathtt{Z}_1$ is reached with probability greater than 0. The results of the expressions can be obtained using the algorithms, also called $\mathsf{dpre0}$ and $\mathsf{dpre1}$ for simplicity, shown in Figure 3. Using $\mathsf{dpre0}$ and $\mathsf{dpre1}$, we then proceed to define the following expressions, given the additional set $\mathtt{Y}$ of symbolic states:

$$\begin{aligned}\mathsf{pre0}_{\mathtt{Y}}(\mathtt{Z}) &\overset{\text{def}}{=} (\mathtt{Y} \wedge \mathsf{dpre0}(\mathtt{Z})) \vee \mathsf{tpre}_{\mathtt{Y}}(\mathtt{Z}) \\ \mathsf{pre1}_{\mathtt{Y}}(\mathtt{Z}_0, \mathtt{Z}_1) &\overset{\text{def}}{=} (\mathtt{Y} \wedge \mathsf{dpre1}(\mathtt{Z}_0, \mathtt{Z}_1)) \vee \mathsf{tpre}_{\mathtt{Y}}(\mathtt{Z}_0 {\wedge} \mathtt{Z}_1)\ .\end{aligned}$$

Resolving the expressions $\mathsf{pre0}_{\mathtt{Y}}(\mathtt{Z})$ and $\mathsf{pre1}_{\mathtt{Y}}(\mathtt{Z}_0, \mathtt{Z}_1)$ results in sets of symbolic states which correspond exactly to the state sets obtained by the functions *pre0* and *pre1*, as stated formally by the lemmas given below.

First however, we introduce the following notation. Let $\langle\!\langle\cdot\rangle\!\rangle : 2^{L\times Zones(\mathcal{X}\cup\mathcal{Z})} \rightarrow 2^{S\times\mathbb{R}^{\mathcal{Z}}_{\geqslant 0}}$ be the function which, for any set of symbolic states returns the set of state and formula clock valuation pair which these symbolic states encodes. Observe that for any sets of symbolic states $\mathtt{Y}_0, \mathtt{Y}_1 \subseteq L \times Zones(\mathcal{X} \cup \mathcal{Z})$, we have $\langle\!\langle \mathtt{Y}_0 \wedge \mathtt{Y}_1 \rangle\!\rangle = \langle\!\langle \mathtt{Y}_0 \rangle\!\rangle \cap \langle\!\langle \mathtt{Y}_1 \rangle\!\rangle$ and $\langle\!\langle \mathtt{Y}_0 \vee \mathtt{Y}_1 \rangle\!\rangle = \langle\!\langle \mathtt{Y}_0 \rangle\!\rangle \cup \langle\!\langle \mathtt{Y}_1 \rangle\!\rangle$.

Lemma 27 *If* $\mathtt{Y}, \mathtt{Z} \subseteq L \times Zones(\mathcal{X} \cup \mathcal{Z})$ *are symbolic states encoding the sets* $Y, Z \subseteq S\times\mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$ *of state and formula clock valuation pairs, then* $\mathsf{pre0}_{\mathtt{Y}}(\mathtt{Z})$ *encodes the set* $pre0_Y(Z)$*.*

Proof. Consider any sets of state and formula clock valuation pairs $Y, Z \subseteq S\times\mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$ and suppose that the sets of symbolic states $\mathtt{Y}, \mathtt{Z} \subseteq L \times Zones(\mathcal{X} \cup \mathcal{Z})$ encode Y and Z. Since $\mathsf{pre0}_{\mathtt{Y}}(\mathtt{Z}) = (\mathtt{Y} \wedge \mathsf{dpre0}(\mathtt{Z})) \vee \mathsf{tpre}_{\mathtt{Y}}(\mathtt{Z})$, using the definition of dpre we can split the proof into two parts showing that:

$$pre0_Y(Z) \subseteq \left(\langle\!\langle \mathtt{Y} \rangle\!\rangle \cap \left(\bigcup_{e\in\mathsf{edges}} \langle\!\langle \mathsf{dpre}(e, \mathtt{Z}) \rangle\!\rangle\right)\right) \cup \langle\!\langle \mathsf{tpre}_{\mathtt{Y}}(\mathtt{Z}) \rangle\!\rangle \qquad (15)$$

$$pre0_Y(Z) \supseteq \left(\langle\!\langle \mathtt{Y} \rangle\!\rangle \cap \left(\bigcup_{e\in\mathsf{edges}} \langle\!\langle \mathsf{dpre}(e, \mathtt{Z}) \rangle\!\rangle\right)\right) \cup \langle\!\langle \mathsf{tpre}_{\mathtt{Y}}(\mathtt{Z}) \rangle\!\rangle\ . \qquad (16)$$

We begin by showing that (15) holds. For any $s, \mathcal{E} \in pre0_Y(Z)$, by Definition 24 there exist $(s, t, \mu) \in TSteps$ and $s' \in S$ such that $s', \mathcal{E}{+}t \in Z$, $\mu(s'){>}0$ and $s{+}t', \mathcal{E}{+}t' \in Y \cup Z$ for all $t'{\leqslant}t$. We consider two cases, depending on whether (s, t, μ) is generated from the *discrete* or *timed* transition rule of Definition 17.

- If (s, t, μ) is derived from the discrete transition rule, then $t{=}0$, and hence $s, \mathcal{E} \in Y{\cup}Z$. If $s, \mathcal{E} \in Z(=\langle\!\langle \mathtt{Z} \rangle\!\rangle)$, then $s, \mathcal{E} \in \langle\!\langle \mathsf{tpre}_{\mathtt{Y}}(\mathtt{Z}) \rangle\!\rangle$ and (15) follows. It therefore remains to consider the case when $s, \mathcal{E} \in Y(=\langle\!\langle \mathtt{Y} \rangle\!\rangle)$. Suppose that $s = (l,v)$ and $((l,v),0,\mu)$ is generated from $(l,g,p) \in \mathit{prob}$. Since $(l',v'),\mathcal{E} \in Z$ and $\mu(l',v'){>}0$ for some $(l',v') \in S$, from Definition 17 there exists an edge $(l,g,p,X,l') \in \mathsf{edges}(l,g,p)$ such that $(l',v[X{:=}0]),\mathcal{E} \in Z$ and $v \rhd g$. Since $(l,v) \in S$, we have that $v \rhd \mathit{inv}(l)$, and hence $v \rhd g \wedge \mathit{inv}(l)$. Now, since $X \subseteq \mathcal{X}$ we have:

$$\begin{aligned} v,\mathcal{E} \rhd [X{:=}0]\zeta_{\mathsf{Z}}^{l'} &\Leftrightarrow v[X{:=}0],\mathcal{E} \rhd \zeta_{\mathsf{Z}}^{l'} \\ &\Leftrightarrow (l',v[X{:=}0]),\mathcal{E} \in (l',\zeta_{\mathsf{Z}}^{l'}) && \text{rearranging} \\ &\Leftrightarrow (l',v[X{:=}0]),\mathcal{E} \in \langle\!\langle \mathsf{Z} \rangle\!\rangle && \text{by definition of } \zeta_{\mathsf{Z}}^{l'}. \end{aligned}$$

By definition of dpre and the hypothesis that $\langle\!\langle \mathsf{Z} \rangle\!\rangle{=}Z$ it follows that:

$$\begin{aligned} &\langle\!\langle \mathsf{dpre}((l,g,p,X,l'),\mathsf{Z}) \rangle\!\rangle = \\ &\{(l,v),\mathcal{E} \mid (v \rhd g \wedge \mathit{inv}(l)) \wedge (l',v[X{:=}0]),\mathcal{E} \in Z\}\ . \end{aligned} \tag{17}$$

Therefore, $(l,v),\mathcal{E} \in \langle\!\langle \mathsf{dpre}((l,g,p,X,l'),\mathsf{Z}) \rangle\!\rangle$ and combing this with the fact that $s,\mathcal{E} \in Y{=}\langle\!\langle \mathsf{Y} \rangle\!\rangle$ it follows that (15) holds in this case.

- We now consider the case when (s,t,μ) is derived from the timed transition rule. Let $s = (l,v)$. From Definition 17 and Definition 24 it follows that $(l,v{+}t),\mathcal{E}{+}t \in Z$, and $(l,v{+}t'),\mathcal{E}{+}t' \in Y{\cup}Z$ for all $t'{\leqslant}t$. By definition of $\swarrow_{\zeta'} \zeta$ (see Section 3.1):

$$\begin{aligned} \swarrow_{\zeta_{\mathsf{Y}}^{l} \wedge \mathit{inv}(l)} (\zeta_{\mathsf{Z}}^{l} {\wedge} \mathit{inv}(l)) &= \Big\{ v',\mathcal{E}' \,\Big|\, \exists t{\geqslant}0. \big(v'{+}t,\mathcal{E}'{+}t \rhd (\zeta_{\mathsf{Z}}^{l} \wedge \mathit{inv}(l)) \\ &\qquad \wedge \forall t'{\leqslant}t.\ (v'{+}t',\mathcal{E}'{+}t' \rhd (\zeta_{\mathsf{Y}}^{l} {\vee} \zeta_{\mathsf{Z}}^{l}) \wedge \mathit{inv}(l)) \big) \Big\} \\ &= \Big\{ v',\mathcal{E}' \,\Big|\, \exists t{\geqslant}0. \big((l,v'{+}t),\mathcal{E}'{+}t \in Z \wedge v'{+}t \rhd \mathit{inv}(l)) \\ &\qquad \wedge \forall t'{\leqslant}t.\ ((l,v'{+}t'),\mathcal{E}'{+}t' \in (Y \cup Z) \wedge v'{+}t' \rhd \mathit{inv}(l)) \big) \Big\} \end{aligned}$$

where the final step follows from the fact that $v,\mathcal{E} \rhd \zeta_{\mathsf{Y}}^{l}$ if and only if $((l,v),\mathcal{E}) \in Y$ (and similarly for ζ_{Z}^{l} and Z). Now, since $(l,v{+}t),\mathcal{E}{+}t \in Z$ and $s{+}t',\mathcal{E}{+}t' \in Y \cup Z$ for all $t'{\leqslant}t$, we have $v,\mathcal{E} \in \swarrow_{\zeta_{\mathsf{Y}}^{l} \wedge \mathit{inv}(l)} (\zeta_{\mathsf{Z}}^{l} {\wedge} \mathit{inv}(l))$ and (15) follows from the definition of $\mathsf{tpre}_{\mathsf{Y}}(\mathsf{Z})$.

Since these are the only possible cases we conclude that (15) holds.

It therefore remains to show that (16) holds. Now for any $(l,v),\mathcal{E} \in (\langle\!\langle \mathsf{Y} \rangle\!\rangle \cap (\cup_{e \in \mathsf{edges}} \langle\!\langle \mathsf{dpre}(e,\mathsf{Z}) \rangle\!\rangle)) \cup \langle\!\langle \mathsf{tpre}_{\mathsf{Y}}(\mathsf{Z}) \rangle\!\rangle$, again we split the proof into two cases.

- If $(l,v),\mathcal{E} \in \langle\!\langle \mathsf{Y} \rangle\!\rangle {\cap} (\cup_{e \in \mathsf{edges}} \langle\!\langle \mathsf{dpre}(e,\mathsf{Z}) \rangle\!\rangle)$, then there exists $e = (l,g,p,X,l') \in \mathsf{edges}$ such that $(l,v),\mathcal{E} \in \langle\!\langle \mathsf{Y} \rangle\!\rangle \cap \langle\!\langle \mathsf{dpre}(e,\mathsf{Z}) \rangle\!\rangle$. Using Definition 17, (17) and since we assume the probabilistic timed automaton is well-formed we can use (l,g,p) to construct a probabilistic transition $((l,v),0,\mu) \in \mathit{TSteps}$. To show that $s,\mathcal{E} \in \mathit{pre0}_{Y}(Z)$, and hence that (16) holds in this case, from Definition 24 and the fact that $(i,v),\mathcal{E} \in \langle\!\langle \mathsf{Y} \rangle\!\rangle ({=}Y)$ it is sufficient to show that:

$$\exists (l'',v'') \in S.\ ((l'',v''),\mathcal{E} \in Z \wedge \mu(l'',v''){>}0)\ .$$

Since $(l,v),\mathcal{E} \in \langle\!\langle \mathsf{dpre}(e,\mathsf{Z})\rangle\!\rangle$ and $e \in \mathsf{edges}$ and using Definition 17 we have $(l',v[X{:=}0]),\mathcal{E} \in Z$ and $\mu(l',v[X{:=}0]){>}0$ as required.

- If $(l,v),\mathcal{E} \in \langle\!\langle \mathsf{tpre}_\mathsf{Y}(\mathsf{Z})\rangle\!\rangle$, then $v,\mathcal{E} \in \swarrow_{\zeta^l_\mathsf{Y}\wedge inv(l)} (\zeta^l_\mathsf{Z}\wedge inv(l))$, and it follows that there exists $t{\geqslant}0$ such that $(l,v{+}t),\mathcal{E}{+}t \in Z$, $(l,v{+}t'),\mathcal{E}{+}t' \in Y \cup Z$ and $v+t' \triangleright inv(l)$ for all $t'{\leqslant}t$. Now, from the construction of time transitions in Definition 17, it follows that $((l,v),t,\mu_{(l,v+t)}) \in \mathit{TSteps}$, and hence using this probabilistic transition in Definition 24 we have $(l,v),\mathcal{E} \in \mathit{pre0}_Y(Z)$. □

Lemma 28 *If* $\mathsf{Y},\mathsf{Z}_0,\mathsf{Z}_1 \subseteq L \times \mathit{Zones}(\mathcal{X} \cup \mathcal{Z})$ *are symbolic states encoding the sets* $Y,Z_0,Z_1 \subseteq S{\times}\mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$ *of state and formula clock valuation pairs, then* $\mathsf{pre1}_\mathsf{Y}(\mathsf{Z}_0,\mathsf{Z}_1)$ *encodes the set* $\mathit{pre1}_Y(Z_0,Z_1)$.

Proof. Consider any sets of state and formula clock valuation pairs $Y,Z_0,Z_1 \subseteq S{\times}\mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$ and suppose that the sets of symbolic states $\mathsf{Y},\mathsf{Z}_0,\mathsf{Z}_1 \subseteq L \times \mathit{Zones}(\mathcal{X} \cup \mathcal{Z})$ encode Y, Z_0 and Z_1. By definition of $\mathsf{pre1}$it is sufficient show that:

$$\mathit{pre1}_Y(Z_0,Z_1) \subseteq (\langle\!\langle \mathsf{Y}\rangle\!\rangle \cap \langle\!\langle \mathsf{dpre1}(\mathsf{Z}_0,\mathsf{Z}_1)\rangle\!\rangle) \cup \langle\!\langle \mathsf{tpre}_\mathsf{Y}(\mathsf{Z}_0{\wedge}\mathsf{Z}_1)\rangle\!\rangle \tag{18}$$
$$\mathit{pre1}_Y(Z_0,Z_1) \supseteq (\langle\!\langle \mathsf{Y}\rangle\!\rangle \cap \langle\!\langle \mathsf{dpre1}(\mathsf{Z}_0,\mathsf{Z}_1)\rangle\!\rangle) \cup \langle\!\langle \mathsf{tpre}_\mathsf{Y}(\mathsf{Z}_0{\wedge}\mathsf{Z}_1)\rangle\!\rangle \tag{19}$$

where

$$\mathsf{dpre1}(\mathsf{Z}_0,\mathsf{Z}_1) = \bigcup_{(l,g,p)\in prob} \left[\left(\bigcap_{e\in\mathsf{edges}(l,g,p)} \langle\!\langle \mathsf{dpre}(e,\mathsf{Z}_0)\rangle\!\rangle \right) \cap \left(\bigcup_{e\in\mathsf{edges}(l,g,p)} \langle\!\langle \mathsf{dpre}(e,\mathsf{Z}_1)\rangle\!\rangle \right) \right] .$$

Considering (18), for any $s,\mathcal{E} \in \mathit{pre0}_Y(Z_0,Z_1)$ by Definition 24 there exists $(s,t,\mu) \in \mathit{TSteps}$ such that:

- for all $s' \in S$, if $\mu(s') > 0$ then $s',\mathcal{E}{+}t \in Z_0$;
- there exists $s' \in S$ such that $s',\mathcal{E}{+}t \in Z_1$ and $\mu(s'){>}0$;
- $s{+}t',\mathcal{E}{+}t' \in Y{\cup}(Z_0 \cap Z_1)$ for all $t'{\leqslant}t$.

We consider two cases, depending on whether (s,t,μ) is derived from the discrete transition or timed transition rule of Definition 17.

- The case of timed transitions is similar to that considered in the proof of Lemma 27, by substituting $Z_0 \cap Z_1$ for Y', and $\mathsf{Z}_0 \wedge \mathsf{Z}_1$ for Y'.
- If (s,t,μ) is derived from the discrete transition rule, then $t{=}0$. If $s,\mathcal{E} \in Z_0{\cap}Z_1$, then the result follows from the fact that $Z_0{\cap}Z_1 \subseteq \langle\!\langle \mathsf{tpre}_\mathsf{Y}(\mathsf{Z}_0{\wedge}\mathsf{Z}_1)\rangle\!\rangle$. It therefore remains to consider the case when $s,\mathcal{E} \in Y$. Supposing $s = (l,v)$, and $((l,v),0,\mu)$ is generated from $(l,g,p) \in prob$, since $Y{=}\langle\!\langle \mathsf{Y}\rangle\!\rangle$ it is sufficient to show that:

$$(l,v),\mathcal{E} \in \left(\bigcap_{e\in\mathsf{edges}(l,g,p)} \langle\!\langle \mathsf{dpre}(e,\mathsf{Z}_0)\rangle\!\rangle \right) \cap \left(\bigcup_{e\in\mathsf{edges}(l,g,p)} \langle\!\langle \mathsf{dpre}(e,\mathsf{Z}_1)\rangle\!\rangle \right) .$$

The arguments for demonstrating that $(l,v),\mathcal{E} \in \bigcup_{e\in\mathsf{edges}(l,g,p)} \langle\!\langle \mathsf{dpre}(e,\mathsf{Z}_1)\rangle\!\rangle$ are similar to those used in analogous result in the proof of Lemma 27. We prove that $(l,v),\mathcal{E} \in \bigcap_{e\in\mathsf{edges}(l,g,p)} \langle\!\langle \mathsf{dpre}(e,\mathsf{Z}_0)\rangle\!\rangle$ by contradiction. Therefore assume $(l,v),\mathcal{E} \not\in \langle\!\langle \mathsf{dpre}(e,\mathsf{Z}_0)\rangle\!\rangle$ for some $e{=}(l,g,p,X,l') \in \mathsf{edges}(l,g,p)$. Using (17) and the fact that $((l,v),0,\mu)$ is derived from (l,g,p), it follows that $(l',v[X{:=}0]),\mathcal{E} \not\in Z_0$. However, again because $((l,v),0,\mu)$ is derived from (l,g,p), we have $\mu(l',v[X{:=}0]){>}0$, and from the hypothesis it follows that $s',\mathcal{E} \in Z_0$ which is a contradiction.

It therefore remains to show that (19) holds. The proof is again split into two cases: when $(l,v),\mathcal{E} \in \langle\!\langle \mathsf{tpre}_{\mathsf{Y}}(\mathsf{Z}_0{\wedge}\mathsf{Z}_1)\rangle\!\rangle$ which can be dealt with in the same manner as in the proof of Lemma 27, and when $(l,v),\mathcal{E} \in \langle\!\langle \mathsf{Y}\rangle\!\rangle{\cap}\langle\!\langle \mathsf{dpre1}(\mathsf{Z}_0,\mathsf{Z}_1)\rangle\!\rangle$ which is demonstrated below.

Consider any $(l,v),\mathcal{E} \in \langle\!\langle \mathsf{Y}\rangle\!\rangle \cap \langle\!\langle \mathsf{dpre1}(\mathsf{Z}_0,\mathsf{Z}_1)\rangle\!\rangle$. From Definition 24 and since $(l,v),\mathcal{E} \in \langle\!\langle \mathsf{Y}\rangle\!\rangle({=}Y)$ to prove that $(l,v),\mathcal{E} \in \mathit{pre1}_Y(Z_0,Z_1)$ it is sufficient to show that there exists $((l,v),0,\mu) \in \mathit{TSteps}$ such that the following two conditions are satisfied:

(1) for all $s' \in S$, if $\mu(s'){>}0$ then $s',\mathcal{E} \in Z_0$;
(2) there exists $s' \in S$ such that $s',\mathcal{E} \in Z_1$ and $\mu(s'){>}0$.

Since $(l,v),\mathcal{E} \in \langle\!\langle \mathsf{dpre1}(\mathsf{Z}_0,\mathsf{Z}_1)\rangle\!\rangle$, by definition of $\mathsf{dpre1}$ there exists $(l,g,p) \in \mathit{prob}$ such that

$$(l,v),\mathcal{E} \in \cap_{e\in\mathsf{edges}(l,g,p)} \langle\!\langle \mathsf{dpre}(e,\mathsf{Z}_0)\rangle\!\rangle \;\text{ and }\; (l,v),\mathcal{E} \in \cup_{e\in\mathsf{edges}(l,g,p)} \langle\!\langle \mathsf{dpre}(e,\mathsf{Z}_1)\rangle\!\rangle\;.$$

Since we assume the probabilistic timed automaton is well-formed, using Definition 17 there exists $((l,v),0,\mu) \in \mathit{TSteps}$ generated from (l,g,p). We proceed by showing that this probabilistic transition satisfies the two conditions given above.

