Automatic Verification of Real-time Systems with Discrete Probability Distributions

Marta Kwiatkowska[1⋆], Gethin Norman[1⋆],
Roberto Segala[2⋆⋆], and Jeremy Sproston[1]

[1] University of Birmingham, Birmingham B15 2TT, United Kingdom
{M.Z.Kwiatkowska,G.Norman,J.Sproston}@cs.bham.ac.uk
[2] Università di Bologna, Mura Anteo Zamboni 7, 40127 Bologna, Italy
segala@cs.unibo.it

Abstract. We consider the timed automata model of [3], which allows the analysis of real-time systems expressed in terms of quantitative timing constraints. Traditional approaches to real-time system description express the model purely in terms of nondeterminism; however, we may wish to express the likelihood of the system making certain transitions. In this paper, we present a model for real-time systems augmented with discrete probability distributions. Furthermore, using the algorithm of [5] with fairness, we develop a model checking method for such models against temporal logic properties which can refer both to timing properties and probabilities, such as, "with probability 0.6 or greater, the clock x remains below 5 until clock y exceeds 2".

1 Introduction

The proliferation of digital technology embedded into real-life environments has led to increased interest in computer systems expressed in terms of quantitative timing constraints. Examples of such *real-time systems* include communication protocols, digital circuits with uncertain delay lengths, and media synchronization protocols. A number of frameworks exist within which the formal reasoning and analysis of such systems can be carried out. A formalism that has received much attention, both in terms of theoretical and practical developments, is that of *timed automata*; in particular, the theory of automatically verifying timed automata against properties of a real-time temporal logic is advanced, and is supported by a number of tools [6, 8].

Traditional approaches to the formal description of real-time systems express the system model purely in terms of nondeterminism. However, it may be desirable to express the relative likelihood of the system exhibiting certain behaviour. For example, we may wish to model a system for which the likelihood of a certain event occurring changes with respect to the amount of time elapsed. This notion is particularly important when considering fault-tolerant systems. Furthermore,

⋆ supported in part by EPSRC grant GR/M04617
⋆⋆ supported in part by EPSRC grant GR/M13046

we may also wish to refer to the likelihood of certain temporal logic properties being satisfied by the real-time system, and to have a model checking algorithm for verifying the truth of these assertions. The remit of this paper is to address these problems.

Therefore, we present a model for real-time systems that are described partially in terms of discrete probability distributions, and an automatic verification method for this model against a new, probabilistic real-time logic. The system model is called a *probabilistic timed graph*, and differs from the timed automata based model of [2] in the following respects. Firstly, the edge relation of probabilistic timed graphs is both nondeterministic and probabilistic in nature. More precisely, instead of making a purely nondeterministic choice over the set of currently enabled edges, we choose amongst the set of enabled discrete probability distributions, each of which is defined over a finite set of edges. We then make a probabilistic choice as to which edge to take according to the selected distribution. As with usual timed automata techniques, the underlying model of time is assumed to be *dense*; that is, the time domain is modelled by the reals ($\mathbb{R}$) or rationals ($\mathbb{Q}$). However, in contrast to [2], probabilistic timed graphs are defined over *weakly monotonic time*, which allows us to express the notion of more than one system event occurring at a given point in time.

Furthermore, we adapt the specification language commonly used for stating real-time system requirements, TCTL (Timed Computation Tree Logic) [14], to cater for probability. A common approach taken in probabilistic temporal logics is to augment certain formulae with a parameter referring to a bound on probability which must be satisfied for the formula to be true. For example, $[\phi_1 \exists \mathcal{U} \phi_2]_{\geq p}$ is true if the probability of $[\phi_1 \exists \mathcal{U} \phi_2]$ is at least p. Therefore, we develop our specification language, PTCTL (Probabilistic Timed Computation Tree Logic), by adding such probabilistic operators to TCTL. The resulting logic allows us to express such quality of service properties as, "with probability 0.7, there will be a response between 5 and 7 time units after a query".

The denseness of the time domain means that the state space of timed automata is infinite. Therefore, automatic verification of timed automata is performed by constructing a finite-state quotient of the system model. This quotient takes the form of a state-labelled transition system which represents all of the timed automaton's behaviours, and which can be analyzed using analogues of traditional model checking techniques. We adopt this method in order to construct a finite quotient of probabilistic timed graphs; naturally, the transitions of the resulting model are both nondeterministic and probabilistic in nature, and therefore the model checking methods employed must accommodate this characteristic. The verification algorithms of [5] are used for this purpose. However, they are defined with respect to PBTL (Probabilistic Branching Time Logic), which does not allow the expression of dense timing constraints. Hence, we present a method for translating a given PTCTL formula into a corresponding PBTL formula. The model checking algorithm of [5] is then used to verify the PBTL properties over our probabilistic-nondeterministic quotient structure, the results of which allow us to conclude whether the original probabilistic timed graph satisfied its PTCTL specification. Furthermore, the verification methods of [5] allow us to model check *fair* paths of the quotient construction. In the context of real-time systems, fair paths correspond to behaviours which allow the progress of time, a notion which also corresponds to realisable behaviours.