(1) Assume that $\mu(s'){>}0$ and $s',\mathcal{E} \not\in Z_0$ for some $s' \in S$. From Definition 17 there exists $e = (l,g,p,X,l') \in \mathsf{edges}(l,g,p)$ such that $s' = (l',v[X{:=}0])$. Because $(l',v[X{:=}0]),\mathcal{E} \not\in Z_0$, we have that $(l,v),\mathcal{E} \not\in \{(l,v),\mathcal{E} \mid (v \rhd g \wedge \mathit{inv}(l)) \wedge (v,\mathcal{E} \rhd [X{:=}0]\zeta^{l'}_{\mathsf{Z}_0})\}$. Using (17) it follows that $(l,v),\mathcal{E} \not\in \langle\!\langle \mathsf{dpre}(e,\mathsf{Z}_0)\rangle\!\rangle$. Hence $(l,v),\mathcal{E} \not\in \bigcap_{e\in\mathsf{edges}(l,g,p)} \langle\!\langle \mathsf{dpre}(e,\mathsf{Z}_0)\rangle\!\rangle$ which is a contradiction.
(2) The argument in this case proceeds in a similar manner to the analogous part of the proof of Lemma 27.

As these are the only case to consider (12) holds which completes the proof. □

It remains to embed the expressions $\mathsf{pre0}_{\mathsf{Y}}(\mathsf{Y}')$ and $\mathsf{pre1}_{\mathsf{Y}}(\mathsf{Y}_0,\mathsf{Y}_1)$ within the fixpoints given in Proposition 25 and Proposition 26 respectively. Figure 4

```
algorithm $\mathsf{MaxU}_{>0}(\mathtt{U},\mathtt{V})$

$\mathtt{Z} := [\![\mathtt{false}]\!]$
repeat
   $\mathtt{Y} := \mathtt{Z}$
   $\mathtt{Z} := \mathtt{V} \vee (\mathtt{U} \wedge \mathsf{dpre0}(\mathtt{Y}))$
   $\mathtt{Z} := \mathtt{Z} \vee \mathsf{tpre}_{\mathtt{U}\vee\mathtt{V}}(\mathtt{Y})$
until $\mathtt{Z} = \mathtt{Y}$
return $\mathtt{Z}$
```

```
algorithm $\mathsf{MaxU}_{\geqslant 1}(\mathtt{U},\mathtt{V})$

$\mathtt{Z}_0 := [\![\mathtt{true}]\!]$
repeat
   $\mathtt{Y}_0 := \mathtt{Z}_0$
   $\mathtt{Z}_1 := [\![\mathtt{false}]\!]$
   repeat
      $\mathtt{Y}_1 := \mathtt{Z}_1$
      $\mathtt{Z}_1 := \mathtt{V} \vee (\mathtt{U} \wedge \mathsf{dpre1}(\mathtt{Y}_0,\mathtt{Y}_1))$
      $\mathtt{Z}_1 := \mathtt{Z}_1 \vee \mathsf{tpre}_{\mathtt{U}\vee\mathtt{V}}(\mathtt{Y}_0 \wedge \mathtt{Y}_1)$
   until $\mathtt{Z}_1 = \mathtt{Y}_1$
   $\mathtt{Z}_0 := \mathtt{Z}_1$
until $\mathtt{Z}_0 = \mathtt{Y}_0$
return $\mathtt{Z}_0$
```

Fig. 4. $\mathsf{MaxU}_{>0}$ and $\mathsf{MaxU}_{\geqslant 1}$ algorithms

presents algorithms for computing these fixpoints using operations on symbolic states, with $\mathsf{MaxU}_{>0}([\![\phi]\!],[\![\psi]\!])$ therefore corresponding to $\{s,\mathcal{E} \,|\, p^{\max}_{s,\mathcal{E}}(\phi\,\mathcal{U}\,\psi){>}0\}$, and $\mathsf{MaxU}_{\geqslant 1}([\![\phi]\!],[\![\psi]\!])$ corresponding to $\{s,\mathcal{E} \,|\, p^{\max}_{s,\mathcal{E}}(\phi\,\mathcal{U}\,\psi){\geqslant}1\}$.

Using these results we set:

$$\begin{aligned}
\mathsf{Until}([\![\phi]\!],[\![\psi]\!],\leqslant 0) &\stackrel{\text{def}}{=} [\![\mathtt{true}]\!] \setminus \mathsf{MaxU}_{>0}([\![\phi]\!],[\![\psi]\!])\\
\mathsf{Until}([\![\phi]\!],[\![\psi]\!],< 1) &\stackrel{\text{def}}{=} [\![\mathtt{true}]\!] \setminus \mathsf{MaxU}_{\geqslant 1}([\![\phi]\!],[\![\psi]\!])\\
\mathsf{Release}([\![\phi]\!],[\![\psi]\!],> 0) &\stackrel{\text{def}}{=} [\![\mathtt{true}]\!] \setminus \mathsf{MaxU}_{\geqslant 1}([\![\neg\phi]\!],[\![\neg\psi]\!])\\
\mathsf{Release}([\![\phi]\!],[\![\psi]\!],\geqslant 1) &\stackrel{\text{def}}{=} [\![\mathtt{true}]\!] \setminus \mathsf{MaxU}_{>0}([\![\neg\phi]\!],[\![\neg\psi]\!])\,.
\end{aligned}$$

4.2.2 The Quantitative Case

In the case of computing *quantitative* maximum probabilities of until path formulae we use the algorithm $\mathsf{MaxU}(\cdot,\cdot,\gtrsim\lambda)$ given in Figure 5. The algorithm iteratively applies timed-predecessor, discrete-predecessor and *conjunction* operations on symbolic states until a fixpoint is reached. The key observation is that to preserve the probabilistic branching one must take the conjunctions of symbolic states generated by edges from the same distribution. More precisely, one needs to identify the state sets from which *multiple edges* within the support of the *same distribution* of the probabilistic timed automaton can be used to reach previously generated state sets. Upon termination of the fixpoint algorithm, the set of generated symbolic states is used to construct a finite-state probabilistic system which has sufficient information to compute the maximum probability of interest using well-established finite-state probabilistic model checking methods [17].

```
algorithm $\mathsf{MaxU}(\mathtt{U}, \mathtt{V}, \gtrsim \lambda)$

1.  $\mathtt{Z} := \mathsf{tpre}_{\mathtt{U}\vee\mathtt{V}}(\mathtt{V})$
2.  for $(l, g, p) \in \mathit{prob}$
3.    $E_{(l,g,p)} := \emptyset$
4.  end for
5.  repeat
6.    $\mathtt{Y} := \mathtt{Z}$
7.    for $\mathtt{y} \in \mathtt{Y} \wedge (l, g, p) \in \mathit{prob} \wedge e = (l, g, p, X, l') \in \mathsf{edges}(l, g, p)$
8.      $\mathsf{z} := \mathtt{U} \wedge \mathsf{dpre}(e, \mathsf{tpre}_{\mathtt{U}\vee\mathtt{V}}(\mathtt{y}))$
9.      if $(\mathsf{z} \neq \emptyset) \wedge (\mathsf{z} \notin \mathsf{tpre}_{\mathtt{U}\vee\mathtt{V}}(\mathtt{V}))$
10.       $\mathtt{Z} := \mathtt{Z} \cup \{\mathsf{z}\}$
11.       $E_{(l,g,p)} := E_{(l,g,p)} \cup \{(\mathsf{z}, (X, l'), \mathtt{y})\}$
12.       for $(\bar{\mathsf{z}}, (\bar{X}, \bar{l}'), \bar{\mathtt{y}}) \in E_{(l,g,p)}$
13.         if $(\mathsf{z} \wedge \bar{\mathsf{z}} \neq \emptyset) \wedge ((\bar{X}, \bar{l}') \neq (X, l')) \wedge (\mathsf{z} \wedge \bar{\mathsf{z}} \notin \mathsf{tpre}_{\mathtt{U}\vee\mathtt{V}}(\mathtt{V}))$
14.           $\mathtt{Z} := \mathtt{Z} \cup \{\mathsf{z} \wedge \bar{\mathsf{z}}\}$
15.         end if
16.       end for
17.     end if
18.   end for
19. until $\mathtt{Z} = \mathtt{Y}$
20. construct $\mathsf{PS} = (\mathtt{Z}, \mathit{Steps})$ where $(\mathsf{z}, \rho) \in \mathit{Steps}$ if and only if
        there exists $(l, g, p) \in \mathit{prob}$ and $E \subseteq E_{(l,g,p)}$ such that
          – $\mathsf{z} \in \{\mathsf{z}' \mid (\mathsf{z}', \mathsf{e}, \mathsf{z}'') \in E\}$
          – $(\mathsf{z}', \mathsf{e}, \mathsf{z}'') \in E \;\Rightarrow\; \mathsf{z}' \supseteq \mathsf{z}$
          – $(\mathsf{z}'_1, \mathsf{e}, \mathsf{z}') \neq (\mathsf{z}'_2, \mathsf{e}', \mathsf{z}'') \in E \;\Rightarrow\; \mathsf{e} \neq \mathsf{e}'$
          – $E$ is maximal
          – $\rho(\mathsf{z}') = \sum\{\!|\, p(X, l') \mid (\mathsf{z}, (X, l'), \mathsf{z}') \in E \,|\!\} \;\; \forall \mathsf{z}' \in \mathtt{Z}$
21. return $\bigvee\{\mathsf{tpre}_{\mathtt{U}\vee\mathtt{V}}(\mathsf{z}) \mid \mathsf{z} \in \mathtt{Z} \wedge p^{\max}_{\mathsf{z}}(\Diamond\, \mathsf{tpre}_{\mathtt{U}\vee\mathtt{V}}(\mathtt{V})) \gtrsim \lambda\}$
```

Fig. 5. Algorithm $\mathsf{MaxU}(\cdot, \cdot, \gtrsim \lambda)$

We now explain the algorithm $\mathsf{MaxU}(\cdot, \cdot, \gtrsim \lambda)$ in more detail. Let $\phi\, \mathcal{U}\, \psi$ be the until path formula of interest. Then the parameters of the algorithm are $\mathtt{U} = [\![\phi]\!]$, $\mathtt{V} = [\![\psi]\!]$, $\gtrsim \in \{\geqslant, >\}$, and $\lambda \in [0, 1]$. Lines 1–4 deal with the initialisation of $\mathtt{Z}$, which is set equal to the set of time predecessors of $\mathtt{V}$, and the set of edges $E_{(l,g,p)}$ associated with each probabilistic edge $(l, g, p) \in \mathit{prob}$. Lines 5–20 generate a finite-state graph, the nodes of which are symbolic states, obtained by iterating timed and discrete predecessor operations (line 8), and taking conjunctions (lines 12–16). The edges of the graph are partitioned into the sets $E_{(l,g,p)}$ for $(l, g, p) \in \mathit{prob}$, with the intuition that $(\mathsf{z}, (X, l'), \mathsf{z}') \in E_{(l,g,p)}$ corresponds to a transition from any state in the symbolic state z to some state in the symbolic state z' when the outcome (X, l') of the probabilistic edge (l, g, p) is chosen. The graph edges are added in lines 11. Line 20 describes the manner in which the probabilistic edges of the probabilistic timed automaton are used in combination with the computed edge sets to construct the probabilistic system PS. The states of PS are the symbolic states generated by the previous steps of the algorithm, and the probabilistic transition relation of PS is constructed by grouping the graph edges generated by the same probabilistic edge of the probabilistic timed automaton under study. Finally, in line 21, the maximum probability of reaching $\mathsf{tpre}_{\mathtt{U}\vee\mathtt{V}}(\mathtt{V})$ is computed for each $\mathtt{z} \in \mathtt{Z}$. Note that we write $\mathtt{z} \neq \emptyset$ if and only if $\mathtt{z}$ encodes at least one state and formula clock valuation pair.

Note that the probabilistic transitions $(\mathtt{z}, \rho) \in \mathit{Steps}$ could feature *sub-distributions*, which are distributions for which $\sum_{\mathtt{z}' \in \mathtt{Z}} \rho(\mathtt{z}') < 1$. The computation of maximum reachability properties can also be performed on finite-state systems with sub-distributions. The following proposition states the correctness of our algorithm.

Proposition 29 *For any probabilistic timed automaton* PTA*, corresponding timed probabilistic system* $\mathsf{TPS} = (S, \mathit{TSteps}, \mathcal{L}')$*,* PTCTL *formulae* ϕ *and* ψ*,* $\gtrsim \in \{\geqslant, >\}$ *and* $\lambda \in [0,1]$*, if* $\mathsf{PS} = (\mathtt{Z}, \mathit{Steps})$ *is the probabilistic system generated by* $\mathsf{MaxU}([\![\phi]\!], [\![\psi]\!], \gtrsim \lambda)$*, then for any* $s, \mathcal{E} \in S \times \mathbb{R}_{\geqslant 0}^{\mathcal{Z}}$:

- $p_{s,\mathcal{E}}^{\max}(\phi\, \mathcal{U}\, \psi) > 0$ *if and only if* $s, \mathcal{E} \in \mathsf{tpre}_{[\![\phi\vee\psi]\!]}(\mathtt{Z})$*;*
- *if* $p_{s,\mathcal{E}}^{\max}(\phi\, \mathcal{U}\, \psi) > 0$*, then* $p_{s,\mathcal{E}}^{\max}(\phi\, \mathcal{U}\, \psi)$ *equals*

$$\max \left\{ p_{\mathtt{z}}^{\max}(\Diamond\, \mathsf{tpre}_{[\![\phi\vee\psi]\!]}[\![\psi]\!]) \;\middle|\; \mathtt{z} \in \mathtt{Z} \textit{ and } s, \mathcal{E} \in \mathsf{tpre}_{[\![\phi\vee\psi]\!]}(\mathtt{z}) \right\} .$$

Before we give the proof we require a number of definitions and lemmas. First we define for any adversary A and finite path ω, an adversary, denoted $A[\omega]$, which acts essentially as A assuming that the path ω has already occurred.

Definition 30 *For a probabilistic system* $\mathsf{PS} = (S, \mathit{Steps}, \mathcal{L})$*, adversary* A *of* PS *and finite path* ω*, let* $A[\omega]$ *be the adversary such that for any finite path* ω' *of* PS*:*

$$A[\omega](\omega') \stackrel{\text{def}}{=} \begin{cases} A(\omega \xrightarrow{\mu} \omega'') & \textit{if } \omega' \textit{ is of the form } \mathit{last}(\omega) \xrightarrow{\mu} \omega'' \\ A(\omega') & \textit{otherwise.} \end{cases}$$

Next, for any adversary A of TPS we introduce the sequence of functions $\langle U_n^A \rangle_{n \in \mathbb{N}}$. Intuitively, for $s, \mathcal{E} \in S \times \mathbb{R}_{\geqslant 0}^{\mathcal{Z}}$, the value $U_n^A(\phi, \psi, s, \mathcal{E})$ equals the probability of reaching from $s, \mathcal{E}$, under the adversary A, a state which satisfies ψ in *at most* n transitions, while passing through only states satisfying ϕ. Since adversaries can choose on the basis of history, we first define U_n^A more generally, mapping from paths rather than states, then restrict to the case of states (paths of length 0).

Definition 31 *For any, adversary $A \in Adv_{\mathsf{TPS}}$, $\mathcal{E} \in \mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$ and finite path $\omega \in Path^A_{fin}$ where $last(\omega) = (l, v)$ and $A(\omega) = (t, \mu)$:*

- *if there exists $t' \leqslant t$ such that $(l, v+t'), \mathcal{E}+t' \models \psi$ and $(l, v+t''), \mathcal{E}+t'' \models \phi \vee \psi$ for all $t'' \leqslant t'$, then $U^A_0(\phi, \psi, (l, v), \mathcal{E}) = 1$;*
- *otherwise, $U^A_0(\phi, \psi, \omega, \mathcal{E}) = 0$;*

and for any $n \geqslant 0$:

- *if there exists $t' \leqslant t$ such that $(l, v+t'), \mathcal{E}+t' \models \psi$ and $(l, v+t''), \mathcal{E}+t'' \models \phi \vee \psi$ for all $t'' \leqslant t'$, then $U^A_{n+1}(\phi, \psi, \omega, \mathcal{E}) = 1$;*
- *else if $(l, v+t'), \mathcal{E} + t' \models \phi \wedge \neg\psi$ for all $t' \leqslant t$, then*

$$U^A_{n+1}(\phi, \psi, \omega, \mathcal{E}) = \sum_{(l',v') \in S} \mu(l', v') \cdot U^A_n(\phi, \psi, \omega \xrightarrow{t,\mu} (l', v'), \mathcal{E}+t);$$

- *otherwise, $U^A_{n+1}(\phi, \psi, \omega, \mathcal{E}) = 0$.*

Lemma 32 *For any $A \in Adv_{\mathsf{TPS}}$ and $s, \mathcal{E} \in S \times \mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$: $\langle U^A_n(\phi, \psi, s, \mathcal{E})\rangle_{n \in \mathbb{N}}$ is a non-decreasing sequence in $[0, 1]$ converging to $p^A_{s,\mathcal{E}}(\phi\, \mathcal{U}\, \psi)$.*

Next, for any adversary B of a probabilistic system PS, we define a sequence of functions $\langle R^B_n \rangle_{n \in \mathbb{N}}$, where $R^B_n(F, s)$ equals the probability, of reaching, from s under the adversary B, a state in F in at most n steps.

Definition 33 *Let $\mathsf{PS} = (S, Steps)$ be a probabilistic system. For any subset of states F, adversary $B \in Adv_{\mathsf{PS}}$ and $\pi \in Path^B_{fin}$, if $last(\pi) = s$ and $B(\pi) = \rho$, let:*

$$R^B_0(F, \pi) = \begin{cases} 1 \text{ if } s \in F \\ 0 \text{ otherwise} \end{cases}$$

and for any $n \geqslant 0$:

$$R^B_{n+1}(F, \pi) = \begin{cases} 1 & \text{if } s \in F \\ \sum_{s' \in S} \rho(s') \cdot R^B_n(F, \pi \xrightarrow{\rho} s') & \text{otherwise.} \end{cases}$$

Lemma 34 *For any probabilistic system $\mathsf{PS} = (S, Steps)$, adversary $B \in Adv_{\mathsf{PS}}$, state $s \in S$ and subset of states $F \subseteq S$: $\langle R^B_n(F, s)\rangle_{n \in \mathbb{N}}$ is a non-decreasing sequence in $[0, 1]$ converging to $p^A_s(\diamond\, F)$.*

We are now in a position to prove Proposition 29.

Proof of Proposition 29. Let $\mathsf{PS} = (\mathsf{Z}, Steps)$ be the probabilistic system generated by $\mathsf{MaxU}([\![\phi]\!], [\![\psi]\!], \gtrsim \lambda)$ and $\{E_{(l,g,p)} \,|(l, g, p) \in prob\}$ the set of edges used in this construction. We split the proof into proving a sequence of properties: (a), (b), (c) and (d).

(a) If $(\mathsf{z},(X,l'),\mathsf{z}') \in E_{(l,g,p)}$ and $(l,v),\mathcal{E} \in \mathsf{z}$, then the following conditions hold:
- $(l,v),\mathcal{E} \models \phi{\vee}\psi$;
- $v \triangleright \mathit{inv}(l){\wedge}g$;
- $(l',v[X{:=}0]),\mathcal{E} \in \mathsf{tpre}_{[\![\phi\vee\psi]\!]}(\mathsf{z}')$.

The result follows from the definition of dpre and tpre (see Section 4.1).

(b) For any $s,\mathcal{E} \in S{\times}\mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$, $p^{\max}_{s,\mathcal{E}}(\phi\,\mathcal{U}\,\psi) > 0$ if and only if $s,\mathcal{E} \in \mathsf{tpre}_{[\![\phi\vee\psi]\!]}(\mathsf{z})$ for some $\mathsf{z} \in \mathsf{Z}$.

The proof follows by induction on the shortest path to reach a state satisfying ψ passing through only ϕ states.

The main step in the proof involves showing, for all $n \in \mathbb{N}$, the following correspondence between the values of U^A_n for $A \in \mathit{Adv}_{\mathsf{TPS}}$ and R^B_n for $B \in \mathit{Adv}_{\mathsf{PS}}$.

(c) For any $B \in \mathit{Adv}_{\mathsf{PS}}$, $\mathsf{z} \in \mathsf{Z}$ and $(l,v),\mathcal{E} \in \mathsf{tpre}_{[\![\phi\vee\psi]\!]}(\mathsf{z})$, there exists $A \in \mathit{Adv}_{\mathsf{TPS}}$ such that

$$U^A_{2n}(\phi,\psi,(l,v),\mathcal{E}) \geqslant R^B_n(\mathsf{tpre}_{[\![\phi\vee\psi]\!]}[\![\psi]\!],\mathsf{z})\,.$$

(d) For any $A \in \mathit{Adv}_{\mathsf{TPS}}$ and $(l,v),\mathcal{E} \in S{\times}\mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$, if $p^{\max}_{(l,v),\mathcal{E}}(\phi\,\mathcal{U}\,\psi) > 0$, then there exists $\mathsf{z} \in \mathsf{Z}$ with $(l,v),\mathcal{E} \in \mathsf{tpre}_{[\![\phi\vee\psi]\!]}(\mathsf{z})$ and $B \in \mathit{Adv}_{\mathsf{PS}}$ such that

$$R^B_n(\mathsf{tpre}_{[\![\phi\vee\psi]\!]}[\![\psi]\!],\mathsf{z}) \geqslant U^A_n(\phi,\psi,(l,v),\mathcal{E})\,.$$

It follows from (b), Lemma 32 and Lemma 34 that to prove Proposition 29 it is sufficient to show that (c) and (d) hold. We now prove (c) and (d) by induction on $n \in \mathbb{N}$.

Proof of (c). Consider any $B \in \mathit{Adv}_{\mathsf{PS}}$, $\mathsf{z} \in \mathsf{Z}$ and $(l,v),\mathcal{E} \in \mathsf{tpre}_{[\![\phi\vee\psi]\!]}(\mathsf{z})$. If $n = 0$, then from Definition 33 we have the following two cases to consider.

- If $R^B_0(\mathsf{tpre}_{[\![\phi\vee\psi]\!]}[\![\psi]\!],\mathsf{z}) = 1$, then $\mathsf{z} \in \mathsf{tpre}_{[\![\phi\vee\psi]\!]}[\![\psi]\!]$ and by definition of tpre there exists $t \in \mathbb{R}_{\geqslant 0}$ such that $(l,v{+}t),\mathcal{E}{+}t \models \psi$ and $(l,v{+}t'),\mathcal{E}{+}t' \models \phi{\vee}\psi$ for all $t' \leqslant t$. Therefore letting A be the adversary such that $A(l,v) = (t,\mu_{(l,v+t)})$, it follows that:

$$U^A_{2\cdot 0}(\phi,\psi,(l,v),\mathcal{E}) \;=\; 1 \;=\; R^B_0(\mathsf{tpre}_{[\![\phi\vee\psi]\!]}[\![\psi]\!],\mathsf{z})\,.$$

- If $R^B_0(\mathsf{tpre}_{[\![\phi\vee\psi]\!]}[\![\psi]\!],\mathsf{z}) = 0$, then choosing any $A \in \mathit{Adv}_{\mathsf{TPS}}$ we have:

$$U^A_{2\cdot 0}(\phi,\psi,(l,v),\mathcal{E}) \;\geqslant\; 0 \;=\; R^B_0(\mathsf{tpre}_{[\![\phi\vee\psi]\!]}[\![\psi]\!],\mathsf{z})\,.$$

Since these are the only cases to consider (c) holds when $n = 0$.

Next, suppose that (c) holds for some $n \in \mathbb{N}$ and consider $U^A_{2(n+1)}(\phi, \psi, (l, v), \mathcal{E})$. If $\mathsf{z} \in \mathsf{tpre}_{[\![\phi\vee\psi]\!]}[\![\psi]\!]$ the result follows as in the case for $n = 0$. We are therefore left to consider the case when $\mathsf{z} \notin \mathsf{tpre}_{[\![\phi\vee\psi]\!]}[\![\psi]\!]$.

By construction, $B(\mathsf{z}) = \rho$ for some $(\mathsf{z}, \rho) \in \mathit{Steps}$, and from the construction of PS, there exists $(l, g, p) \in \mathit{prob}$ and set of edges $E_\rho \subseteq E_{(l,g,p)}$ such that $\mathsf{z} = (l, \zeta)$ for some $\zeta \in \mathit{Zones}(\mathcal{X} \cup \mathcal{Z})$ and for any $\mathsf{z}' \in \mathsf{Z}$:

$$\rho(\mathsf{z}') = \sum_{(\mathsf{z},(X,l'),\mathsf{z}')\in E_\rho} p(X, l') \,. \tag{20}$$

From Definition 33 we have:

$$\begin{aligned}
R^B_{n+1}(\mathsf{tpre}_{[\![\phi\vee\psi]\!]}[\![\psi]\!], \mathsf{z}) \;&=\; \sum_{\mathsf{z}'\in\mathsf{Z}} \rho(\mathsf{z}') \cdot R^B_n(\mathsf{tpre}_{[\![\phi\vee\psi]\!]}[\![\psi]\!], \mathsf{z} \xrightarrow{\rho} \mathsf{z}') \\
= \; \sum_{\mathsf{z}'\in\mathsf{Z}} \rho(\mathsf{z}') \cdot R^{B[\mathsf{z}\xrightarrow{\rho}\mathsf{z}']}_n(\mathsf{tpre}_{[\![\phi\vee\psi]\!]}[\![\psi]\!], \mathsf{z}') & \qquad \text{by Definition 30} \\
= \; \sum_{(\mathsf{z},(X,l'),\mathsf{z}')\in E_\rho} p(X, l') \cdot R^{B[\mathsf{z}\xrightarrow{\rho}\mathsf{z}']}_n(\mathsf{tpre}_{[\![\phi\vee\psi]\!]}[\![\psi]\!], \mathsf{z}') & \qquad \text{by (20).}
\end{aligned} \tag{21}$$

Since $(l, v), \mathcal{E} \in \mathsf{tpre}_{[\![\phi\vee\psi]\!]}(\mathsf{z})$, it follows that there exists $t \in \mathbb{R}_{\geqslant 0}$ such that $(l, v{+}t), \mathcal{E}{+}t \in \mathsf{z}$ and $((l, v), (t, \mu_{(l,v+t)})) \in \mathit{TSteps}$. Now, for any $(\mathsf{z}, (X, l'), \mathsf{z}') \in E_\rho$ using (a) we have that $(l', (v{+}t)[X{:=}0]), \mathcal{E}{+}t \in \mathsf{tpre}_{[\![\phi\vee\psi]\!]}(\mathsf{z}')$. Therefore, by induction, for any $e = (\mathsf{z}, (X, l'), \mathsf{z}') \in E_\rho$ there exists $A^{X,l'} \in \mathit{Adv}_{\mathsf{TPS}}$ such that:

$$U^{A^{X,l'}}_{2n}(\phi, \psi, (l', (v{+}t)[X{:=}0]), \mathcal{E} + t) \geqslant R^{B[\mathsf{z}\xrightarrow{\rho}\mathsf{z}']}_n(\mathsf{tpre}_{[\![\phi\vee\psi]\!]}[\![\psi]\!], \mathsf{z}') \,. \tag{22}$$

Let $A \in \mathit{Adv}_{\mathsf{TPS}}$ be the adversary such that

- $A(l, v) = (t, \mu_{(l,v+t)})$;
- $A\Big((l, v) \xrightarrow{t,\mu_{(l,v+t)}} (l, v{+}t)\Big) = (0, \mu)$ where for any $(l', v') \in S$:

$$\mu(l', v') = \sum_{\substack{X\subseteq\mathcal{X}\,\&\\ v'=(v+t)[X:=0]}} p(X, l'); \tag{23}$$

- for any $e = (\mathsf{z}, (X, l'), \mathsf{z}') \in E_\rho$:

$$A[(l, v) \xrightarrow{t,\mu_{(l,v+t)}} (l, v{+}t) \xrightarrow{0,\mu} (l', (v{+}t)[X{:=}0])] = A^{X,l'} \,.$$

Note that, the existence of the above distributions follows from Definition 17.

It then follows from Definition 31 and the construction of A that:

$$
\begin{aligned}
& U^{A}_{2(n+1)}(\phi, \psi, (l,v), \mathcal{E}) \\
= & \sum_{(l',v') \in \mathsf{support}(\mu)} p(X,l') \cdot U^{A^{X,l'}}_{2n}(\phi, \psi, (l',v'), \mathcal{E}{+}t) \\
= & \sum_{(X,l') \in \mathsf{support}(p)} p(X,l') \cdot U^{A^{X,l'}}_{2n}(\phi, \psi, (l', (v{+}t)[X{:=}0]), \mathcal{E}{+}t) && \text{by (23)} \\
\geqslant & \sum_{(\mathsf{z},(X,l'),\mathsf{z}') \in E_\rho} p(X,l') \cdot U^{A^{X,l'}}_{2n}(\phi, \psi, (l', (v{+}t)[X{:=}0]), \mathcal{E}{+}t) \\
& && \text{by construction of } E_\rho \\
\geqslant & \sum_{(\mathsf{z},(X,l'),\mathsf{z}') \in E_\rho} p(X,l') \cdot R^{B[\mathsf{z} \xrightarrow{\rho} \mathsf{z}']}_{n}(\mathsf{tpre}_{[\![\phi \vee \psi]\!]}[\![\psi]\!], \mathsf{z}') && \text{by (22)} \\
= \; & R^{B}_{n+1}(\mathsf{tpre}_{[\![\phi \vee \psi]\!]}[\![\psi]\!], \mathsf{z}) && \text{by (21)}
\end{aligned}
$$

and since z and B are arbitrary, (c) holds by induction.

Proof of (d). Consider any $A \in Adv_{\mathsf{TPS}}$ and $(l,v), \mathcal{E} \in S{\times}\mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$ such that $p^{\max}_{(l,v),\mathcal{E}}(\phi \, \mathcal{U} \, \psi) > 0$. When $n = 0$, by Definition 31 we have the following two possibilities.

- $U^{A}_{0}(\phi, \psi, (l,v), \mathcal{E}) = 1$: in this case there exists $t \in \mathbb{R}_{\geqslant 0}$ such that $(l, v{+}t), \mathcal{E}{+}t \models \psi$ and $(l, v{+}t'), \mathcal{E}{+}t' \models \phi{\vee}\psi$ for all $t' \leqslant t$. By definition of tpre it follows that $(l,v), \mathcal{E} \in \mathsf{tpre}_{[\![\phi \vee \psi]\!]}[\![\psi]\!]$, and, by construction of Z, there exists $\mathsf{z} \in Z$ such that $\mathsf{z} \in \mathsf{tpre}_{[\![\phi \vee \psi]\!]}[\![\psi]\!]$ and $(l,v), \mathcal{E} \in \mathsf{tpre}_{[\![\phi \vee \psi]\!]}(\mathsf{z})$. Combining these facts we have:
$$R^{B}_{0}(\mathsf{tpre}_{[\![\phi \vee \psi]\!]}[\![\psi]\!], \mathsf{z}) \;=\; 1 \;=\; U^{A}_{0}(\phi, \psi, (l,v), \mathcal{E})$$
for all $B \in Adv_{\mathsf{PS}}$.
- $U^{A}_{0}(\phi, \psi, (l,v), \mathcal{E}) = 0$: choosing any $B \in Adv_{\mathsf{PS}}$ and $\mathsf{z} \in Z$ such that $(l,v), \mathcal{E} \in \mathsf{tpre}_{[\![\phi \vee \psi]\!]}(\mathsf{z})$ (the existence of z follows from (b)) we have:
$$R^{B}_{0}(\mathsf{tpre}_{[\![\phi \vee \psi]\!]}[\![\psi]\!], \mathsf{z}) \;\geqslant\; 0 = \; U^{A}_{0}(\phi, \psi, (l,v), \mathcal{E}) \; .$$

Since these are the only cases to consider, (d) holds when $n = 0$.

Now suppose that (d) holds from some $n \in \mathbb{N}$ and consider $U^{A}_{n+1}(\phi, \psi, (l,v), \mathcal{E})$. If $U^{A}_{n+1}(\phi, \psi, (l,v), \mathcal{E}) = 0$, then choosing any $B \in Adv_{\mathsf{PS}}$ and $\mathsf{z} \in Z$ such that $(l,v), \mathcal{E} \in \mathsf{tpre}_{[\![\phi \vee \psi]\!]}(\mathsf{z})$ (the existence of z follows from (b)) we have:

$$R^{B}_{n+1}(\mathsf{tpre}_{[\![\phi \vee \psi]\!]}[\![\psi]\!], \mathsf{z}) \;\geqslant\; 0 = \; U^{A}_{n+1}(\phi, \psi, (l,v), \mathcal{E})$$

as required. It therefore remains to consider the case when $U^{A}_{n+1}(\phi, \psi, (l,v), \mathcal{E}) > 0$. From Definition 17 and Definition 31 we have the following possibilities.

- $A(l,v) = (t, \mu_{(l,v+t)})$ and there exists $t' \leqslant t$ such that $(l, v{+}t'), \mathcal{E}{+}t' \models \psi$ and $(l, v{+}t''), \mathcal{E}{+}t'' \models \phi{\vee}\psi$ for all $t'' \leqslant t'$. By definition of tpre it follows that $(l,v),\mathcal{E} \in \mathsf{tpre}_{[\![\phi\vee\psi]\!]}[\![\psi]\!]$, and hence

$$R^B_{n+1}(\mathsf{tpre}_{[\![\phi\vee\psi]\!]}[\![\psi]\!],\mathsf{z}) \;=\; 1 \;=\; U^A_{n+1}(\phi,\psi,(l,v),\mathcal{E})$$

for all $B \in Adv_{\mathsf{PS}}$.

- $A(l,v) = (t,\mu_{(l,v+t)})$ such that $(l,v+t'),\mathcal{E}+t' \models \phi\wedge\neg\psi$ for all $t' \leqslant t$. In this case we have

$$U^A_{n+1}(\phi,\psi,(l,v),\mathcal{E}) = U^A_n(\phi,\psi,(l,v) \xrightarrow{t,\mu_{(l,v+t)}} (l,v+t),\mathcal{E}+t)\,.$$

  and the result follows by induction and Lemma 34.
- $A(l,v) = (0,\mu)$. Then by Definition 31 we have:

$$U^A_{n+1}(\phi,\psi,(l,v),\mathcal{E}) = \sum_{(l',v')\in S} \mu(l',v')\cdot U^A_n(\phi,\psi,(l,v) \xrightarrow{0,\mu} (l',v'),\mathcal{E})$$

  and $(l,v),\mathcal{E} \models \phi\wedge\neg\psi$. Now, from Definition 17, there exists $(l,g,p) \in prob$ such that $v \triangleright g$ and for any $(l',v') \in S$:

$$\mu(l',v') = \sum_{\substack{X\subseteq\mathcal{X}\,\&\\ v'=v[X:=0]}} p(X,l')\,.$$

  Therefore, by Definition 31 and Definition 30:

$$\begin{aligned} & U^A_{n+1}(\phi,\psi,(l,v),\mathcal{E}) \\ & = \sum_{(X,l')\in\mathsf{support}(p)} p(X,l')\cdot U^{A[X,l']}_n(\phi,\psi,(l',v[X:=0]),\mathcal{E}) \end{aligned} \tag{24}$$

  where, to ease notation, we use $A[X,l']$ to denote the adversary $A[(l,v) \xrightarrow{0,\mu} (l',v[X:=0])]$.

  Now consider any $(X,l') \in \mathsf{support}(p)$ such that $U^{A[X,l']}_n(\phi,\psi,l',v[X:=0]),\mathcal{E}) > 0$. By definition $(l,g,p,X,l') \in \mathsf{edges}$. By induction and Lemma 32 there exists $(l',\zeta'_{X,l'}) \in \mathsf{Z}$ and adversary $B^{(X,l')}$ such that $(l',v[X:=0]),\mathcal{E} \in \mathsf{tpre}_{[\![\phi\vee\psi]\!]}(l',\zeta'_{X,l'})$ and

$$R^{B^{(X,l')}}_n(\mathsf{tpre}_{[\![\phi\vee\psi]\!]}[\![\psi]\!],(l',\zeta'_{X,l'})) \geqslant U^{A[X,l']}_n(\phi,\psi,(l',v[X:=0]),\mathcal{E})\,. \tag{25}$$

  Since $(l,v),\mathcal{E} \models \phi\wedge\neg\psi$, letting:

$$(l,\zeta_{X,l'}) = \mathsf{dpre}((l,g,p,X,l'),\mathsf{tpre}_{[\![\phi\vee\psi]\!]}(l',\zeta'_{X,l'})),$$

  it follows that $((l,\zeta_{X,l'}),(X,l'),(l',\zeta'_{X,l'})) \in E_{(l,g,p)}$, $(l,\zeta_{X,l'}) \in \mathsf{Z}$ and $(l,v),\mathcal{E} \in (l,\zeta_{X,l'})$. Therefore, from the construction of PS, by setting z equal to:

$$\Big(l,\bigwedge\{\zeta_{X,l'} \mid (X,l') \in \mathsf{support}(p) \text{ and } p^{\max}_{(l',v[X:=0]),\mathcal{E}}(\phi\,\mathcal{U}\,\psi) > 0\}\Big)$$

we have $\mathsf{z} \in \mathsf{Z}$ and $(l, v), \mathcal{E} \in \mathsf{z}$. Furthermore, by construction of PS there exists $(\mathsf{z}, \rho) \in \mathit{Steps}$ such that for any $(l', \zeta') \in \mathsf{Z}$:

$$\rho(l', \zeta') \geqslant \sum_{\substack{(X,l') \in \mathsf{support}(p),\ \zeta' = \zeta_{X,l'}\ \& \\ U_n^{A[X,l']}(\phi,\psi,(l',v[X:=0]),\mathcal{E})>0}} p(X, l'). \tag{26}$$

Now, setting B to be the adversary of PS such that $B(\mathsf{z}) = \rho$ and $B[\mathsf{z} \xrightarrow{\rho} (l', \zeta'_{X,l'})] = B^{(X,l')}$, by Definition 33 we have:

$$\begin{aligned}
& R^B_{n+1}(\mathsf{tpre}_{[\![\phi \vee \psi]\!]}[\![\psi]\!], \mathsf{z}) = \sum_{\mathsf{z}' \in \mathsf{Z}} \rho(\mathsf{z}') \cdot R^B_n(\mathsf{tpre}_{[\![\phi \vee \psi]\!]}[\![\psi]\!], \mathsf{z} \xrightarrow{\rho} \mathsf{z}') \\
\geqslant & \sum_{\substack{(X,l') \in \mathsf{support}(p)\ \& \\ U_n^{A[X,l']}(l',v[X:=0]),\mathcal{E})>0}} p(X, l') \cdot R^B_n(\mathsf{tpre}_{[\![\phi \vee \psi]\!]}[\![\psi]\!], \mathsf{z} \xrightarrow{\rho} (l', \zeta'_{X,l'})) && \text{by (26)} \\
= & \sum_{\substack{(X,l') \in \mathsf{support}(p)\ \& \\ U_n^{A[X,l']}(l',v[X:=0]),\mathcal{E})>0}} p(X, l') \cdot R^{B^{(X,l')}}_n(\mathsf{tpre}_{[\![\phi \vee \psi]\!]}[\![\psi]\!], (l', \zeta'_{X,l'})) && \text{by construction} \\
\geqslant & \sum_{\substack{(X,l') \in \mathsf{support}(p)\ \& \\ U_n^{A[X,l']}(l',v[X:=0]),\mathcal{E})>0}} p(X, l') \cdot U_n^{A[X,l']}(\phi, \psi, (l', v[X:=0]), \mathcal{E}) && \text{by (25)} \\
= & \sum_{(X,l') \in \mathsf{support}(p)} p(X, l') \cdot U_n^{A[X,l']}(\phi, \psi, (l, v[X:=0]), \mathcal{E}) && \text{rearranging} \\
= & \ U^A_{n+1}(\phi, \psi, (l, v), \mathcal{E}) && \text{by (24).}
\end{aligned}$$

Since these are all the cases to consider, (d) holds by induction as required. □

Using this result, for $\lambda \in (0, 1)$, we set:

$$\begin{aligned}
\mathsf{Until}([\![\phi]\!], [\![\psi]\!], \lesssim \lambda) &\overset{\text{def}}{=} [\![\mathtt{true}]\!] \setminus \mathsf{MaxU}([\![\phi]\!], [\![\psi]\!], \not\lesssim \lambda) \\
\mathsf{Release}([\![\phi]\!], [\![\psi]\!], \gtrsim \lambda) &\overset{\text{def}}{=} [\![\mathtt{true}]\!] \setminus \mathsf{MaxU}([\![\neg\phi]\!], [\![\neg\psi]\!], \gtrsim 1 - \lambda)\,.
\end{aligned}$$

4.2.3 Example

We now return to the PTA in Example 16 and verify the property $z.\mathcal{P}_{<\lambda}[\phi\,\mathcal{U}\,\psi]$, where $\phi = \mathtt{true}$ and $\psi = \mathsf{sr}\wedge(z{<}6)$, which involves computing the maximal probability of a message being correctly delivered before 6 time units have elapsed. In particular, we consider this maximum probability when starting from the location di with the clock x equal to 0. In this example, we do not distinguish between the name of a location and the atomic proposition with which it is labelled.

According to our methodology, the set of states satisfying $\mathcal{P}_{<\lambda}[\phi\,\mathcal{U}\,\psi]$ is given

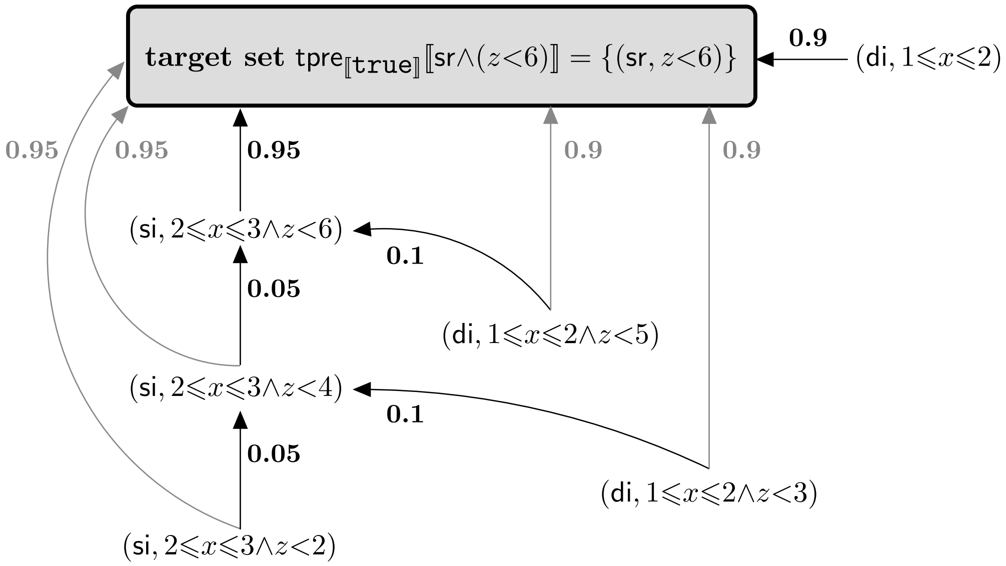
target set $\mathsf{tpre}_{[\![\mathtt{true}]\!]}[\![\mathsf{sr}\wedge(z<6)]\!] = \{(\mathsf{sr}, z<6)\}$
0.9
$(\mathsf{di}, 1\leqslant x\leqslant 2)$
0.95
0.95
0.95
0.9
0.9
$(\mathsf{si}, 2\leqslant x\leqslant 3\wedge z<6)$
0.1
0.05
$(\mathsf{di}, 1\leqslant x\leqslant 2\wedge z<5)$
$(\mathsf{si}, 2\leqslant x\leqslant 3\wedge z<4)$
0.1
0.05
$(\mathsf{di}, 1\leqslant x\leqslant 2\wedge z<3)$
$(\mathsf{si}, 2\leqslant x\leqslant 3\wedge z<2)$

Fig. 6. Probabilistic system generated by $\mathsf{MaxU}([\![\mathtt{true}]\!], [\![\mathsf{sr}\wedge(z<6)]\!], \geqslant 1-\lambda)$

by: $[\![\mathtt{true}]\!]\backslash\mathsf{MaxU}(\,[\![\phi]\!], [\![\psi]\!], \geqslant \lambda)$. Applying $\mathsf{MaxU}([\![\mathtt{true}]\!], [\![\mathsf{sr}\wedge(z<6)]\!], \geqslant \lambda)$ the probabilistic system given in Figure 6 is generated where the darker arrows correspond to those edges generated by time and discrete predecessor operations (line 11 of Figure 5) and the lighter arrows are those generated in the construction of the probabilistic system (line 20 of Figure 5). Appendix A presents the computations performed by MaxU in the construction of the states and edges of this probabilistic system. From Proposition 29 we have that, starting from di with x equal to 0, the maximum probability of satisfying $\mathtt{true}\,\mathcal{U}\,(\mathsf{sr}\wedge(z<6))$ is 0.99525, corresponding to the maximum probability of $(\mathsf{di}, 1\leqslant x\leqslant 2 \wedge z<3)$ reaching the target set in the probabilistic system given in Figure 6.