An example of a real-time system which could be subject to these techniques is the bounded retransmission protocol, which is modelled as a network of purely nondeterministic timed automata in [9]. Each communication channel is represented as a timed automaton which features a nondeterministic choice over two edges, one of which corresponds to the correct transmission of the message, the other to the message's loss. Using our framework, the relative likelihood of such a loss occurring could be represented by replacing this nondeterministic choice by a probabilistic choice between the two edges; for example, a probabilistic timed graph could be used to model that a message is lost with probability 0.05 each time a communication channel is used. Similarly, the system requirements of the bounded retransmission protocol could be expanded to admit reasoning about the probability of certain system behaviours. For instance, we may require that, with probability at least 0.99, any data chunk transmitted by the sender is successfully processed by the receiver within 10 time units.

The model presented in this paper has similarities with other frameworks for probabilistic real-time systems. In particular, the approach of [10] is also to augment timed automata with discrete probability distributions; however, these distributions are obtained by normalization of edge-labelling weights. Furthermore, the model checking algorithm of [10] is with respect to an action-based logic, rather than a state-based logic such as PTCTL. A dense time, automata-based model with discrete and *continuous* probability distributions is presented in [1], along with a quotient construction and TCTL model checking method similar to that of [2]. However, the model of Alur et al. does not permit any nondeterministic choice, and its use of continuous probability distributions, while a highly expressive modelling mechanism, does not permit the model to be automatically verified against logics which include bounds on probability. Furthermore, note that the temporal logic of [11] has syntactic similarities with the logic PTCTL, although this former logic is interpreted with respect to discrete, not dense time.

The paper proceeds as follows. Section 2 introduces some preliminary concepts and notation relating to execution sequences. Section 3 presents the underlying model of our probabilistic timed graphs, which are used to interpret formulae of the logic, PTCTL, introduced in section 4. Probabilistic timed graphs are defined in section 5 as our model for probabilistic-nondeterministic real-time systems, and a method for translating them into their underlying probabilistic timed structure is presented. Section 6 explores the model checking problem for probabilistic timed graphs, and presents a finite-state quotient construction for this model, a method for translating a PTCTL formula into a series of equivalent PBTL formulae, and finally a verification method. To conclude, section 7 analyzes the complexity of the model checking technique, and suggests further directions of research.

2 Preliminaries

Labelled paths (or execution sequences) are non-empty finite or infinite sequences of the form:

$$\omega = \sigma_0 \xrightarrow{l_0} \sigma_1 \xrightarrow{l_1} \sigma_2 \xrightarrow{l_2} \cdots$$

where σ_i are states and l_i are labels for transitions. We use the following notation for such paths. Take any path ω. Then the first state of ω is denoted by $\mathit{first}(\omega)$. If ω is finite then the last state of ω is denoted by $\mathit{last}(\omega)$. The length of a path, $|\omega|$, is defined in the usual way: if ω is the finite path $\omega = \sigma_0 \xrightarrow{l_0} \sigma_1 \xrightarrow{l_1} \cdots \xrightarrow{l_{n-1}} \sigma_n$, then $|\omega| = n$; if ω is an infinite path, then we let $|\omega| = \infty$. If $k \leq |\omega|$ then $\omega(k)$ denotes the k-th state of ω and $\mathit{step}(\omega, k)$ is the label of the k-th step (that is, $\omega(k) = \sigma_k$ and $\mathit{step}(\omega, k) = l_k$). $\omega^{(k)}$ is the k-th prefix of ω; that is, if $k < |\omega|$ then $\omega^{(k)} = \sigma_0 \xrightarrow{l_0} \sigma_1 \xrightarrow{l_1} \cdots \xrightarrow{l_{k-1}} \sigma_k$, and if $k \geq |\omega|$ then $\omega^{(k)} = \omega$. If $\omega = \sigma_0 \xrightarrow{l_0} \sigma_1 \xrightarrow{l_1} \cdots \xrightarrow{l_{n-1}} \sigma_n$ is a finite path and $\omega' = \sigma'_0 \xrightarrow{l'_0} \sigma'_1 \xrightarrow{l'_1} \cdots$ is a finite or infinite path with $\mathit{last}(\omega) = \mathit{first}(\omega')$, then we let the *concatenation* of ω and ω' be:

$$\omega\omega' = \sigma_0 \xrightarrow{l_0} \sigma_1 \xrightarrow{l_1} \sigma_2 \cdots \xrightarrow{l_{n-1}} \sigma_n \xrightarrow{l'_0} \sigma'_1 \xrightarrow{l'_1} \cdots$$