4.3 Computing Maximum Release Probabilities

In this section we present methods for calculating the set of states satisfying a formula of the form $\mathcal{P}_{\lesssim\lambda}[\phi\,\mathcal{V}\,\psi]$ and $\mathcal{P}_{\gtrsim\lambda}[\phi\,\mathcal{U}\,\psi]$ which, from (6) and (7), reduce to the computation of $p^{\max}_{s,\mathcal{E}}(\phi\,\mathcal{V}\,\psi)$ or $p^{\max}_{s,\mathcal{E}}(\neg\phi\,\mathcal{V}\,\neg\psi)$ for all state and formula clock valuation pairs $s,\mathcal{E}$. We first consider computing the set of state and formula clock valuation pairs $\{s,\mathcal{E}\,|\,p^{\max}_{s,\mathcal{E}}(\phi\,\mathcal{V}\,\psi)\geqslant 1\}$ which we achieved by derive a probabilistic analogue of (4). More precisely, by replacing the $\exists$ operator with $\neg\mathcal{P}_{<1}[\cdot]$ (recall that $\neg\mathcal{P}_{<1}[\cdot]$ stands for ‘it is not the case that all adversaries satisfy the path formula with probability less than 1’, which in turn can be translated as ‘there exists an adversary satisfying the path formula with probability 1’). We then arrive at the following proposition.

Proposition 35 *For any positive integer $c \in \mathbb{N}$ and* PTCTL *formulae ϕ,ψ, if $z \in \mathcal{Z}$ does not appear in either ϕ or ψ, then the set* $\{s,\mathcal{E}\,|\,p^{\max}_{s,\mathcal{E}}(\phi\,\mathcal{V}\,\psi)\geqslant 1\}$

is given by the fixpoint

$$\mathit{gfp}\, Y.\, \Big(\psi \wedge z.\neg \mathcal{P}_{<1} \Big[Y\, \mathcal{U} \Big((Y \wedge \phi) \vee (z{>}c) \Big) \Big] \Big) \,.$$

Proof. Consider any positive integer $c \in \mathbb{N}$, PTCTL formulae ϕ, ψ, and formula clock $z \in \mathcal{Z}$ which does not appear in either ϕ or ψ. To ease notation we use $p^{\max}_{\geqslant 1}(\phi\, \mathcal{V}\, \psi)$ to denote the set of state and formula clock valuation pairs $\{s, \mathcal{E} \,|\, p^{\max}_{s,\mathcal{E}}(\phi\, \mathcal{V}\, \psi) {\geqslant} 1\}$, and for any $X \subseteq S {\times} \mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$ let:

$$G_1(X, c) \overset{\text{def}}{=} \psi \wedge z.\neg \mathcal{P}_{<1}[X\, \mathcal{U}\, ((X \wedge \phi) \vee (z{>}c))] \,.$$

The proposition is proved by showing:

(1) the set $p^{\max}_{\geqslant 1}(\phi\, \mathcal{V}\, \psi)$ is a fixpoint of $G_1(\cdot, c)$;
(2) if $G_1(Y, c) = Y$, then $Y \subseteq p^{\max}_{\geqslant 1}(\phi\, \mathcal{V}\, \psi)$.

First, since, for any $X \subseteq S {\times} \mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$, $X \supseteq [\![z.\neg \mathcal{P}_{<1}[X\, \mathcal{U}\, ((X \wedge \phi) \vee (z{>}c))]]\!]$ it follows that $X \supseteq G_1(X, c)$ for all $X \subseteq S {\times} \mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$. Therefore, to prove that $p^{\max}_{\geqslant 1}(\phi\, \mathcal{V}\, \psi)$ is a fixpoint it is sufficient to show that:

$$G_1\Big(p^{\max}_{\geqslant 1}(\phi\, \mathcal{V}\, \psi), c \Big) \supseteq p^{\max}_{\geqslant 1}(\phi\, \mathcal{V}\, \psi) \,.$$

By definition of $\phi\, \mathcal{V}\, \psi$ (see Section 3.3) the following properties hold.

- For any $s, \mathcal{E} \in S {\times} \mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$, if $s, \mathcal{E} \models \phi {\wedge} \psi$, then $\omega, \mathcal{E} \models \phi\, \mathcal{V}\, \psi$ for all paths $\omega \in \mathit{Path}_{\mathit{ful}}(s)$.

  Therefore, if $s, \mathcal{E} \models \phi {\wedge} \psi$, it follows that $s, \mathcal{E} \in p^{\max}_{\geqslant 1}(\phi\, \mathcal{V}\, \psi)$, and hence

  $$s, \mathcal{E} \models (\phi {\wedge} \psi) \vee (z{>}c) \quad \Rightarrow \quad s, \mathcal{E} \models (\, p^{\max}_{\geqslant 1}(\phi\, \mathcal{V}\, \psi) \wedge \phi \,) \vee (z{>}c) \,.$$

  Using this result and Definition 19 it follows that for any $s, \mathcal{E} \in S {\times} \mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$:

  $$\begin{aligned} & s, \mathcal{E} \models z.\neg \mathcal{P}_{<1} \Big[p^{\max}_{\geqslant 1}(\phi\, \mathcal{V}\, \psi)\, \mathcal{U} \Big((\phi {\wedge} \psi) \vee (z{>}c) \Big) \Big] \\ & \quad \Rightarrow s, \mathcal{E} \models z.\neg \mathcal{P}_{<1} \Big[p^{\max}_{\geqslant 1}(\phi\, \mathcal{V}\, \psi)\, \mathcal{U} \, \Big(\Big(p^{\max}_{\geqslant 1}(\phi\, \mathcal{V}\, \psi) \wedge \phi \Big) \vee (z{>}c) \Big) \Big] \,. \end{aligned} \tag{27}$$

- For any $s, \mathcal{E} \in S {\times} \mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$ and $\omega \in \mathit{Path}_{\mathit{ful}}(s)$, if $\omega, \mathcal{E} \models \phi\, \mathcal{V}\, \psi$, then $s, \mathcal{E} \models \psi$.

  Thus, for any $s, \mathcal{E} \in S {\times} \mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$ we have:

  $$s, \mathcal{E} \in p^{\max}_{\geqslant 1}(\phi\, \mathcal{V}\, \psi) \;\Rightarrow\; s, \mathcal{E} \models \psi \,. \tag{28}$$

- As the satisfaction of PTCTL is with respect to divergent adversaries, for any $s, \mathcal{E} \in p^{\max}_{\geqslant 1}(\phi\, \mathcal{V}\, \psi)$, there exists an adversary A such that, from $s, \mathcal{E}$ with probability 1, one remains in $p^{\max}_{\geqslant 1}(\phi\, \mathcal{V}\, \psi)$ until either a state satisfying $\phi {\wedge} \psi$ is reached or more than c time units pass.

Therefore, since the clock z does not appear in ϕ or ψ, for any $s,\mathcal{E} \in S\times\mathbb{R}_{\geqslant 0}^{\mathcal{Z}}$:

$$\begin{aligned}&s,\mathcal{E} \in p_{\geqslant 1}^{\max}(\phi \,\mathcal{V}\, \psi)\\ &\Rightarrow s,\mathcal{E}[z{:=}0] \models \neg\mathcal{P}_{<1}\left[p_{\geqslant 1}^{\max}(\phi \,\mathcal{V}\, \psi)\,\mathcal{U}\left((\phi\wedge\psi)\vee(z{>}c)\right)\right].\end{aligned} \tag{29}$$

Now, by definition of G_1:

$$\begin{aligned}&G_1(p_{\geqslant 1}^{\max}(\phi \,\mathcal{V}\, \psi), c)\\ &=\ \psi \wedge z.\neg\mathcal{P}_{<1}\left[p_{\geqslant 1}^{\max}(\phi \,\mathcal{V}\, \psi)\,\mathcal{U}\ \left(\left(p_{\geqslant 1}^{\max}(\phi \,\mathcal{V}\, \psi)\wedge\phi\right)\vee(z{>}c)\right)\right]\\ &\supseteq\ \psi \wedge z.\neg\mathcal{P}_{<1}\left[p_{\geqslant 1}^{\max}(\phi \,\mathcal{V}\, \psi)\,\mathcal{U}\left((\phi\wedge\psi)\vee(z{>}c)\right)\right] && \text{by (27)}\\ &\supseteq\ \psi \wedge p_{\geqslant 1}^{\max}(\phi \,\mathcal{V}\, \psi) && \text{by (29) and Definition 19}\\ &=\ p_{\geqslant 1}^{\max}(\phi \,\mathcal{V}\, \psi) && \text{by (28)}\end{aligned}$$

and hence $p_{\geqslant 1}^{\max}(\phi \,\mathcal{V}\, \psi)$ is a fixpoint of $G_1(X, c)$.

To complete the proof it remains to show that, if $G_1(Y, c) = Y$, then $Y \subseteq p_{\geqslant 1}^{\max}(\phi \,\mathcal{V}\, \psi)$ which we prove by contradiction. Therefore, suppose that there exists $Y \subseteq S\times\mathbb{R}_{\geqslant 0}^{\mathcal{Z}}$ such that $G_1(Y, c) = Y$ and $Y \setminus p_{\geqslant 1}^{\max}(\phi \,\mathcal{V}\, \psi) \neq \emptyset$. Now for any $s,\mathcal{E} \in Y \setminus p_{\geqslant 1}^{\max}(\phi \,\mathcal{V}\, \psi)$, and (divergent) adversary A, since $s,\mathcal{E} \notin p_{\geqslant 1}^{\max}(\phi \,\mathcal{V}\, \psi)$, under A starting from $s,\mathcal{E}$ the probability of satisfying $\phi \,\mathcal{V}\, \psi$ is less than 1, and therefore the probability of satisfying the dual formula $\neg\phi\,\mathcal{U}\,\neg\psi$ is greater than 0. More precisely, there exists a path $\omega \in Path_{ful}^{A}(s)$ such that $\omega,\mathcal{E} \models \neg\phi\,\mathcal{U}\,\neg\psi$, and since z does not appear in either ϕ or ψ, we have $\omega,\mathcal{E}[z{:=}0] \models \neg\phi\,\mathcal{U}\,\neg\psi$. Hence, there exists some duration $t_A \in \mathbb{R}_{\geqslant 0}$ such that at some point along this path $\neg\psi \wedge (z{=}t_A)$ is true and at all preceding points $\neg\phi\vee\neg\psi$ is true.

However, since $s,\mathcal{E} \in Y$, and therefore $s,\mathcal{E} \in G_1(Y, c)$, it follows that there exists an adversary such that with probability 1 from $s,\mathcal{E}[z{:=}0]$ one remains in Y while $z \leqslant c$ unless a state in Y which satisfies ϕ is reached. Since the above holds for any $s',\mathcal{E}' \in Y$ and z does not appear in ϕ or ψ, iterating the result n times, we can construct an adversary A' such that, from $s,\mathcal{E}$, with probability 1 one remains in Y while $z \leqslant n\cdot c$ unless a state in Y which satisfies ϕ is reached. Furthermore, since $Y = G_1(Y, c)$ it follows that $Y \subseteq \psi$, and hence under A', for any $n \in \mathbb{N}$, with probability 1, from $s,\mathcal{E}$ one remains in states satisfying ψ while $z \leqslant n\cdot c$ unless a state satisfying $\phi\wedge\psi$ is reached. From the reasoning of the preceding paragraph, there exists some duration $t_{A'}$ and path $\omega' \in Path_{ful}^{A'}(s)$ such that at some point along this path $\neg\psi \wedge (z{=}t_{A'})$ is true and at all preceding points $\neg\phi\vee\neg\psi$ is true. However, considering any $n \in \mathbb{N}$ such that $n\cdot c > t_{A'}$ (which exists since $c > 0$) leads to a contradiction. □

Using Proposition 35, the algorithm for calculating the set of state and formula clock valuation pairs $\{s,\mathcal{E} \mid p_{s,\mathcal{E}}^{\max}(\phi \,\mathcal{V}\, \psi){\geqslant}1\}$ is presented in Figure 7. As in

```
algorithm $\mathsf{MaxV}_{\geqslant 1}(c, \mathtt{U}, \mathtt{V})$

$\mathtt{Z}:=[\![\mathtt{true}]\!]$
repeat
  $\mathtt{Y}:=\mathtt{Z}$
  $\mathtt{Z}:=\mathtt{V} \wedge z.\mathsf{MaxU}_{\geqslant 1}(\mathtt{Y}, (\mathtt{U} \wedge \mathtt{Y}) \vee [\![z>c]\!])$
until $\mathtt{Z} = \mathtt{Y}$
return $\mathtt{Z}$
```

```
algorithm NonZeno

$\mathtt{Z}:=[\![\mathtt{true}]\!]$
repeat
  $\mathtt{Y}:=\mathtt{Z}$
  $\mathtt{Z}:=z.\mathsf{MaxU}_{\geqslant 1}([\![\mathtt{true}]\!], \mathtt{Y} \wedge [\![z=1]\!])$
until $\mathtt{Z} = \mathtt{Y}$
return $\mathsf{Z}_0$
```

Fig. 7. $\mathsf{MaxV}_{\geqslant 1}(c, \mathtt{U}, \mathtt{V})$ and $\mathsf{NonZeno}$ algorithms

the non-probabilistic case (when applying (3)), the choice of the value of c may affect the number of iterations performed in the computation. Intuitively, as c is increased, the number of iterations required by the 'inner' loop (the computation performed in one call to $\mathsf{MaxU}_{\geqslant 1}$) may increase, while the number of iterations performed by the 'outer' loop (calls to the algorithm $\mathsf{MaxU}_{\geqslant 1}$) may decrease.

Unfortunately we cannot use the same approach for calculating the set of state and formula clock valuation pairs $\{s, \mathcal{E} \,|\, p^{\max}_{s,\mathcal{E}}(\phi \,\mathcal{V}\, \psi){>}0\}$, i.e. in (4) replace $\exists$ with $\neg\mathcal{P}_{\leqslant 0}[\cdot]$. This is because the greatest fixpoint in this case yields the set of state and formula clock valuation pairs for which, under some divergent adversary, there exists a path which satisfies $\phi \,\mathcal{V}\, \psi$, which does not imply that the probability of satisfying $\phi \,\mathcal{V}\, \psi$ is greater than zero.

Instead, we employ the following proposition, which together with Proposition 35 provides us with a method for calculating $\{s, \mathcal{E} \,|\, p^{\max}_{s,\mathcal{E}}(\phi \,\mathcal{V}\, \psi){>}0\}$ and computing quantitative maximum release probabilities.

Proposition 36 *For any probabilistic timed automaton* PTA*, corresponding timed probabilistic system* $\mathsf{TPS} = (S, \mathit{TSteps}, \mathcal{L}')$*, state and formula clock valuation pair* $s, \mathcal{E} \in S \times \mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$ *and* PTCTL *formulae* ϕ, ψ:

$$p^{\max}_{s,\mathcal{E}}(\phi \,\mathcal{V}\, \psi) = p^{\max}_{s,\mathcal{E}}\Big(\psi \,\mathcal{U}\, \neg\mathcal{P}_{<1}[\phi \,\mathcal{V}\, \psi]\Big)\,.$$

Proof. Consider any probabilistic timed automaton PTA, corresponding timed probabilistic system $\mathsf{TPS} = (S, \mathit{TSteps}, \mathcal{L}')$ and PTCTL formulae ϕ and ψ. We begin by showing that for any state and formula clock valuation pair $s, \mathcal{E} \in S \times \mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$:

$$p^{\max}_{s,\mathcal{E}}(\phi \,\mathcal{V}\, \psi) = p^{\max}_{s,\mathcal{E}}\Big(\psi \,\mathcal{U}\, \big((\phi \wedge \psi) \vee \neg\mathcal{P}_{<1}[\Box(\neg\phi \wedge \psi)]\big)\Big)\,. \qquad (30)$$

First, given an adversary $A \in \mathit{Adv}_{\mathsf{TPS}}$, let A' be the adversary which behaves as A unless a state and formula clock valuation pair $s', \mathcal{E}'$ satisfying

$\neg\mathcal{P}_{<1}[\Box(\neg\phi\wedge\psi)]$ is reached, in which case A' behaves like the adversary $A^{\max}$ for which:

$$p^{A^{\max}}_{s',\mathcal{E}'}\Big(\Box\,(\neg\phi\wedge\psi)\Big) = 1\,.$$

The existence of $A^{\max}$ follows from the fact that $s',\mathcal{E}' \models \neg\mathcal{P}_{<1}[\Box(\neg\phi\wedge\psi)]$ and Lemma 22. Using Lemma 8, for any $s,\mathcal{E} \in S\times\mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$ and $A'' \in \mathit{Adv}_{\mathsf{TPS}}$, we have that:

$$p^{A''}_{s,\mathcal{E}}(\Box\,(\neg\phi\wedge\psi)) = 1 \;\;\Leftrightarrow\;\; \forall\omega \in \mathit{Path}^{A''}_{\mathit{ful}}(s).\; \omega,\mathcal{E} \models \Box\,(\neg\phi\wedge\psi)\;,$$

and hence, by construction of A', for any $s,\mathcal{E} \in S\times\mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$ and path $\omega \in \mathit{Path}^{A'}_{\mathit{ful}}(s)$, if $\omega,\mathcal{E} \models \psi\,\mathcal{U}\,\Big((\phi\wedge\psi) \vee \neg\mathcal{P}_{<1}[\Box(\neg\phi\wedge\psi)]\Big)$, then $\omega,\mathcal{E} \models \phi\,\mathcal{V}\,\psi$. Therefore, for any $s,\mathcal{E} \in S\times\mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$:

$$\begin{aligned} p^{A'}_{s,\mathcal{E}}(\phi\,\mathcal{V}\,\psi) &\geqslant\; p^{A'}_{s,\mathcal{E}}\Big(\,\psi\,\mathcal{U}\,\Big((\phi\wedge\psi) \vee \neg\mathcal{P}_{<1}[\Box(\neg\phi\wedge\psi)]\Big)\Big) \\ &=\; p^{A}_{s,\mathcal{E}}\Big(\,\psi\,\mathcal{U}\,\Big((\phi\wedge\psi) \vee \neg\mathcal{P}_{<1}[\Box(\neg\phi\wedge\psi)]\Big)\Big) \quad \text{by construction of } A'. \end{aligned}$$

Since this construction was for an arbitrary adversary $A \in \mathit{Adv}_{\mathsf{TPS}}$ and state and formula clock valuation pair $s,\mathcal{E} \in S\times\mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$ it follows that:

$$p^{\max}_{s,\mathcal{E}}(\phi\,\mathcal{V}\,\psi) \geqslant p^{\max}_{s,\mathcal{E}}\Big(\psi\;\mathcal{U}\,\Big((\phi\wedge\psi) \vee \neg\mathcal{P}_{<1}[\Box(\neg\phi\wedge\psi)]\Big)\Big)\;\; \forall s,\mathcal{E} \in S\times\mathbb{R}^{\mathcal{Z}}_{\geqslant 0}. \quad (31)$$

We now show that the reverse inequality holds. Let R be the region graph of PTA and the PTCTL formula

$$\theta = \mathcal{P}_{\sim\lambda}[\phi\,\mathcal{V}\,\psi] \wedge \mathcal{P}_{\sim\lambda}[\Psi\,\mathcal{U}\,((\Phi\wedge\Psi) \vee \neg\mathcal{P}_{<1}[\Box\,(\neg\Phi\wedge\Psi)])]$$

(see Proposition 21). Now, for any adversary $A \in \mathit{Adv}_{\mathsf{TPS}}$ and state and formula clock valuation pair $s,\mathcal{E} \in S\times\mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$ there exists an adversary $B \in \mathit{Adv}_R$ of the region graph R and PCTL formulae Φ and Ψ, such that:

$$\begin{aligned} p^{A}_{s,\mathcal{E}}(\phi\,\mathcal{V}\,\psi) &=\; p^{B}_{r}(\Phi\,\mathcal{V}\,\Psi) \\ &\leqslant\; p^{B}_{r}\Big(\,\Psi\,\mathcal{U}\,\Big((\Phi\wedge\Psi) \vee \neg\mathcal{P}_{<1}[\Box\,(\neg\Phi\wedge\Psi)]\Big)\Big) \qquad \text{by Lemma 10} \\ &=\; p^{A}_{s,\mathcal{E}}\Big(\psi\;\mathcal{U}\,\Big((\phi\wedge\psi) \vee \neg\mathcal{P}_{<1}[\Box(\neg\phi\wedge\psi)]\Big)\Big) \quad \text{by Proposition 21.} \end{aligned}$$

Since this was for an arbitrary adversary $A \in \mathit{Adv}_{\mathsf{TPS}}$ and state and formula clock valuation pair $s,\mathcal{E} \in S\times\mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$ we have:

$$p^{\max}_{s,\mathcal{E}}(\phi\,\mathcal{V}\,\psi) \leqslant p^{\max}_{s,\mathcal{E}}\Big(\psi\;\mathcal{U}\,\Big((\phi\wedge\psi) \vee \neg\mathcal{P}_{<1}[\Box(\neg\phi\wedge\psi)]\Big)\Big)\;\; \forall s,\mathcal{E} \in S\times\mathbb{R}^{\mathcal{Z}}_{\geqslant 0},$$

which together with (31) proves the correctness of (30).

Next, from (30) it follows that for any state and formula clock valuation pair $s,\mathcal{E} \in S\times\mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$:

$$s,\mathcal{E} \models \neg\mathcal{P}_{<1}\Big[\,\psi\,\mathcal{U}\,\Big((\phi\wedge\psi) \vee \neg\mathcal{P}_{<1}[\Box(\neg\phi\wedge\psi)]\Big)\Big] \Leftrightarrow s,\mathcal{E} \models \neg\mathcal{P}_{<1}[\phi\,\mathcal{V}\,\psi]\,. \quad (32)$$

Before we give the final step of the proof we require the following property: for any PTCTL formulae θ_1, θ_2 and $s, \mathcal{E} \in S \times \mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$:

$$p^{\max}_{s,\mathcal{E}}(\theta_1 \, \mathcal{U} \, \theta_2) = p^{\max}_{s,\mathcal{E}}(\theta_1 \, \mathcal{U} \, \neg \mathcal{P}_{<1}[\theta_1 \, \mathcal{U} \, \theta_2]) \, . \tag{33}$$

which follows from Proposition 21 and Lemma 5.

Now, for any $s, \mathcal{E} \in S \times \mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$, from (30):

$$\begin{aligned} p^{\max}_{s,\mathcal{E}}(\phi \, \mathcal{V} \, \psi) &= p^{\max}_{s,\mathcal{E}}\Big(\psi \, \mathcal{U} \, \big((\phi {\wedge} \psi) \vee \neg \mathcal{P}_{<1}[\Box(\neg\phi{\wedge}\psi)]\big) \Big) \\ &= p^{\max}_{s,\mathcal{E}}\Big(\psi \, \mathcal{U} \, \neg \mathcal{P}_{<1} \Big[\psi \, \mathcal{U} \, \big((\phi {\wedge} \psi) \vee \neg \mathcal{P}_{<1}[\Box(\neg\phi{\wedge}\psi)]\big) \Big] \Big) && \text{by (33)} \\ &= p^{\max}_{s,\mathcal{E}}\Big(\psi \, \mathcal{U} \, \neg \mathcal{P}_{<1}[\phi \, \mathcal{V} \, \psi] \Big) && \text{by (32)} \end{aligned}$$

as required. □

Proposition 36 provides us with a method for obtaining the maximum probability of satisfying a release formula: first we obtain the set of states satisfying $\neg \mathcal{P}_{<1}[\phi \, \mathcal{V} \, \psi]$, then we obtain the maximum probability of satisfying $\phi \, \mathcal{U} \, \neg \mathcal{P}_{<1}[\phi \, \mathcal{V} \, \psi]$ (which we have shown in Section 4.2). More precisely, we set $\mathsf{Until}([\![\phi]\!], [\![\psi]\!], \gtrsim \lambda)$ to:

- $[\![\mathtt{true}]\!] \setminus \mathsf{MaxV}_{\geqslant 1}(c, [\![\neg\phi]\!], [\![\neg\psi]\!])$ if $\gtrsim \, = \, >$ and $\lambda{=}0$;
- $[\![\mathtt{true}]\!] \setminus \mathsf{MaxU}_{>0}\Big([\![\neg\psi]\!], \mathsf{MaxV}_{\geqslant 1}(c, [\![\neg\phi]\!], [\![\neg\psi]\!])\Big)$ if $\gtrsim \, = \, \geqslant$ and $\lambda{=}1$;
- $[\![\mathtt{true}]\!] \setminus \mathsf{MaxU}\Big([\![\neg\psi]\!], \mathsf{MaxV}_{\geqslant 1}(c, [\![\neg\phi]\!], [\![\neg\psi]\!]), \not\gtrsim 1{-}\lambda\Big)$ if $\lambda \in (0,1)$;

and $\mathsf{Release}([\![\phi]\!], [\![\psi]\!], \lesssim \lambda)$ equal to:

- $[\![\mathtt{true}]\!] \setminus \mathsf{MaxV}_{\geqslant 1}(c, [\![\phi]\!], [\![\psi]\!])$ if $\lesssim \, = \, <$ and $\lambda{=}1$;
- $[\![\mathtt{true}]\!] \setminus \mathsf{MaxU}_{>0}\Big([\![\psi]\!], \mathsf{MaxV}_{\geqslant 1}(c, [\![\phi]\!], [\![\psi]\!])\Big)$ if $\lesssim \, = \, \leqslant$ and $\lambda{=}0$;
- $[\![\mathtt{true}]\!] \setminus \mathsf{MaxU}\Big([\![\psi]\!], \mathsf{MaxV}_{\geqslant 1}(c, [\![\phi]\!], [\![\psi]\!]), \lesssim \lambda\Big)$ if $\lambda \in (0,1)$.

4.3.1 Example

We now return to the PTA in Example 16 and verify the property $z.\mathcal{P}_{>\lambda}[\phi \, \mathcal{U} \, \psi]$, where $\phi = \mathtt{true}$ and $\psi = (\mathsf{sr} {\wedge} (z{<}6))$, which involves computing the minimal probability of a message being correctly delivered before 6 time units have elapsed is greater than λ. This is achieved through the computation of the maximum probability for the dual release formula $\mathtt{false} \, \mathcal{V} \, \neg(\mathsf{sr} {\wedge} (z{<}6))$, that is, computing the maximum probability of remaining in states where either the message has not been delivered or the clock z is greater than or equal to 6. Similarly to Example 4.2.3, we consider these probabilities when starting from the location di with the clock x equal to 0 and do not distinguish between the name of a location and the atomic proposition with which it is labelled.

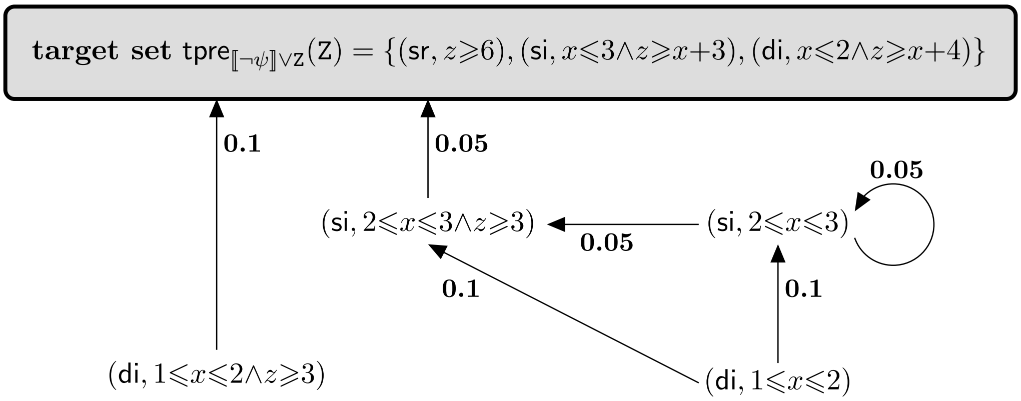
target set $\mathsf{tpre}_{[\![\neg\psi]\!]\vee\mathsf{Z}}(\mathsf{Z}) = \{(\mathsf{sr}, z\geqslant 6), (\mathsf{si}, x\leqslant 3\wedge z\geqslant x+3), (\mathsf{di}, x\leqslant 2\wedge z\geqslant x+4)\}$
0.1
0.05
0.05
$(\mathsf{si}, 2\leqslant x\leqslant 3\wedge z\geqslant 3)$
0.05
$(\mathsf{si}, 2\leqslant x\leqslant 3)$
0.1
0.1
$(\mathsf{di}, 1\leqslant x\leqslant 2\wedge z\geqslant 3)$
$(\mathsf{di}, 1\leqslant x\leqslant 2)$

Fig. 8. Symbolic states and edges generated by $\mathsf{MaxU}([\![\neg\psi]\!], \mathsf{Z}, \geqslant 1{-}\lambda)$

According to our methodology, the set of states satisfying $\mathcal{P}_{>\lambda}[\phi\,\mathcal{U}\,\psi]$ is given by the following set of symbolic states:

$$[\![\mathtt{true}]\!] \setminus \mathsf{MaxU}\Big(\,[\![\neg\psi]\!], \mathsf{MaxV}_{\geqslant 1}(c, [\![\neg\phi]\!], [\![\neg\psi]\!]), \geqslant 1{-}\lambda\,\Big)\,.$$

Therefore, we first compute $\mathsf{MaxV}_{\geqslant 1}(c, [\![\neg\phi]\!], [\![\neg\psi]\!])$, the set of states for which the maximum probability of remaining in states where either the message has not been delivered or the clock z is greater than or equal to 6 is one, which returns (for any positive integer value of c) the set of symbolic states

$$\mathsf{Z} = \{(\mathsf{sr}, z\geqslant 6), (\mathsf{si}, x\leqslant 3 \wedge z\geqslant x{+}3), (\mathsf{di}, x\leqslant 2 \wedge z\geqslant x{+}4)\}\,.$$

The details on the computations performed in the construction of this set of symbolic states can be found in Appendix B and Appendix C.

Next, applying $\mathsf{MaxU}([\![\neg\psi]\!], \mathsf{Z}, \geqslant 1{-}\lambda)$ returns the probabilistic system given in Figure 8. Appendix D presents the computations performed by MaxU in the construction of the states and edges of this probabilistic system. Now $(\mathsf{di}, 1\leqslant x\leqslant 2)$ is the only symbolic state of the probabilistic system given in Figure 8 for which the time predecessor set includes a state and formula clock valuation pair $(\mathsf{di}, x = 0), \mathcal{E}$ such that $\mathcal{E}(z) = 0$. Therefore, using Proposition 29, from location di with x equal to 0 the maximum probability of satisfying $\neg\psi\,\mathcal{U}\,(\neg\mathcal{P}_{<1}[\neg\phi\,\mathcal{V}\,\neg\psi])$ equals the maximum probability of $(\mathsf{di}, 1\leqslant x\leqslant 2)$ reaching $\mathsf{tpre}_{[\![\neg\psi]\!]\vee\mathsf{Z}}(\mathsf{Z})$, and hence equals 0.005.

Finally, using Proposition 36, we have that starting from di with x equal to 0, the minimum probability of correctly delivering before 6 time units have elapsed equals $1 - 0.005 = 0.995$.

We now present a method to check that the probabilistic timed automaton under study is non-zeno. In the non-probabilistic case checking non-zenoness corresponds to finding the greatest fixpoint $\mathit{gfp}\, Y. (z.(\mathtt{true}\ \exists\mathcal{U}\ ((z{=}1) \wedge Y)))$. The states satisfying this expression are those from which there exists a divergent path. For probabilistic timed automata, we can replace $\exists$ with $\neg\mathcal{P}_{<1}[\cdot]$, i.e replace 'there exists a path that reaches $(z{=}1) \wedge Y$' with 'there exists an adversary which reaches $(z{=}1) \wedge Y$ with probability 1'. Following this approach, the algorithm for calculating the set of non-zeno states is given in Figure 7. A probabilistic timed automaton is then non-zeno if and only if the algorithm NonZeno returns the set of symbolic states $[\![\mathtt{true}]\!]$. Formally, we have the following proposition.

Proposition 37 *A probabilistic timed automaton* PTA *is non-zeno if and only if* $\{(l, \mathit{inv}(l)), \mathcal{E} \mid l \in L \text{ and } \mathcal{E} \in \mathbb{R}^{\mathcal{Z}}_{\geqslant 0}\}$ *is characterised by the fixpoint*

$$\mathit{gfp}\, Y. \Big(z.\neg\mathcal{P}_{<1}[\Diamond\ ((z{=}1) \wedge Y)]\Big)\,.$$

Proof. Consider any probabilistic timed automaton PTA and corresponding timed probabilistic system $\mathsf{TPS} = (S, \mathit{TSteps}, \mathcal{L})$. To ease notation we let S_{nz} denote the set of state and formula clock valuation pairs:

$$\Big\{s, \mathcal{E} \in S{\times}\mathbb{R}^{\mathcal{Z}}_{\geqslant 0} \,\Big|\, \exists A \in \mathit{Adv}_{\mathsf{TPS}}.\ \Big(\mathit{Prob}^A_s\{\omega \in \mathit{Path}^A_{\mathit{ful}}(s) \mid \omega \text{ is divergent}\} = 1\Big)\Big\}\,.$$

Letting $G_{\mathrm{nz}}(X) = z.\neg\mathcal{P}_{<1}[\Diamond\ (z{=}1){\wedge}X]$, we prove the proposition by showing that:

(1) the set S_{nz} is a fixpoint of $G_{\mathrm{nz}}(\cdot)$;
(2) if $G_{\mathrm{nz}}(Y) = Y$, then $Y \subseteq S_{\mathrm{nz}}$.

To prove that S_{nz} is a fixpoint of $G_{\mathrm{nz}}(\cdot)$ we show that both $S_{\mathrm{nz}} \subseteq G_{\mathrm{nz}}(S_{\mathrm{nz}})$ and $S_{\mathrm{nz}} \supseteq G_{\mathrm{nz}}(S_{\mathrm{nz}})$.

- For any $s, \mathcal{E} \in S_{\mathrm{nz}}$, by construction there exists an adversary $A \in \mathit{Adv}_{\mathsf{TPS}}$ such that $\mathit{Prob}^A_s\{\omega \in \mathit{Path}^A_{\mathit{ful}}(s) \mid \omega \text{ is divergent}\} = 1$, and hence, with probability 1, under the adversary A one time unit will elapse and we will reach a state in S_{nz}. It follows that $s, \mathcal{E}[z{:=}0] \models \neg\mathcal{P}_{<1}[\Diamond\ (z{=}1){\wedge}S_{\mathrm{nz}}]$, and since $s, \mathcal{E} \in S_{\mathrm{nz}}$ was arbitrary, $S_{\mathrm{nz}} \subseteq G_{\mathrm{nz}}(S_{\mathrm{nz}})$.
- For any $s, \mathcal{E} \in G_{\mathrm{nz}}(S_{\mathrm{nz}})$, by construction there exists an adversary A under which, with probability 1, from s one reaches a state in S_{nz} after 1 time unit. Therefore consider the adversary which behaves as A except that when a state in S_{nz} is reached: in such a case the adversary lets time diverge with probability 1 (such a choice exists by the definition of S_{nz}). It follows that,

under this adversary, time diverges from s with probability 1, and hence $s, \mathcal{E} \in S_{\mathrm{nz}}$. Hence, since $s, \mathcal{E} \in G_{\mathrm{nz}}(S_{\mathrm{nz}})$ was arbitrary, $S_{\mathrm{nz}} \supseteq G_{\mathrm{nz}}(S_{\mathrm{nz}})$.

It therefore remains to show that for any set of state and formula clock valuation pairs Y, if $G_{\mathrm{nz}}(Y) = Y$, then $Y \subseteq S_{\mathrm{nz}}$. The proof is by contradiction: suppose that there exists a set of state and formula clock valuation pairs $Y \subseteq S \times \mathbb{R}^{\mathcal{Z}}_{\geqslant 0}$ such that $G_{\mathrm{nz}}(Y) = Y$ and $Y \backslash S_{\mathrm{nz}} \neq \emptyset$. Now, for any $s, \mathcal{E} \in Y \backslash S_{\mathrm{nz}}$, since $G_{\mathrm{nz}}(Y) = Y$, there exists an adversary for which from s with probability 1 one reaches a state in Y after 1 time unit. Iterating this fact, we have that, for any $n \in \mathbb{N}$, there exists an adversary which with probability 1 lets n time units elapse. Therefore $s, \mathcal{E} \in S_{\mathrm{nz}}$ which is a contradiction. □

Similarly to [7], the algorithm can be used to convert a 'zeno' probabilistic timed automaton into a non-zeno automaton by strengthening invariants. More precisely, supposing NonZeno returns Z, we can construct a new invariant condition by letting $inv_{\mathrm{nz}}(l) = \zeta^{l}_{\mathsf{Z}}$ for each location l of the automaton under study.

4.5 Termination

As in [7], the termination of the model checking algorithms introduced in this paper relies on the fact that only a finite number of zones can be generated by the algorithms. More precisely, from inspection of the definitions of the operations on symbolic states presented in Section 4.1, the zones of the symbolic states computed during our model-checking algorithms will refer only to constants less than or equal to the maximal constant appearing in the probabilistic timed automaton PTA (either in a guard or invariant condition), and the PTCTL formula ϕ (and the parameter c when either of the algorithms $\mathsf{MaxV}_{\geqslant 1}$ and NonZeno are called). Furthermore, the computed symbolic states will refer only to the clocks of PTA and ϕ (and one additional clock when either of the algorithms $\mathsf{MaxV}_{\geqslant 1}$ and NonZeno are called). Hence, only a finite number of symbolic states can be computed during the execution of the algorithms.

That the algorithms of Section 4.2, Section 4.3 and Section 4.4 terminate is a consequence of the following facts. Firstly, the algorithms for qualitative PTCTL formulae and checking non-zenoness (those presented in Figure 4 and Figure 7) correspond to a (possibly nested) fixpoint expression on a monotonic function mapping between sets of symbolic states. Similarly, the algorithm for quantitative formula of Figure 5 corresponds to least fixpoint expressions on a monotonic function mapping between sets of sets of symbolic states. Then, as the number of possible symbolic states is finite, termination of the algorithms is guaranteed.

5 Case Studies

In this section we report on a prototype implementation of the algorithms of Section 4, together with its application to two case studies: the CSMA/CD communication protocol [19], and the IEEE1394 FireWire root contention protocol [20]. We include only the results for the generation of the finite-state probabilistic system, and not the verification of this system which is performed by the probabilistic model-checking tool PRISM, and is therefore standard. Further details are available from the PRISM web page [34].

We confirm the results with those obtained using the digital clocks approach in PRISM [10,8] and, when possible, the 'forward reachability' approach [3,15]. The comparison with the digital clocks approach is feasible because the models are 'closed' and 'diagonal-free' (they do not feature either strict inequalities or comparisons between clocks in their zones), and hence are amenable to discrete-time analysis; however, our algorithms are applicable to general probabilistic timed automata. When calculating minimum probabilities of deadline properties, for comparison we also use an alternative method introduced in [8], as explained by the following remark.

Remark 38 *We observe that certain deadline properties referring to minimum probability can be expressed in terms of properties referring to maximum probability. Consider a property* $z.\mathcal{P}_{\geqslant\lambda}[\diamond(\phi \wedge (z \leqslant D))]$ *and assume that* ϕ *is reachable with probability 1 for all adversaries and states. We adjust the model so that states in which* ϕ *is true are forced to make a transition to a sink-location; furthermore, we allow the model to make a transition to a different, 'deadline exceeded' sink-location, denoted* $\mathsf{exceeded}$*, as soon as the value of the clock* z *exceeds* D *[8] (provided that we are not in a state satisfying* ϕ*). We define the labelling of the location* $\mathsf{exceeded}$ *so that* ϕ *is not true in this location, and, because* $\mathsf{exceeded}$ *is a sink,* ϕ *cannot become true after it is entered. Then, given any adversary* A*, state* s *and formula clock valuation* $\mathcal{E}$*, we have that* $p^A_{s,\mathcal{E}}(\diamond(\phi \wedge (z \leqslant D))) = 1 - p^A_{s,\mathcal{E}}(\diamond\ \mathsf{exceeded})$*, and* $s, \mathcal{E} \models z.\mathcal{P}_{\geqslant\lambda}[\diamond\ (\phi \wedge (z \leqslant D))]$ *if and only if* $s, \mathcal{E} \models \mathcal{P}_{\leqslant 1-\lambda}[\diamond\ \mathsf{exceeded}]$*. Hence, we are able to reduce the computation of a minimum probability to a maximum probability in such situations.*

5.1 Implementation

In this section we briefly summarise our prototype implementation of the model-checking algorithms given in Section 4. It is important to note that the aim of our implementation is to validate the algorithms presented for model checking probabilistic timed automata against PTCTL, rather than to devise an efficient implementation; the latter will be the subject of future work. Note that, to perform the final step of the algorithm MaxU (line 22 of Figure 5), that is to compute maximum reachability probabilities on a finite-state probabilistic system, we export the problem to the probabilistic symbolic model checker PRISM [35,34].

The main step in the implementation of our techniques is the representation of (sets of) symbolic states and the operations required on them. More precisely, since a symbolic state is a pair (l, ζ) where $l \in L$ and ζ is a zone, we require a method for representing zones and performing operations on zones.