3 Probabilistic Timed Structures

In this section, we introduce an underlying model for probabilistic timed graphs, called *probabilistic timed structures*, which are obtained by augmenting the timed structures of [13] with a probabilistic choice over transitions. More precisely, instead of a nondeterministic choice over transitions that consist of a real-valued duration and a next state, as is the case in traditional timed structures, the transition function of probabilistic timed structures results in a choice over pairs consisting of a duration and a *discrete probability distribution* over next states.

Let AP be a set of atomic propositions. A *clock* x is a real-valued variable which increases at the same rate as real-time. Let $\mathcal{X}$ be a set of clocks, and let $\nu : \mathcal{X} \to \mathbb{R}$ be a function assigning a real value to each of the clocks in this set. Such a function is called a *clock valuation*. For some $C \subseteq \mathcal{X}$ we write $\nu[C \mapsto 0]$ for the clock valuation that assigns 0 to all clocks in C, and agrees with ν for all clocks in $\mathcal{X} \setminus C$ (informally, we write $\nu[x \mapsto 0]$ if C contains the single clock x). In addition, for some $t \in \mathbb{R}$, $\nu + t$ denotes the clock valuation for which all clocks x in $\mathcal{X}$ take the value $\nu(x) + t$.

Definition 1 (State). *A state σ is an interpretation of all propositions and a valuation over the set of clocks: σ assigns to each proposition a in* AP *a boolean value (therefore, $\sigma(a) \in \{\texttt{true}, \texttt{false}\}$) and to each clock in $\mathcal{X}$ a non-negative real (therefore, $\sigma(x) \in \mathbb{R}$).*

We denote the set of discrete probability distributions over a set S by $\mu(S)$. Therefore, each $p \in \mu(S)$ is a function $p : S \to [0, 1]$ such that $\sum_{s \in S} p(s) = 1$.

Definition 2 (Probabilistic Timed Structure). *A* probabilistic timed structure $\mathcal{M}$*, is a tuple* (Σ, Tr, End) *where* Σ *is a set of states,* Tr *is a function which assigns to each state* $\sigma \in \Sigma$ *a set* $Tr(\sigma)$ *of pairs of the form* (t, p) *where* $t \in \mathbb{R}$ *and* $p \in \mu(\Sigma)$*, and* End *is a set of states from which time is allowed to increase without bound.*

$Tr(\sigma)$ is the set of transitions that can be nondeterministically chosen in state σ. Each transition takes the form (t, p), where t represents the duration of the transition and p is the probability distribution used over the set of successor states. Therefore, given the nondeterministic choice of $(t, p) \in Tr(\sigma)$ in state σ, then, after t time units have elapsed, a probabilistic transition is made to state σ' with probability $p(\sigma')$.

Paths in a probabilistic timed structure arise by resolving both the nondeterministic and probabilistic choices. A *path* of the probabilistic timed structure $\mathcal{M} = (\Sigma, Tr, End)$ is a non-empty finite or infinite sequence:

$$\omega = \sigma_0 \xrightarrow{t_0, p_0} \sigma_1 \xrightarrow{t_1, p_1} \sigma_2 \xrightarrow{t_2, p_2} \cdots$$

where $\sigma_i \in \Sigma$, $(t_i, p_i) \in Tr(\sigma_i)$ and $p_i(\sigma_{i+1}) > 0$ for all $0 \leq i \leq |\omega|$.

Sets of labelled paths are denoted in the following way. $Path_{fin}$ is the set of finite paths, and $Path_{fin}(\sigma)$ is the set of paths in $Path_{fin}$ such that $\omega(0) = \sigma$. $Path_{ful}$ is the set of paths such that $\omega \in Path_{ful}$ if either ω is infinite, or ω is finite and $last(\omega) \in End$. $Path_{ful}(\sigma)$ is the set of paths in $Path_{ful}$ such that $\omega(0) = \sigma$.