Difference Bound Matrices (DBMs) [18] are a well known data-structure for the representation of convex zones and are used in the model checkers Uppaal [13] and Kronos [14]. As the operations required by our algorithm can introduce non-convexity, we also represent non-convex zones. Following the approach presented in [36,37,28], we represent non-convex zones by lists of DBMs; that is, we represent a non-convex zone ζ by a list of convex zones $\zeta_1, \ldots, \zeta_n$ such that $\zeta = \zeta_1 \cup \cdots \cup \zeta_n$. It thus follows that a symbolic state can be represented by a location and a list of DBMs. Recall that [28] presents algorithms (used by Kronos [38]) for the operations we require when zones are represented as lists of DBMs. Based on [28], we have implemented, in Java, a prototype DBM package and the operations on lists of DBMs required by our model-checking algorithms. Note that the equality checking performed by the algorithms $\mathsf{MaxV}_{\geqslant 1}$, $\mathsf{MaxU}_{\geqslant 1}$ and $\mathsf{NonZeno}$ reduces to an inclusion test based on whether a least or greatest fixpoint is being performed.

5.2 CSMA/CD protocol

We proceed to describe the application of our prototype implementation to the first case study. The CSMA/CD (Carrier Sense, Multiple Access with Collision Detection) protocol is designed for networks with a single channel and specifies the behaviour of stations with the aim of minimising simultaneous use of the channel (data collision). The basic structure of the protocol is as follows: when a station has data to send, it listens to the medium, after which, if the medium was free (no other station is transmitting), the station starts to send its data. On the other hand, if the medium was sensed busy, the station waits a random amount of time, based on the number of failed transmissions of the packet, and then repeats this process. The model we consider here is a probabilistic extension of the timed automata model given in [39]. We consider the case when there are two stations trying to send data at the same time. The overall model is given by the parallel composition of three probabilistic timed automata, representing the medium and two stations trying to send data. The following parameters are taken from the IEEE standard 802.3 for

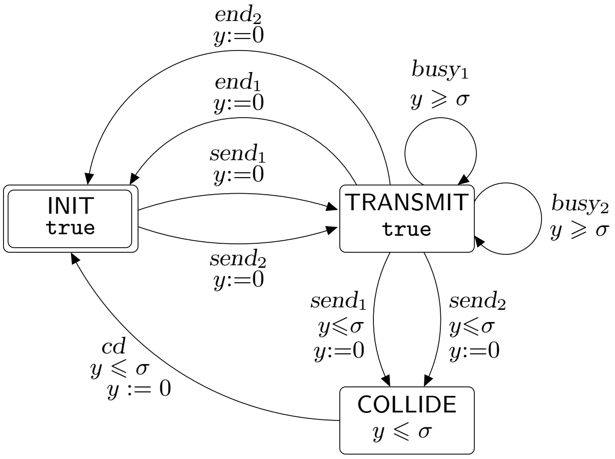

Fig. 9. A probabilistic timed automaton modelling the medium.

10 Mbps Ethernet [19].

- Propagation delay of the channel $\sigma = 26\mu s$.
- Time to send a data packet (plus the propagation delay) $\lambda = 808\mu s$.
- Length of one time slot (used in the randomised truncated binary exponential backoff process) $slotTime = 2 \cdot \sigma$.

Our model of the protocol is obtained from three probabilistic timed automata sub-models which are composed in parallel using synchronisation on common, probabilistic edge labelling events. The formal definition of event-labelled probabilistic timed automata, and of their parallel composition is presented in [8]. We proceed to explain our probabilistic timed automata sub-models in turn.

The Medium The probabilistic timed automaton representing the medium is given in Figure 9. The medium is initially ready to accept data from any station (event $send_i$ for $i \in \{1,2\}$). Once a station starts sending its data there is an interval of time (at most σ), representing the time it takes for a signal to propagate between the stations, in which the medium will accept data from the other station (possibly resulting in a collision). After this interval, if the other station tries to send data it will get the busy signal ($busy_i$). If a collision occurs, there is a delay (again at most σ) before the stations realise there has been a collision, after which the medium will become free (represented by the event cd). If the stations do not collide, then when a station finishes sending its data (event end_i) the medium becomes idle.

Note that the guard ($y \leqslant \sigma$) on the transitions from TRANSMIT to COLLIDE differs from that of the model of [39], in which the inequality is strict. The reason we have made this change is to allow us to use the integer semantics approach where, unlike in the model checking algorithms presented in this paper,

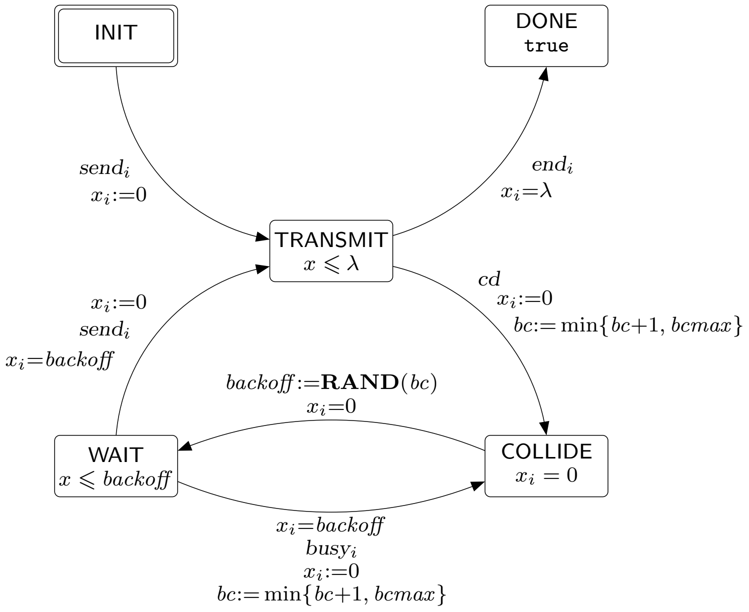

Fig. 10. A probabilistic timed automaton modelling a station.

only non-strict inequalities are allowed in the probabilistic timed automaton under study.

The Stations In Figure 10 we have presented the probabilistic timed automata model of a station. Note that, as in [39], we assume that only packets of equal length are sent. Observe also that we use bounded-range, natural numbered variables within the model description. Such variables can be represented within the probabilistic timed automaton framework by encoding their values within 'copies' of locations (one copy for each possible valuation of the variables). We can then permit random assignment to such variables, because such assignment corresponds to probabilistic choice between the copies of a location.

A station starts in location INIT with the values of its clock x_i and its discrete variables *bc* and *backoff* equal to zero. The behaviour of the station commences by the sending of data (event *send*$_i$). If there is no collision, then, after λ time units, the station finishes sending its data (event *end*$_i$). On the other hand, if there is a collision (event *cd*), the station attempts to retransmit the packet where the scheduling of the retransmission is determined by a *truncated binary exponential backoff* process. The delay before retransmitting is is measured as an integer number of time slots (each of length *slotTime*). The number of slots that the station waits after the nth transmission failure is chosen as a uniformly distributed random integer in the range:

$$0, 1, 2, \ldots, 2^{bc+1}-1$$

where $bc = \min(n, bcmax)$ and $bcmax$ is the constant referring to the trunca-

Table 1. Statistics for $\mathsf{MaxV}_{\geqslant 1}$ as c varies, when verifying the CSMA/CD protocol

bcmax	*c*	$\mathcal{P}_{\geqslant 1}[\Diamond\, \mathsf{done}]$			$z.\mathcal{P}_{\gtrsim\lambda}[\Diamond(\mathsf{done} \wedge z{\leqslant}2000)]$		
		time	iterations		time	iterations	
		(sec)	$\mathsf{MaxV}_{\geqslant 1}$	$\mathsf{MaxU}_{\geqslant 1}$	(sec)	$\mathsf{MaxV}_{\geqslant 1}$	$\mathsf{MaxU}_{\geqslant 1}$
1	1	1,048	966	7,409	269.6	132	1,062
	10	106.7	99	755	31.94	15	126
	26	48.38	40	321	21.60	9	94
	30	44.31	35	281	27.75	9	94
	40	34.06	27	218	18.20	7	78
	50	29.88	22	184	18.67	6	70
	60	88.01	21	168	201.2	8	91
	70	79.83	18	150	199.3	8	91
	80	72.39	16	141	182.4	7	84
	90	72.17	15	123	179.5	7	84
	100	66.20	13	121	189.3	7	84
	200	57.18	7	87	83.67	6	90
	808	474.9	4	89	659.5	5	115
2	1	2,058	1,070	7,830	660.9	236	1,899
	10	223.1	109	800	78.58	26	219
	26	126.6	44	347	93.48	13	136
	30	142.4	38	303	86.36	12	128
	40	109.6	29	237	73.51	10	108
	50	94.77	24	201	64.65	8	92
	60	213.6	22	177	270.7	8	91
	70	179.3	19	157	270.9	8	91
	80	149.5	16	136	272.4	8	88
	90	158.3	16	130	271.2	8	88
	100	115.6	14	117	256.7	8	88
	200	232.6	9	93	1,101	7	94
	808	21,682	5	147	103,134	6	217

tion point of the backoff process. The slot length and the randomly-chosen integer are combined within the probabilistic assignment *backoff* $:= \mathbf{RAND}(bc)$. Once *backoff* time units have elapsed, if the medium appears free the station resends the data (event *send*), while if the medium is sensed busy (event *busy*) the station repeats this process.

Note that, to simplify the model, we have removed the limit on the number of times a station attempts to retransmit a packet as specified in the standard. For our experiments we consider the cases when *bcmax* is either 1 or 2.

5.2.1 Model Checking

The first property we check is that the minimum probability that both stations correctly deliver their packets is 1; that is, we verify $\mathcal{P}_{\geqslant 1}[\Diamond\, \mathsf{done}]$ where done is the atomic proposition labelling these states where both stations are in the location DONE. From Section 4.3, the verification of such a property requires a call to the algorithm $\mathsf{MaxV}_{\geqslant 1}$ within a call to $\mathsf{MaxU}_{>0}$; more precisely, the set of states satisfying $\mathcal{P}_{\geqslant 1}[\Diamond\, \mathsf{done}]$ is given by:

$$[\![\mathtt{true}]\!] \setminus \mathsf{MaxU}_{>0}\Big(\, [\![\neg\mathsf{done}]\!], \mathsf{MaxV}_{\geqslant 1}(c, [\![\mathtt{false}]\!], [\![\neg\mathsf{done}]\!])\,\Big)\,.$$

Table 2. Model sizes (and generation times in seconds) for CSMA/CD protocol

bcmax	D	$z.\mathcal{P}_{\sim\lambda}[\Diamond(\mathsf{done} \wedge z \leqslant D)]$				$\mathcal{P}_{\lesssim\lambda}[\Diamond\mathsf{exceeded}]$		digital clocks
	(μs)	maximum [$\sim\ = \ \lesssim$]		minimum [$\sim\ = \ \gtrsim$]				[10,8]
	1000	71	(1.177)	351	(40.96)	362	(11.44)	1,876,105
	1200	191	(11.70)	351	(41.29)	362	(11.49)	2,671,305
	1400	311	(26.51)	351	(41.98)	362	(11.48)	3,546,505
	1600	431	(55.65)	351	(41.64)	362	(11.46)	4,501,705
	1800	617	(76.68)	441	(45.65)	440	(14.76)	5,528,692
1	2000	725	(84.33)	591	(57.52)	562	(20.08)	6,570,692
	2200	861	(98.82)	783	(69.67)	722	(37.24)	7,612,692
	2400	997	(120.5)	975	(97.70)	882	(75.31)	8,654,692
	2600	1,129	(145.3)	1,143	(133.7)	1,022	(103.0)	9,696,692
	2800	1,263	(174.2)	1,335	(188.1)	1,182	(182.7)	10,738,692
	3000	1,399	(239.8)	1,527	(278.3)	1,342	(261.2)	11,780,692
	1000	91	(2.176)	724	(232.6)	737	(171.8)	4,170,287
	1200	423	(178.0)	724	(242.3)	737	(169.0)	5,813,169
	1400	759	(833.4)	724	(260.5)	737	(163.0)	7,535,969
	1600	1,095	(192.6)	724	(244.5)	737	(164.0)	9,338,769
	1800	1,834	(337.8)	1,203	(1,430)	1,208	(589.9)	11,211,180
2	2000	2,615	(600.6)	1,760	(5,646)	1,751	(1,374)	13,072,580
	2200	3,019	(655.9)	2,170	(6,455)	2,145	(2,104)	14,930,180
	2400	3,415	(706.7)	2,578	(9,326)	2,537	(3,301)	16,787,780
	2600	3,795	(773.9)	2,935	(12,335)	2,880	(5,144)	18,645,380
	2800	4,183	(8,762)	3,343	(15,153)	3,272	(7,875)	20,502,980
	3000	4,579	(9,852)	3,751	(16,936)	3,664	(9,835)	22,360,580

The algorithm $\mathsf{MaxV}_{\geqslant 1}$ returns no symbolic states, and thus the $\mathsf{MaxU}_{>0}$ algorithm also trivially returns no symbolic states, which implies that all states satisfy $\mathcal{P}_{\geqslant 1}[\Diamond\, \mathsf{done}]$.

In Table 1 we give the model-checking statistics for $\mathsf{MaxV}_{\geqslant 1}$ as we vary the parameter c (where 26 and 808 are the smallest and largest non-zero constants appearing in the model). The results presented show that, as c increases, the number of iterations of the $\mathsf{MaxV}_{\geqslant 1}$ algorithm decreases, while the number of iterations required by each call to the algorithm $\mathsf{MaxU}_{\geqslant 1}$ increases (recall that the $\mathsf{MaxU}_{\geqslant 1}$ algorithm is called once in each iteration of the $\mathsf{MaxV}_{\geqslant 1}$ algorithm). As in the non-probabilistic case [40], further investigations and case studies are needed to establish if there is any way of finding a 'good' choice for the parameter c in advance.

The remaining properties we consider are the maximum and minimum probabilities that both stations deliver their packets by time D; that is, the property $z.\mathcal{P}_{\sim\lambda}[\Diamond(\mathsf{done} \wedge (z \leqslant D))]$. In Table 2 we have presented the model sizes (and generation times) of the finite-state probabilistic system generated by our implementation and, for comparison, the size of the model constructed using the digital clocks approach [10,8] (there are no generation times in this case as the digital semantics leads directly to a finite-state system). The results show a significant decrease in the model size when compared to the digital clocks approach. Comparing the results for $z.\mathcal{P}_{\geqslant\lambda}[\Diamond(\mathsf{done} \wedge (z \leqslant D))]$ and $\mathcal{P}_{\leqslant\lambda}[\Diamond\mathsf{exceeded}]$, we see that using Remark 38 can decrease both the states

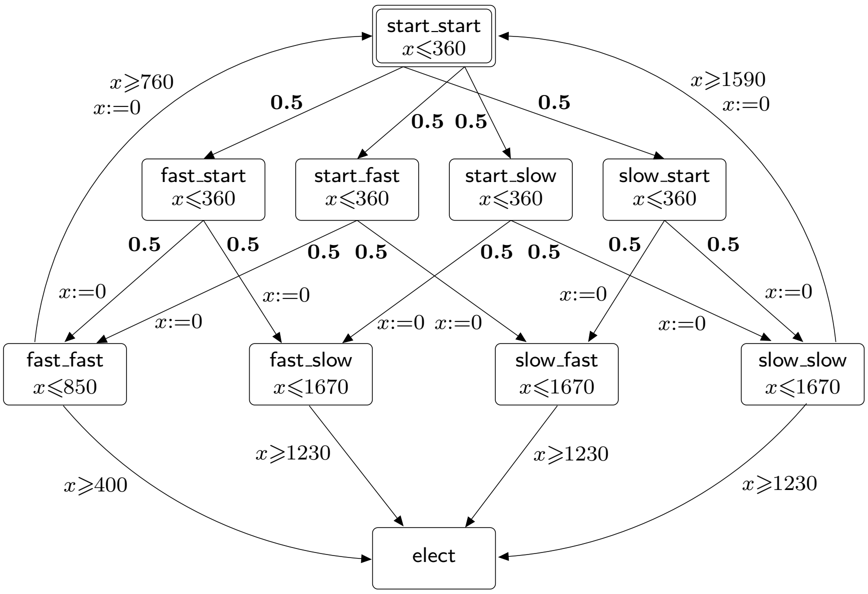

Fig. 11. The probabilistic timed automaton $\mathtt{I}_1^{\mathtt{p}}$.

and generation time. Table 1 includes the model-checking statistics for the $\mathsf{MaxV}_{\geqslant 1}$ algorithm when verifying $z.\mathcal{P}_{\sim\lambda}[\Diamond(\mathsf{done} \wedge (z\leqslant 2000))]$, and we see a similar pattern to that obtained for $\mathcal{P}_{\geqslant 1}[\Diamond\, \mathsf{done}]$.

5.3 FireWire root contention protocol

We consider the abstract probabilistic timed automaton model $\mathtt{I}_1^{\mathtt{p}}$ (see Figure 11), which is a probabilistic extension of the classical timed automaton $\mathtt{I}_1$ of [41], as studied in [15,8]. The IEEE1394 FireWire root contention protocol concerns the election of a leader between two contending nodes of a network. The protocol consists of a number of rounds in which each of the contending nodes flips a coin; given the result of the coin flip, a node may decide to wait for a short amount of time or a long amount of time. After this amount of time has elapsed, a node then checks to see if the other node has already deferred, and declares itself to be the leader if so; otherwise, this node defers. Intuitively, in the case in which the result of the two nodes' coin flips are different, the 'faster' node defers to the 'slower' node, the latter of which then becomes leader, signalling the end of protocol execution. However, if the results of the coin flips are the same, the communication delay between the two nodes means that it is possible that both nodes attempt to defer to the other, requiring another round of the protocol.

The timing constraints are derived from those given in the standard when the communication delay is 360ns. The properties we consider concern the minimum probability to elect a leader with and without a deadline, that is, the properties $\mathcal{P}_{\geqslant\lambda}[\Diamond\, \mathsf{elect}]$ and $z.\mathcal{P}_{\geqslant\lambda}[\Diamond(\mathsf{elect} \wedge (z\leqslant D))]$.

Table 3. Statistics for $\mathsf{MaxV}_{\geqslant 1}$ as c varies, when verifying $\mathtt{I}_1^{\mathtt{p}}$

c	$\mathcal{P}_{\geqslant 1}[\Diamond\ \mathsf{elect}]$			$z.\mathcal{P}_{\gtrsim\lambda}[\Diamond(\mathsf{elect}\wedge z\leqslant 10000)]$		
	time	iterations		time	iterations	
	(sec)	$\mathsf{MaxV}_{\geqslant 1}$	$\mathsf{MaxU}_{\geqslant 1}$	(sec)	$\mathsf{MaxV}_{\geqslant 1}$	$\mathsf{MaxU}_{\geqslant 1}$
10	23.66	372	1597	7.800	50	304
100	2.804	39	171	2.141	11	70
360	1.246	13	67	1.692	7	55
1,670	0.679	5	30	1.641	5	46
2,000	0.764	4	30	1.566	4	42
3,000	0.514	4	23	1.312	4	41
4,000	0.495	3	22	1.083	3	32
5,000	0.520	3	24	1.063	3	34
6,000	0.532	3	25	1.078	3	35
7,000	0.570	3	27	1.123	3	37
8,000	0.577	3	28	1.143	3	38
9,000	0.612	3	30	1.180	3	40
10,000	0.621	3	31	1.199	3	41

When verifying $\mathcal{P}_{\geqslant\lambda}[\Diamond\ \mathsf{elect}]$, the algorithm $\mathsf{MaxV}_{\geqslant 1}$ returns no symbolic states, which implies that the probability is 1 in all states. In Table 3 we give the model-checking statistics for the $\mathsf{MaxV}_{\geqslant 1}$ algorithm as the value of c changes (360 and 1670 are the smallest and largest non-zero constants appearing in the model). As for the CSMA/CD case study, we see that increasing c decreases the number of iterations required by $\mathsf{MaxV}_{\geqslant 1}$, while increasing the iterations performed by each call to the algorithm $\mathsf{MaxU}_{\geqslant 1}$.

In Table 4 we have reported on the size and generation times in seconds when verifying $z.\mathcal{P}_{\geqslant\lambda}[\Diamond(\mathsf{elect} \wedge (z\leqslant D))]$ for a range of deadlines. As for the CSMA/CD case study, we can use Remark 38 and instead verify $\mathcal{P}_{\leqslant\lambda}[\Diamond\mathsf{exceeded}]$ on a modified model. Additionally, in Table 4 we include the results obtained when applying the forwards approach [3,15] and using digital clocks [10,8]. Note that the approach of [3,15] cannot be used to calculate the minimum probability of eventually electing a leader. The results show that the use of the algorithms presented in this paper leads to a smaller state space than the other approaches. Comparing the results obtained when verifying $z.\mathcal{P}_{\geqslant\lambda}[\Diamond(\mathsf{elect} \wedge (z\leqslant D))]$ and $\mathcal{P}_{\leqslant\lambda}[\Diamond\mathsf{exceeded}]$, we see that the direct approach leads to a smaller state space and, for large deadlines, is faster than the approach based on Remark 38. The generation times for our prototype implementation are considerably greater than those obtained with the forwards approach. This is due to the fact that the latter are generated with the optimised tool KRONOS, and also to the fact that the operations on state sets in the forwards approach are simpler than those used in the techniques given in this paper (in particular, the forwards approach does not generate non-convex zones, and does not require the computation of nested fixpoints). However, recall that the forwards approach can be used only to compute an upper bound on the maximal probability of reaching a state set; instead our techniques can compute *exact* probabilities for a richer class of properties.

Table 4. Model sizes (and generation times in seconds) when verifying $\mathtt{I}_1^{\mathtt{p}}$

D	$z.\mathcal{P}_{\gtrsim\lambda}[\Diamond(\mathsf{elect}\wedge z\leqslant D)]$		$\mathcal{P}_{\lesssim\lambda}[\Diamond\mathsf{exceeded}]$		forwards		digital clocks
(10^3ns)					[15]		[10,8]
2	15	(2.71)	25	(0.203)	53	(0.00)	68,056
4	25	(3.01)	47	(0.261)	131	(0.00)	220,565
6	47	(3.05)	47	(0.431)	216	(0.01)	375,765
8	81	(2.42)	126	(0.803)	372	(0.02)	530,965
10	126	(2.70)	183	(0.979)	526	(0.03)	686,165
20	528	(12.1)	643	(16.8)	1,876	(0.09)	1,462,165
30	1,206	(113.5)	1,380	(179.2)	4,049	(0.20)	2,238,165
40	2,168	(1,032)	2,395	(1,523)	7,034	(0.46)	3,014,165
50	3,426	(6,465)	3,714	(8,880)	10,865	(1.23)	3,790,165
60	4,964	(26,997)	5,308	(34,986)	15,511	(2.74)	4,566,165

6 Conclusions

We have presented the theoretical foundations for the symbolic model checking of probabilistic timed automata and PTCTL and validated them through a prototype implementation using DBMs. For quantitative formulae, our algorithm is expensive, as, in the worst case, the MaxU algorithm constructs the powerset of the region graph, which itself is exponential in the largest constant used in zones and the number of clocks. However, for the case studies considered, we observe much smaller state spaces than this upper bound, which confirms that the algorithms are feasible in practice. Note that we do not construct a partition of the state space (as in [42], for example), but rather a (property dependent) set of overlapping symbolic states to avoid potentially expensive disjunction operations on zones within MaxU.

Future work will address the efficient symbolic implementation of the presented algorithms, adaptations to probabilistic polyhedral hybrid automata and symbolic probabilistic systems [43] (a probabilistic formulation of the symbolic transition systems of [44]) and a comparison of our approach with state partitioning techniques, for example [42], extended to the probabilistic setting.

References

[1] E. Clarke, O. Grumberg, D. Peled, Model Checking, MIT Press, 1999.

[2] J. Burch, E. Clarke, K. McMillan, D. Dill, L. Hwang, Symbolic model checking: 10^{20} states and beyond, Information and Computation 98 (2) (1992) 142–170.

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

[4] H. Jensen, Model checking probabilistic real time systems, in: B. Bjerner, M. Larsson, B. Nordström (Eds.), Proc. 7th Nordic Workshop on Programming Theory, Report 86, Chalmers University of Technology, 1996, pp. 247–261.

[5] D. Beauquier, Probabilistic timed automata, Theoretical Computer Science 292 (1) (2003) 65–84.

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

[7] T. Henzinger, X. Nicollin, J. Sifakis, S. Yovine, Symbolic model checking for real-time systems, Information and Computation 111 (2) (1994) 193–244.

[8] M. Kwiatkowska, G. Norman, J. Sproston, Probabilistic model checking of deadline properties in the IEEE 1394 FireWire root contention protocol, Formal Aspects of Computing 14 (2003) 295–318.

[9] M. Kwiatkowska, G. Norman, J. Sproston, Probabilistic model checking of the IEEE 802.11 wireless local area network protocol, in: H. Hermanns, R. Segala (Eds.), Proc. 2nd Joint International Workshop on Process Algebra and Probabilistic Methods, Performance Modeling and Verification (PAPM/PROBMIV'02), Vol. 2399 of Lecture Notes in Computer Science, Springer-Verlag, 2002, pp. 169–187.

[10] M. Kwiatkowska, G. Norman, D. Parker, J. Sproston, Performance analysis of probabilistic timed automata using digital clocks, in: K. Larsen, P. Niebert (Eds.), Proc. 1st International Workshop on Formal Modeling and Analysis of Timed Systems (FORMATS'03), Vol. 2791 of LNCS, Springer-Verlag, 2003, pp. 105–120.

[11] H. Hansson, B. Jonsson, A logic for reasoning about time and reliability, Formal Aspects of Computing 6 (4) (1994) 512–535.

[12] R. Alur, C. Courcoubetis, D. Dill, Model checking in dense real time, Information and Computation 104 (1) (1993) 2–34.

[13] G. Behrmann, A. David, K. Larsen, O. Möller, P. Pettersson, W. Yi, Uppaal - present and future, in: Proceedings of the 40th IEEE Conference on Decision and Control (CDC'01), Vol. 3, IEEE Computer Society Press, 2001, pp. 2881–2886.

[14] C. Daws, A. Olivero, S. Tripakis, S. Yovine, The tool Kronos, in: R. Alur, T. Henzinger, E. Sontag (Eds.), Hybrid Systems III: Verification and Control, Vol. 1066 of Lecture Notes in Computer Science, Springer-Verlag, 1996, pp. 208–219.

[15] C. Daws, M. Kwiatkowska, G. Norman, Automatic verification of the IEEE 1394 root contention protocol with Kronos and PRISM, International Journal on Software Tools for Technology Transfer (STTT) 5 (2–3) (2004) 221–236.

[16] L. de Alfaro, Formal verification of probabilistic systems, Ph.D. thesis, Stanford University (1997).

[17] A. Bianco, L. de Alfaro, Model checking of probabilistic and nondeterministic systems, in: P. Thiagarajan (Ed.), Proc. 15th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'95), Vol. 1026 of Lecture Notes in Computer Science, Springer-Verlag, 1995, pp. 499–513.

[18] D. Dill, Timing assumptions and verification of finite-state concurrent systems, in: J. Sifakis (Ed.), Proc. Automatic Verification Methods for Finite State Systems, Vol. 407 of Lecture Notes in Computer Science, Springer-Verlag, 1990, pp. 197–212.

[19] IEEE 802.3-2002, Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Standard (2002).

[20] IEEE 1394-1995, High Performance Serial Bus Standard (1995).

[21] M. Kwiatkowska, G. Norman, J. Sproston, F. Wang, Symbolic model checking for probabilistic timed automata, in: Y. Lakhnech, S. Yovine (Eds.), Joint Conference on Formal Modelling and Analysis of Timed Systems (FORMATS) and Formal Techniques in Real-Time and Fault Tolerant Systems (FTRTFT), Vol. 3253 of Lecture Notes in Computer Science, Springer-Verlag, 2004, pp. 293–308.

[22] J. Kemeny, J. Snell, A. Knapp, Denumerable Markov Chains, 2nd Edition, Springer-Verlag, 1976.

[23] C. Derman, Finite-State Markovian Decision Processes, New York: Academic Press, 1970.

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

[25] Z. Manna, A. Pnueli, The Temporal Logic of Reactive and Concurrent Systems: Specification, Springer Verlag, New York, 1992.

[26] R. Segala, Modelling and verification of randomized distributed real time systems, Ph.D. thesis, Massachusetts Institute of Technology (1995).

[27] D. Dill, Timing assumptions and verification of finite-state concurrent system, in: J. Sifakis (Ed.), Proc. International Workshop on Automatic Verification Methods for Finite State Systems, Vol. 407 of Lecture Notes in Computer Science, Springer-Verlag, 1989, pp. 197–212.

[28] S. Tripakis, L'analyse formelle des systèmes temporisés en pratique, Ph.D. thesis, Université Joseph Fourier (1998).

[29] E. Clarke, E. Emerson, A. Sistla, Automatic verification of finite-state concurrent systems using temporal logics, ACM Transactions on Programming Languages and Systems 8 (2) (1986) 244–263.

[30] C. Baier, On algorithmic verification methods for probabilistic systems, habilitation thesis, Fakultät für Mathematik & Informatik, Universität Mannheim (1998).

[31] S. Hart, M. Sharir, A. Pnueli, Termination of probabilistic concurrent programs, ACM Transactions on Programming Languages and Systems 5 (3) (1983) 356–380.

[32] A. Pnueli, On the extremely fair treatment of probabilistic algorithms, in: Proc. 15th Annual ACM Symposium on Theory of Computing (STOC'83), ACM Press, 1983, pp. 278–290.

[33] L. de Alfaro, M. Kwiatkowska, G. Norman, D. Parker, R. Segala, Symbolic model checking of concurrent probabilistic processes using MTBDDs and the Kronecker representation, in: S. Graf, M. Schwartzbach (Eds.), Proc. 6th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'00), Vol. 1785 of LNCS, Springer, 2000, pp. 395–410.

[34] PRISM Web site, www.cs.bham.ac.uk/˜dxp/prism.

[35] M. Kwiatkowska, G. Norman, D. Parker, PRISM: Probabilistic symbolic model checker, in: T. Field, P. Harrison, J. Bradley, U. Harder (Eds.), Proc. TOOLS'02, Vol. 2324 of Lecture Notes in Computer Science, Springer-Verlag, 2002, pp. 200–204.

[36] S. Yovine, Méthodes et outils pour la vérification symbolique de systemés temporisés, Ph.D. thesis, Institut National Polytechnique de Grenoble (1993).

[37] A. Olivero, Modélisation et analyse de systémes temporisés et hybrides, Ph.D. thesis, Institut National Polytechnique de Grenoble (1994).

[38] C. Daws, Private communication (2004).

[39] X. Nicollin, J. Sifakis, S. Yovine, Compiling real-time specifications into extended automata, IEEE Transactions on Software Engineering 18 (9) (1992) 794–804.

[40] F. Wang, G.-D. Hwang, F. Yu, TCTL inevitability analysis of dense-time systems, in: O. H. Ibarra, Z. Dang (Eds.), Proc. 8th International Conference on Implementation and Application of Automata (CIAA'03), Vol. 2759 of Lecture Notes in Computer Science, Springer-Verlag, 2003, pp. 176–187.

[41] D. Simons, M. Stoelinga, Mechanical verification of the IEEE 1394a root contention protocol using Uppaal2k, Springer International Journal of Software Tools for Technology Transfer (STTT) 3 (4) (2001) 469–485.

[42] R. Alur, C. Courcoubetis, D. Dill, N. Halbwachs, H. Wong-Toi, Minimization of timed transition systems, in: R. Cleaveland (Ed.), Proc. 3rd International Conference on Concurrency Theory (CONCUR'92), Vol. 630 of Lecture Notes in Computer Science, Springer-Verlag, 1992, pp. 340–354.

[43] M. Kwiatkowska, G. Norman, J. Sproston, Symbolic computation of maximal probabilistic reachability, in: K. Larsen, M. Nielsen (Eds.), Proc. 13th International Conference on Concurrency Theory (CONCUR'01), Vol. 2154 of Lecture Notes in Computer Science, Springer-Verlag, 2001, pp. 169–183.

[44] T. Henzinger, R. Majumdar, J.-F. Raskin, A classification of symbolic transition systems, ACM Transactions on Computational Logic 6 (1) (2004) 1–32.

A $\mathsf{MaxU}([\![\mathtt{true}]\!], [\![\mathsf{sr}{\wedge}(z{<}6)]\!], \gtrsim \lambda)$

```
Z := $\{(\mathsf{sr}, z<6)\}$
repeat

    Y := Z
    begin for
        z = $(\mathsf{sr}, z<6)$ [two edges (from si and di) taking predecessors]
            y$_1$ = $(\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z<6)$
            $E_{\mathsf{si},g,p} = \{(\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z<6), (\mathsf{sr}, \{x\}), (\mathsf{sr}, z<6))\}$
            y$_2$ = $(\mathsf{di}, 1\leqslant x\leqslant 2 \wedge z<6)$
            $E_{\mathsf{di},g,p} = \{(\mathsf{di}, 1\leqslant x\leqslant 2 \wedge z<6), (\mathsf{sr}, \{x\}), (\mathsf{sr}, z<6))\}$
    end for
    Z := $\{(\mathsf{sr}, z<6), (\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z<6), (\mathsf{di}, 1\leqslant x\leqslant 2 \wedge z<6)\}$

    Y := Z
    begin for
        z = $(\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z<6)$ [two edges (from si and di) taking predecessors]
            y$_1$ = $(\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z<4)$
            $E_{\mathsf{si},g,p} = E_{\mathsf{si},g,p} \cup \{(\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z<4), (\mathsf{si}, \{x\}), (\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z<6))\}$
            y$_2$ = $(\mathsf{di}, 1\leqslant x\leqslant 2 \wedge z<5)$
            $E_{\mathsf{di},g,p} = E_{\mathsf{di},g,p} \cup \{(\mathsf{di}, 1\leqslant x\leqslant 2 \wedge z<5), (\mathsf{si}, \{x\}), (\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z<6))\}$
        z = $(\mathsf{di}, 1\leqslant x\leqslant 2 \wedge z<6)$ [no edges]
    end for
    Z := $\{(\mathsf{sr}, z\geqslant 6),$
        $(\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z<6), (\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z<4),$
        $(\mathsf{di}, 1\leqslant x\leqslant 2 \wedge z<6), (\mathsf{di}, 1\leqslant x\leqslant 2 \wedge z<5)\}$

    Y := Z
    begin for
        z = $(\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z<4)$ [two edges (from si and di) taking predecessors]
            y$_1$ = $(\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z<2)$
            $E_{\mathsf{si},g,p} = E_{\mathsf{si},g,p} \cup \{(\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z<2), (\mathsf{si}, \{x\}), (\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z<6))\}$
            y$_2$ = $(\mathsf{di}, 1\leqslant x\leqslant 2 \wedge z<3)$
            $E_{\mathsf{di},g,p} = E_{\mathsf{di},g,p} \cup \{(\mathsf{di}, 1\leqslant x\leqslant 2 \wedge z<3), (\mathsf{si}, \{x\}), (\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z<6))\}$
        z = $(\mathsf{di}, 1\leqslant x\leqslant 2 \wedge z<5)$ [no edges]
    end for
    Z := $\{(\mathsf{sr}, z\geqslant 6),$
        $(\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z<6), (\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z<4), (\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z<2),$
        $(\mathsf{di}, 1\leqslant x\leqslant 2 \wedge z<6), (\mathsf{di}, 1\leqslant x\leqslant 2 \wedge z<5), (\mathsf{di}, 1\leqslant x\leqslant 2 \wedge z<3)\}$

    Y := Z
    begin for
        z = $(\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z<2)$ [two edges (from si and di) taking predecessors]
            y$_1$ = $(\mathsf{si}, \mathtt{false})$
            y$_2$ = $(\mathsf{di}, \mathtt{false})$
        z = $(\mathsf{di}, 1\leqslant x\leqslant 2 \wedge z<3)$ [no edges]
    end for
    Z := $\{(\mathsf{sr}, z\geqslant 6),$
        $(\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z<6), (\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z<4), (\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z<2),$
        $(\mathsf{di}, 1\leqslant x\leqslant 2 \wedge z<6), (\mathsf{di}, 1\leqslant x\leqslant 2 \wedge z<5), (\mathsf{di}, 1\leqslant x\leqslant 2 \wedge z<3)\}$

end repeat
```

B $\mathsf{MaxV}_{\geqslant 1}(c, [\![\mathtt{false}]\!], [\![\mathsf{si}\vee\mathsf{di}\vee z\geqslant 6]\!])$

```
$\mathtt{Z} := [\![\mathtt{true}]\!]$
repeat

    $\mathtt{Y} := \mathtt{Z}$
    $\mathtt{Z} := [\![\mathsf{si}\vee\mathsf{di}\vee z\geqslant 6]\!] \wedge y.\mathsf{MaxU}_{\geqslant 1}(\mathtt{Y}, ([\![\mathtt{false}]\!] \wedge \mathtt{Y})\vee[\![y>c]\!])$
       $= [\![\mathsf{si}\vee\mathsf{di}\vee z\geqslant 6]\!] \wedge y.\mathsf{MaxU}_{\geqslant 1}([\![\mathtt{true}]\!], [\![y>c]\!])$
       $= [\![\mathsf{si}\vee\mathsf{di}\vee z\geqslant 6]\!]$

    $\mathtt{Y} := \mathtt{Z}$
    $\mathtt{Z} := [\![\mathsf{si}\vee\mathsf{di}\vee z\geqslant 6]\!] \wedge y.\mathsf{MaxU}_{\geqslant 1}(\mathtt{Y}, ([\![\mathtt{false}]\!] \wedge \mathtt{Y})\vee[\![y>c]\!])$
       $= [\![\mathsf{si}\vee\mathsf{di}\vee z\geqslant 6]\!] \wedge y.\mathsf{MaxU}_{\geqslant 1}(\mathtt{Y}, [\![y>c]\!])$
       $= [\![\mathsf{si}\vee\mathsf{di}\vee z\geqslant 6]\!] \wedge y.\{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3\wedge(z\geqslant x+3\vee y>x+c-3)),$
                                  $(\mathsf{di}, x\leqslant 2\wedge(z\geqslant x+4\vee y>x+c-2))\}$
       $= \{(\mathsf{sr}, z\geqslant 6), (\mathsf{si}, x\leqslant 3\wedge(z\geqslant x+3\vee x<3-c)), (\mathsf{di}, x\leqslant 2\wedge(z\geqslant x+4\vee x<2-c))\}$

    $\mathtt{Y} := \mathtt{Z}$
    $\mathtt{Z} := [\![\mathsf{si}\vee\mathsf{di}\vee z\geqslant 6]\!] \wedge y.\mathsf{MaxU}_{\geqslant 1}(\mathtt{Y}, ([\![\mathtt{false}]\!] \wedge \mathtt{Y})\vee[\![y>c]\!])$
       $= [\![\mathsf{si}\vee\mathsf{di}\vee z\geqslant 6]\!] \wedge y.\mathsf{MaxU}_{\geqslant 1}(\mathtt{Y}, [\![y>c]\!])$
       $= [\![\mathsf{si}\vee\mathsf{di}\vee z\geqslant 6]\!] \wedge y.\{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3\wedge(z\geqslant x+3\vee y>c\vee y>x+2\cdot c-3)),$
                                  $(\mathsf{di}, x\leqslant 2\wedge(z\geqslant x+4\vee y>c\vee y>x+2\cdot c-2))\}$
       $= \{(\mathsf{sr}, z\geqslant 6), (\mathsf{si}, x\leqslant 3\wedge(z\geqslant x+3\vee x<3-2\cdot c)), (\mathsf{di}, x\leqslant 2 \wedge (z\geqslant x+4\vee x<2-2\cdot c))\}$

    repeating $n-2$ times such that $n\cdot c \geqslant 3$ and $(n-1)\cdot c < 3$ (which exists as $c>0$)

    $\mathtt{Y} := \mathtt{Z}$
    $\mathtt{Z} := [\![\mathsf{si}\vee\mathsf{di}\vee z\geqslant 6]\!] \wedge y.\mathsf{MaxU}_{\geqslant 1}(\mathtt{Y}, ([\![\mathtt{false}]\!] \wedge \mathtt{Y})\vee[\![y>c]\!])$
       $= [\![\mathsf{si}\vee\mathsf{di}\vee z\geqslant 6]\!] \wedge y.\mathsf{MaxU}_{\geqslant 1}(\mathtt{Y}, [\![y>c]\!])$
       $= [\![\mathsf{si}\vee\mathsf{di}\vee z\geqslant 6]\!] \wedge y.\{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3\wedge(z\geqslant x+3\vee y>c\vee y>x+n\cdot c-3)),$
                                  $(\mathsf{di}, x\leqslant 2\wedge(z\geqslant x+4\vee y>c\vee y>x+n\cdot c-2))\}$
       $= \{(\mathsf{sr}, z\geqslant 6), (\mathsf{si}, x\leqslant 3\wedge(z\geqslant x+3\vee x<3-n\cdot c)), (\mathsf{di}, x\leqslant 2\wedge(z\geqslant x+4\vee x<2-n\cdot c))\}$
       $= \{(\mathsf{sr}, z\geqslant 6), (\mathsf{si}, x\leqslant 3\wedge z\geqslant x+3), (\mathsf{di}, x\leqslant 2\wedge z\geqslant x+4)\}$

endrepeat
```