Consider an infinite path ω of $\mathcal{M}$. A *position* of ω is a pair (i, t'), where $i \in \mathbb{N}$ and $t' \in \mathbb{R}$ such that $0 \leq t' \leq t_i$. The *state at position* (i, t'), denoted by $\sigma_i + t'$, assigns $\sigma_i(a)$ to each proposition a in AP, and $\sigma_i(x) + t'$ to each clock x in $\mathcal{X}$. Given a path ω, $i, j \in \mathbb{N}$ and $t, t' \in \mathbb{R}$ such that $i \leq |\omega|$, $t \leq t_i$ and $t' \leq t_j$, then we say that the position (j, t') *precedes* the position (i, t), written $(j, t') \prec (i, t)$, iff $j < i$, or $j = i$ and $t' < t$.

Definition 3 (Duration of a Path). *For any path* ω *of a probabilistic timed structure* $\mathcal{M}$ *and* $0 \leq i \leq |\omega|$ *we define* $\mathcal{D}_\omega(i)$*, the* elapsed time *until the* i*th transition, as follows:* $\mathcal{D}_\omega(0) = 0$ *and for any* $1 \leq i \leq |\omega|$*:*

$$\mathcal{D}_\omega(i) = \sum_{j=0}^{i-1} t_j.$$

We now introduce *adversaries* of probabilistic timed structures as functions which resolve all the nondeterministic choices of the model.

Definition 4 (Adversary of a Probabilistic Timed Structure). *An* adversary *(or scheduler) of a probabilistic timed structure* $\mathcal{M} = (\Sigma, Tr, End)$ *is a function* A *mapping every finite path* ω *of* $\mathcal{M}$ *to a pair* (t, p) *such that* $A(\omega) \in Tr(last(\omega))$*. Let* $\mathcal{A}$ *be the set of all adversaries of* $\mathcal{M}$*.*

For an adversary A of a probabilistic timed structure $\mathcal{M} = (\Sigma, Tr, End)$ we define $Path^A_{fin}$ to be the set of finite paths such that $step(\omega, i) = A(\omega^{(i)})$ for all $1 \leq i \leq |\omega|$, and $Path^A_{ful}$ to be the set of paths in $Path_{ful}$ such that $step(\omega, i) = A(\omega^{(i)})$ for all $i \in \mathbb{N}$.

With each adversary we associate a sequential Markov chain, which can be viewed as a set of paths in $\mathcal{M}$. Formally, if A is an adversary of the probabilistic timed structure $\mathcal{M}$, then $MC^A = (Path^A_{fin}, \mathbf{P}^A)$ is a Markov chain where:

$$\mathbf{P}^A(\omega, \omega') = \begin{cases} p(\sigma) \text{ if } A(\omega) = (t, p) \text{ and } \omega' = \omega \xrightarrow{t,p} \sigma \\ 0 \quad\;\; \text{otherwise} \end{cases}$$

Definition 5 (Divergent Adversary). *An adversary A of a probabilistic timed structure (Σ, Tr, End) is* divergent *if and only if for any infinite path $\omega \in Path^A_{ful}$ and $t \in \mathbb{R}$, there exists $j \in \mathbb{N}$ such that $\mathcal{D}_\omega(j) > t$. Let $\mathcal{A}_{div}$ be the set of all divergent adversaries.*

Note that this definition of divergent adversaries corresponds to a common restriction imposed in the study of real-time systems, namely that of *time-divergence*. The traditional interpretation of this requirement is that runs of the real-time system that are not time-divergent can be disregarded during analysis, because they do not represent realisable behaviour; in our case, consideration of the class of divergent adversaries means that nondeterministic choice is resolved in such a way as to result only in time-divergent paths.

For any probabilistic timed structure, let $\mathcal{F}_{Path}$ be the smallest σ-algebra on $Path_{ful}$ which contains the sets:

$$\{\omega \mid \omega \in Path_{ful} \text{ and } \omega' \text{ is a prefix of } \omega\}$$

for all $\omega' \in Path_{fin}$.

We now define a measure $Prob$ on the σ-algebra $\mathcal{F}_{Path}$, by first defining the following function on the set of finite paths $Path_{fin}$.

Definition 6. *Let $Prob_{fin} : Path_{fin} \to [0, 1]$ be the mapping inductively defined on the length of paths in $Path_{fin}$ as follows. If $|\omega| = 0$, then $Prob_{fin}(\omega) = 1$.*

Now consider any path ω such that $|\omega| = n + 1$. If $\omega^{(n)} = \omega'$ let:

$$Prob_{fin}(\omega) = Prob_{fin}(\omega') \cdot \mathbf{P}^A(\omega', \omega)$$

where A is any adversary such that $A(\omega') = (t, p)$ and $\omega = \omega' \xrightarrow{t,p} \sigma$.