C $\mathsf{MaxU}_{\geqslant 1}(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket, \llbracket y>c\rrbracket)$

```
$\mathtt{Z}_0 := \llbracket\mathtt{true}\rrbracket$
repeat
  $\mathtt{Y}_0 := \mathtt{Z}_0,\ \mathtt{Z}_1 := \llbracket\mathtt{false}\rrbracket$
  repeat
    $\mathtt{Y}_1 := \mathtt{Z}_1$
    $\mathtt{Z}_1 := \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \mathsf{pre1}(\mathtt{Y}_0,\mathtt{Y}_1)) = \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \llbracket\mathtt{false}\rrbracket)$
    $\mathtt{Z}_1 := \mathtt{Z}_1\vee\mathsf{tpre}_{\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket\vee\llbracket y>c\rrbracket}(\mathtt{Y}_0\wedge\mathtt{Y}_1) = \mathtt{Z}_1\vee\llbracket\mathtt{false}\rrbracket = \llbracket y>c\rrbracket$

    $\mathtt{Y}_1 := \mathtt{Z}_1$
    $\mathtt{Z}_1 := \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \mathsf{pre1}(\mathtt{Y}_0,\mathtt{Y}_1))$
        $= \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket\wedge\{(\mathsf{si}, 2\leqslant x\leqslant 3\wedge y>c), (\mathsf{di}, 1\leqslant x\leqslant 2\wedge y>c)\})$
    $\mathtt{Z}_1 := \mathtt{Z}_1\vee\mathsf{tpre}_{\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket\vee\llbracket y>c\rrbracket}(\mathtt{Y}_0\wedge\mathtt{Y}_1)$
        $= \mathtt{Z}_1\vee\{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3\wedge y>x+(c-3)), (\mathsf{di}, x\leqslant 2\wedge y>x+(c-2))\}$
        $= \{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3\wedge y>x+(c-3)), (\mathsf{di}, x\leqslant 2\wedge y>x+(c-2))\}$

    $\mathtt{Y}_1 := \mathtt{Z}_1$
    $\mathtt{Z}_1 := \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \mathsf{pre1}(\mathtt{Y}_0,\mathtt{Y}_1))$
        $= \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \{(\mathsf{si}, 2\leqslant x\leqslant 3 \wedge (y>c-3\vee z\geqslant 6)), (\mathsf{di}, 1\leqslant x\leqslant 2 \wedge (y>c-3\vee z\geqslant 6))\})$
    $\mathtt{Z}_1 := \mathtt{Z}_1\vee\mathsf{tpre}_{\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket\vee\llbracket y>c\rrbracket}(\mathtt{Y}_0\wedge\mathtt{Y}_1)$
        $= \mathtt{Z}_1\vee\{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3\wedge y>x+(c-3)), (\mathsf{di}, x\leqslant 2\wedge y>x+(c-2))\}$
        $= \{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, (2\leqslant x\leqslant 3 \wedge (y>c-3\vee z\geqslant 6))\vee(x\leqslant 3 \wedge y>x+(c-3)),$
            $(\mathsf{di}, (1\leqslant x\leqslant 2 \wedge (y>c-3\vee z\geqslant 6))\vee(x\leqslant 2 \wedge y>x+(c-2))\}$

    $\mathtt{Y}_1 := \mathtt{Z}_1$
    $\mathtt{Z}_1 := \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \mathsf{pre1}(\mathtt{Y}_0,\mathtt{Y}_1))$
        $= \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \{(\mathsf{si}, 2\leqslant x\leqslant 3 \wedge (y>c-3\vee z\geqslant 6)), (\mathsf{di}, 1\leqslant x\leqslant 2 \wedge (y>c-3\vee z\geqslant 6))\})$
    $\mathtt{Z}_1 := \mathtt{Z}_1\vee\mathsf{tpre}_{\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket\vee\llbracket y>c\rrbracket}(\mathtt{Y}_0\wedge\mathtt{Y}_1)$
        $= \mathtt{Z}_1\vee\{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3 \wedge (y>x+c-6\vee z\geqslant x+3), (\mathsf{di}, x\leqslant 2 \wedge (y>x+c-5\vee z\geqslant x+4))\}$
        $= \{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3 \wedge (y>x+c-6\vee z\geqslant x+3), (\mathsf{di}, x\leqslant 2 \wedge (y>x+c-5\vee z\geqslant x+4))\}$

    $\mathtt{Y}_1 := \mathtt{Z}_1$
    $\mathtt{Z}_1 := \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \mathsf{pre1}(\mathtt{Y}_0,\mathtt{Y}_1))$
        $= \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \{(\mathsf{si}, 2\leqslant x\leqslant 3 \wedge (y>c-6\vee z\geqslant 3)), (\mathsf{di}, 1\leqslant x\leqslant 2 \wedge (y>c-6\vee z\geqslant 3))\})$
    $\mathtt{Z}_1 := \mathtt{Z}_1\vee\mathsf{tpre}_{\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket\vee\llbracket y>c\rrbracket}(\mathtt{Y}_0\wedge\mathtt{Y}_1)$
        $= \mathtt{Z}_1\vee\{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3 \wedge (y>x+c-6\vee z\geqslant x+3), (\mathsf{di}, x\leqslant 2 \wedge (y>x+c-5\vee z\geqslant x+4))\}$
        $= \{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, (2\leqslant x\leqslant 3 \wedge (y>c-6\vee z\geqslant 3)))\vee(x\leqslant 3 \wedge (y>x+c-6\vee z\geqslant x+3))),$
            $(\mathsf{di}, (1\leqslant x\leqslant 2 \wedge (y>c-6\vee z\geqslant 3))\vee(x\leqslant 2 \wedge (y>x+c-5\vee z\geqslant x+4)))\}$

    $\mathtt{Y}_1 := \mathtt{Z}_1$
    $\mathtt{Z}_1 := \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \mathsf{pre1}(\mathtt{Y}_0,\mathtt{Y}_1))$
        $= \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \{(\mathsf{si}, 2\leqslant x\leqslant 3 \wedge (y>c-6\vee z\geqslant 3)), (\mathsf{di}, 1\leqslant x\leqslant 2 \wedge (y>c-6\vee z\geqslant 3))\})$
    $\mathtt{Z}_1 := \mathtt{Z}_1\vee\mathsf{tpre}_{\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket\vee\llbracket y>c\rrbracket}(\mathtt{Y}_0\wedge\mathtt{Y}_1)$
        $= \mathtt{Z}_1\vee\{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3 \wedge (y>x+c-9\vee z\geqslant x), (\mathsf{di}, x\leqslant 2 \wedge (y>x+c-8\vee z\geqslant x+1))\}$
        $= \{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3 \wedge (y>x+c-9\vee z\geqslant x), (\mathsf{di}, x\leqslant 2 \wedge (y>x+c-8\vee z\geqslant x+1))\}$

    $\mathtt{Y}_1 := \mathtt{Z}_1$
    $\mathtt{Z}_1 := \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \mathsf{pre1}(\mathtt{Y}_0,\mathtt{Y}_1))$
        $= \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \{(\mathsf{si}, 2\leqslant x\leqslant 3), (\mathsf{di}, 1\leqslant x\leqslant 2)\})$
    $\mathtt{Z}_1 := \mathtt{Z}_1\vee\mathsf{tpre}_{\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket\vee\llbracket y>c\rrbracket}(\mathtt{Y}_0\wedge\mathtt{Y}_1)$
        $= \mathtt{Z}_1\vee\{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3 \wedge (y>x+c-9\vee z\geqslant x), (\mathsf{di}, x\leqslant 2 \wedge (y>x+c-8\vee z\geqslant x+1))\}$
        $= \{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, (2\leqslant x\leqslant 3)\vee(x\leqslant 3 \wedge (y>x+c-9\vee z\geqslant x))),$
            $(\mathsf{di}, (1\leqslant x\leqslant 2)\vee(x\leqslant 2 \wedge (y>x+c-8\vee z\geqslant x+1)))\}$

    $\mathtt{Y}_1 := \mathtt{Z}_1$
    $\mathtt{Z}_1 := \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \mathsf{pre1}(\mathtt{Y}_0,\mathtt{Y}_1))$
        $= \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \{(\mathsf{si}, 2\leqslant x\leqslant 3), (\mathsf{di}, 1\leqslant x\leqslant 2)\})$
    $\mathtt{Z}_1 := \mathtt{Z}_1\vee\mathsf{tpre}_{\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket\vee\llbracket y>c\rrbracket}(\mathtt{Y}_0\wedge\mathtt{Y}_1)$
        $= \mathtt{Z}_1\vee\{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3), (\mathsf{di}, x\leqslant 2)\}$
        $= \{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3), (\mathsf{di}, x\leqslant 2)\}$

    $\mathtt{Y}_1 := \mathtt{Z}_1$
    $\mathtt{Z}_1 := \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \mathsf{pre1}(\mathtt{Y}_0,\mathtt{Y}_1))$
        $= \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \{(\mathsf{si}, 2\leqslant x\leqslant 3), (\mathsf{di}, 1\leqslant x\leqslant 2)\})$
    $\mathtt{Z}_1 := \mathtt{Z}_1\vee\mathsf{tpre}_{\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket\vee\llbracket y>c\rrbracket}(\mathtt{Y}_0\wedge\mathtt{Y}_1)$
        $= \mathtt{Z}_1\vee\{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3), (\mathsf{di}, x\leqslant 2)\}$
        $= \{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3), (\mathsf{di}, x\leqslant 2)\}$
  endrepeat
```

```
$Z_0 := \{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3), (\mathsf{di}, x\leqslant 2)\}$
  $Y_0 := Z_0,\ Z_1 := \llbracket\mathtt{false}\rrbracket$
  repeat
    $Y_1 := Z_1$
    $Z_1 := \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \mathtt{pre1}(Y_0, Y_1)) = \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \llbracket\mathtt{false}\rrbracket)$
    $Z_1 := Z_1\vee\mathsf{tpre}_{\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket\vee\llbracket y>c\rrbracket}(Y_0\wedge Y_1) = Z_1\vee\llbracket\mathtt{false}\rrbracket = \llbracket y>c\rrbracket$

    $Y_1 := Z_1$
    $Z_1 := \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \mathtt{pre1}(Y_0, Y_1))$
        $= \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \{(\mathsf{si}, 2\leqslant x\leqslant 3\wedge y>c)\vee(\mathsf{di}, 1\leqslant x\leqslant 2\wedge y>c)\}$
    $Z_1 := Z_1\vee\mathsf{tpre}_{\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket\vee\llbracket y>c\rrbracket}(Y_0\wedge Y_1)$
        $= Z_1\vee\{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3\wedge y>x+c-3), (\mathsf{di}, x\leqslant 2\wedge y>x+c-2)\}$
        $= \{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3\wedge y>x+c-3), (\mathsf{di}, x\leqslant 2\wedge y>x+c-2)\}$

    $Y_1 := Z_1$
    $Z_1 := \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \mathtt{pre1}(Y_0, Y_1))$
        $= \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \{(\mathsf{si}, 2\leqslant x\leqslant 3\wedge(z\geqslant 6\vee y>c))\vee(\mathsf{di}, 1\leqslant x\leqslant 2\wedge(z\geqslant 6\vee y>c))\}$
    $Z_1 := Z_1\vee\mathsf{tpre}_{\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket\vee\llbracket y>c\rrbracket}(Y_0\wedge Y_1)$
        $= Z_1\vee\{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3\wedge y>x+c-3), (\mathsf{di}, x\leqslant 2\wedge y>x+c-2)\}$
        $= \{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, (2\leqslant x\leqslant 3\wedge z\geqslant 6)\vee(x\leqslant 3\wedge y>x+c-3)),$
            $(\mathsf{di}, (1\leqslant x\leqslant 2\wedge z\geqslant 6)\vee(x\leqslant 2\wedge y>x+c-2))\}$

    $Y_1 := Z_1$
    $Z_1 := \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \mathtt{pre1}(Y_0, Y_1))$
        $= \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \{(\mathsf{si}, 2\leqslant x\leqslant 3\wedge(z\geqslant 6\vee y>c))\vee(\mathsf{di}, 1\leqslant x\leqslant 2\wedge(z\geqslant 6\vee y>c))\}$
    $Z_1 := Z_1\vee\mathsf{tpre}_{\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket\vee\llbracket y>c\rrbracket}(Y_0\wedge Y_1)$
        $= Z_1\vee\{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3\wedge(z\geqslant x+3\vee y>x+c-3)), (\mathsf{di}, x\leqslant 2\wedge(z\geqslant x+4\vee y>x+c-2))\}$
        $= \{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3\wedge(z\geqslant x+3\vee y>x+c-3)), (\mathsf{di}, x\leqslant 2 \wedge (z\geqslant x+4\vee y>x+c-2))\}$

    $Y_1 := Z_1$
    $Z_1 := \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \mathtt{pre1}(Y_0, Y_1))$
        $= \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \{(\mathsf{si}, 2\leqslant x\leqslant 3\wedge(z\geqslant 6\vee y>c))\vee(\mathsf{di}, 1\leqslant x\leqslant 2\wedge(z\geqslant 6\vee y>c))\}$
    $Z_1 := Z_1\vee\mathsf{tpre}_{\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket\vee\llbracket y>c\rrbracket}(Y_0\wedge Y_1)$
        $= Z_1\vee\{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3\wedge(z\geqslant x+3\vee y>x+c-3)), (\mathsf{di}, x\leqslant 2\wedge(z\geqslant x+4\vee y>x+c-2))\}$
        $= \{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3\wedge(z\geqslant x+3\vee y>x+c-3)), (\mathsf{di}, x\leqslant 2 \wedge (z\geqslant x+4\vee y>x+c-2))\}$
  end repeat
$Z_0 := \{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3\wedge(z\geqslant x+3\vee y>x+c-3)), (\mathsf{di}, x\leqslant 2 \wedge (z\geqslant x+4\vee y>x+c-2))\}$
  $Y_0 := Z_0,\ Z_1 := \llbracket\mathtt{false}\rrbracket$
  repeat
    $Y_1 := Z_1$
    $Z_1 := \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \mathtt{pre1}(Y_0, Y_1)) = \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \llbracket\mathtt{false}\rrbracket)$
    $Z_1 := Z_1\vee\mathsf{tpre}_{\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket\vee\llbracket y>c\rrbracket}(Y_0\wedge Y_1) = Z_1\vee\llbracket\mathtt{false}\rrbracket = \llbracket y>c\rrbracket$

    $Y_1 := Z_1$
    $Z_1 := \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \mathtt{pre1}(Y_0, Y_1))$
        $= \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \{(\mathsf{si}, 2\leqslant x\leqslant 3\wedge y>c)\vee(\mathsf{di}, 1\leqslant x\leqslant 2\wedge y>c)\}$
    $Z_1 := Z_1\vee\mathsf{tpre}_{\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket\vee\llbracket y>c\rrbracket}(Y_0\wedge Y_1)$
        $= Z_1\vee\{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3\wedge y>x+c-3), (\mathsf{di}, x\leqslant 2\wedge y>x+c-2)\}$
        $= \{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3\wedge y>x+c-3), (\mathsf{di}, x\leqslant 2\wedge y>x+c-2)\}$

    $Y_1 := Z_1$
    $Z_1 := \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \mathtt{pre1}(Y_0, Y_1))$
        $= \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \{(\mathsf{si}, 2\leqslant x\leqslant 3\wedge(z\geqslant 6\vee y>c))\vee(\mathsf{di}, 1\leqslant x\leqslant 2\wedge(z\geqslant 6\vee y>c))\}$
    $Z_1 := Z_1\vee\mathsf{tpre}_{\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket\vee\llbracket y>c\rrbracket}(Y_0\wedge Y_1)$
        $= Z_1\vee\{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3\wedge y>x+c-3), (\mathsf{di}, x\leqslant 2\wedge y>x+c-2)\}$
        $= \{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, (2\leqslant x\leqslant 3\wedge z\geqslant 6)\vee(x\leqslant 3\wedge y>x+c-3)),$
            $(\mathsf{di}, (1\leqslant x\leqslant 2\wedge z\geqslant 6)\vee(x\leqslant 2\wedge y>x+c-2))\}$

    $Y_1 := Z_1$
    $Z_1 := \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \mathtt{pre1}(Y_0, Y_1))$
        $= \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \{(\mathsf{si}, 2\leqslant x\leqslant 3\wedge(z\geqslant 6\vee y>c))\vee(\mathsf{di}, 1\leqslant x\leqslant 2\wedge(z\geqslant 6\vee y>c))\}$
    $Z_1 := Z_1\vee\mathsf{tpre}_{\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket\vee\llbracket y>c\rrbracket}(Y_0\wedge Y_1)$
        $= Z_1\vee\{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3\wedge(z\geqslant x+3\vee y>x+c-3)), (\mathsf{di}, x\leqslant 2\wedge(z\geqslant x+4\vee y>x+c-2))\}$
        $= \{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3\wedge(z\geqslant x+3\vee y>x+c-3)), (\mathsf{di}, x\leqslant 2 \wedge (z\geqslant x+4\vee y>x+c-2))\}$

    $Y_1 := Z_1$
    $Z_1 := \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \mathtt{pre1}(Y_0, Y_1))$
        $= \llbracket y>c\rrbracket\vee(\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket \wedge \{(\mathsf{si}, 2\leqslant x\leqslant 3\wedge(z\geqslant 6\vee y>c))\vee(\mathsf{di}, 1\leqslant x\leqslant 2\wedge(z\geqslant 6\vee y>c))\}$
    $Z_1 := Z_1\vee\mathsf{tpre}_{\llbracket \mathsf{si}\vee\mathsf{di}\vee z\geqslant 6\rrbracket\vee\llbracket y>c\rrbracket}(Y_0\wedge Y_1)$
        $= Z_1\vee\{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3\wedge(z\geqslant x+3\vee y>x+c-3)), (\mathsf{di}, x\leqslant 2\wedge(z\geqslant x+4\vee y>x+c-2))\}$
        $= \{(\mathsf{sr}, z\geqslant 6\vee y>c), (\mathsf{si}, x\leqslant 3\wedge(z\geqslant x+3\vee y>x+c-3)), (\mathsf{di}, x\leqslant 2 \wedge (z\geqslant x+4\vee y>x+c-2))\}$
  end repeat
$Z_0 := Z_1$

end repeat
```

D $\mathsf{MaxU}([\![\mathsf{si}\vee z\geqslant 6]\!], \{(\mathsf{sr}, z\geqslant 6), (\mathsf{si}, x\leqslant 3 \wedge z\geqslant x+3), (\mathsf{di}, x\leqslant 2 \wedge z\geqslant x+4)\}, \gtrsim \lambda)$

```
$\mathtt{Z} := \{(\mathsf{sr}, z\geqslant 6), (\mathsf{si}, x\leqslant 3 \wedge z\geqslant x+3), (\mathsf{di}, x\leqslant 2 \wedge z\geqslant x+4)\}$
repeat
  $\mathtt{Y} := \mathtt{Z}$
  begin for
    $\mathsf{z} = (\mathsf{sr}, z\geqslant 6)$ [two edges (from si and di) taking predecessors]
      $\mathsf{y}_1 = (\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z\geqslant 6)$ [subset of target set]
      $\mathsf{y}_2 = (\mathsf{di}, 1\leqslant x\leqslant 2 \wedge z\geqslant 6)$ [subset of target set]
    $\mathsf{z} = (\mathsf{si}, x\leqslant 3 \wedge z\geqslant x+3)$ [two edges (from si and di) taking predecessors]
      $\mathsf{y}_3 = (\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z\geqslant 3)$
      $E_{\mathsf{si},g,p} = E_{\mathsf{si},g,p} \cup \{((\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z\geqslant 3), (\mathsf{si}, \{x\}), (\mathsf{si}, x\leqslant 3 \wedge z\geqslant x+3))\}$
      $\mathsf{y}_4 = (\mathsf{di}, 1\leqslant x\leqslant 2 \wedge z\geqslant 3)$
      $E_{\mathsf{di},g,p} = E_{\mathsf{di},g,p} \cup \{((\mathsf{di}, 1\leqslant x\leqslant 2 \wedge z\geqslant 3), (\mathsf{si}, \{x\}), (\mathsf{si}, x\leqslant 3 \wedge z\geqslant x+3))\}$
    $\mathsf{z} = (\mathsf{di}, x\leqslant 3 \wedge z\geqslant x+4)$
      [no edges]
  end for
  $\mathtt{Z} := \{(\mathsf{sr}, z\geqslant 6), (\mathsf{si}, x\leqslant 3 \wedge z\geqslant x+3), (\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z\geqslant 3), (\mathsf{di}, x\leqslant 2 \wedge z\geqslant x+4), (\mathsf{di}, 1\leqslant x\leqslant 2 \wedge z\geqslant 3)\}$
  $\mathtt{Y} := \mathtt{Z}$
  begin for
    $\mathsf{z} = (\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z\geqslant 3)$ [two edges (from si and di) taking predecessors]
      $\mathsf{y}_1 = (\mathsf{si}, 2\leqslant x\leqslant 3)$
      $E_{\mathsf{si},g,p} = E_{\mathsf{si},g,p} \cup \{((\mathsf{si}, 2\leqslant x\leqslant 3), (\mathsf{si}, \{x\}), (\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z\geqslant 3))\}$
      $\mathsf{y}_2 = (\mathsf{di}, 1\leqslant x\leqslant 2)$
      $E_{\mathsf{di},g,p} = E_{\mathsf{di},g,p} \cup \{((\mathsf{di}, 1\leqslant x\leqslant 2), (\mathsf{si}, \{x\}), (\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z\geqslant 3))\}$
    $\mathsf{z} = (\mathsf{di}, 1\leqslant x\leqslant 2 \wedge z\geqslant 3)$ [no edges]
  end for
  $\mathtt{Z} := \{(\mathsf{sr}, z\geqslant 6), (\mathsf{si}, x\leqslant 3 \wedge z\geqslant x+3), (\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z\geqslant 3), (\mathsf{si}, 2\leqslant x\leqslant 3),$
        $(\mathsf{di}, x\leqslant 2 \wedge z\geqslant x+4), (\mathsf{di}, 1\leqslant x\leqslant 2 \wedge z\geqslant 3), (\mathsf{di}, 1\leqslant x\leqslant 2)\}$
  $\mathtt{Y} := \mathtt{Z}$
  begin for
    $\mathsf{z} = (\mathsf{si}, 2\leqslant x\leqslant 3)$
      [two edges (from si and di) taking predecessors]
      $\mathsf{y}_1 = (\mathsf{si}, 2\leqslant x\leqslant 3)$
      $E_{\mathsf{si},g,p} = E_{\mathsf{si},g,p} \cup \{((\mathsf{si}, 2\leqslant x\leqslant 3), (\mathsf{si}, \{x\}), (\mathsf{si}, 2\leqslant x\leqslant 3))\}$
      $\mathsf{y}_2 = (\mathsf{di}, 1\leqslant x\leqslant 2)$
      $E_{\mathsf{di},g,p} = E_{\mathsf{di},g,p} \cup \{((\mathsf{di}, 1\leqslant x\leqslant 2), (\mathsf{si}, \{x\}), (\mathsf{si}, 2\leqslant x\leqslant 3))\}$
    $\mathsf{z} = (\mathsf{di}, 1\leqslant x\leqslant 2)$ [no edges]
  end for
  $\mathtt{Z} := \{(\mathsf{sr}, z\geqslant 6), (\mathsf{si}, x\leqslant 3 \wedge z\geqslant x+3), (\mathsf{si}, 2\leqslant x\leqslant 3 \wedge z\geqslant 3), (\mathsf{si}, 2\leqslant x\leqslant 3),$
        $(\mathsf{di}, x\leqslant 2 \wedge z\geqslant x+4), (\mathsf{di}, 1\leqslant x\leqslant 2 \wedge z\geqslant 3), (\mathsf{di}, 1\leqslant x\leqslant 2)\}$
end repeat
```