Definition 7. *The measure $Prob$ on $\mathcal{F}_{Path}$ is the unique measure such that:*

$$Prob(\{\omega \mid \omega \in Path_{ful} \textit{ and } \omega' \textit{ is a prefix of } \omega\}) = Prob_{fin}(\omega').$$

4 Probabilistic Timed Computation Tree Logic

We now describe the probabilistic real-time logic PTCTL (Probabilistic Timed Computation Tree Logic) which can be used to specify properties of probabilistic timed systems. PTCTL synthesizes elements from two extensions of the branching temporal logic CTL, namely the real-time temporal logic TCTL [14] and the essentially equivalent, probabilistic temporal logics pCTL and PBTL [7, 5]. In particular, the temporal operator $\mathcal{U}$ ("until") and the path quantifiers $\forall$ and $\exists$ ("for all" and "there exists", respectively) are taken from CTL, the freeze quantifier $z.\phi$ and the facility to refer directly to clock values are taken from TCTL, and the probabilistic operators $[\phi_1 \exists \mathcal{U} \phi_2]_{\sqsupseteq \lambda}$ and $[\phi_1 \forall \mathcal{U} \phi_2]_{\sqsupseteq \lambda}$ are taken from PBTL. Note that the freeze quantifier $z.\phi$ is used to reset the clock z, so that ϕ is evaluated from a state at which $z = 0$. Using our new logic, we can express properties such as, "with probability 0.6 or greater, the value of the system clock x does not exceed 3 before 5 time units have elapsed", which is represented as the PTCTL formula $z.[(x \leq 3) \forall \mathcal{U} (z = 5)]_{\geq 0.6}$.

As with TCTL, PTCTL employs a set of clock variables in order to express timing properties; for this purpose, we introduce a set of *formula clocks*, $\mathcal{Z}$, which is disjoint from $\mathcal{X}$. Such clocks are assigned values by a *formula clock valuation* $\mathcal{E} : \mathcal{Z} \to \mathbb{R}$, which uses the notation for clock valuations in the standard way.

Definition 8 (Atomic Formulae). *Let $\mathcal{C}$ be a set of clocks. A set of* atomic formulae $\mathrm{AF}_{\mathcal{C}}$ *is defined inductively by the syntax:*

$$\varphi ::= c \leq k \,|\, k \leq c \,|\, \neg\varphi \,|\, \varphi \vee \varphi$$

where $c \in \mathcal{C}$ and $k \in \mathbb{N}$. Atomic formulae of the form $c \leq k$ or $k \leq c$ are called minimal atomic formulae.

Definition 9 (Syntax of PTCTL). *The syntax of* PTCTL *is defined as follows:*

$$\phi ::= \texttt{true} \;\;|\;\; a \;\;|\;\; \varphi \;\;|\;\; \phi \wedge \phi \;\;|\;\; \neg\phi \;\;|\;\; z.\phi \;\;|\;\; [\phi \,\exists\, \mathcal{U}\, \phi]_{\sqsupseteq \lambda} \;\;|\;\; [\phi \,\forall\, \mathcal{U}\, \phi]_{\sqsupseteq \lambda}$$

where $a \in \mathrm{AP}$ is an atomic proposition*, $\varphi \in \mathrm{AF}_{\mathcal{X} \cup \mathcal{Z}}$ is an* atomic formula*, $z \in \mathcal{Z}$, $\lambda \in [0, 1]$, and $\sqsupseteq$ is either $\geq$ or $>$.*

Note that the values of system clocks in $\mathcal{X}$ and formula clocks in $\mathcal{Z}$ can be obtained from a state and a formula clock valuation, respectively. Then, if $\varphi \in \mathrm{AF}_{\mathcal{X} \cup \mathcal{Z}}$, and given a state σ and a formula clock valuation $\mathcal{E}$, we denote by $\varphi[\sigma, \mathcal{E}]$ the boolean value obtained by replacing each occurrence of a system clock $x \in \mathcal{X}$ in φ by $\sigma(x)$, and each occurrence of a formula clock $z \in \mathcal{Z}$ in φ by $\mathcal{E}(z)$.

Definition 10 (Satisfaction Relation for PTCTL). *Given a probabilistic timed structure $\mathcal{M}$ and a set $\mathcal{A}$ of adversaries of $\mathcal{M}$, then for any state σ of $\mathcal{M}$, formula clock valuation $\mathcal{E}$, and* PTCTL *formula ϕ, the satisfaction relation $\sigma, \mathcal{E} \models_{\mathcal{A}} \phi$ is defined inductively as follows:*

$$
\begin{array}{ll}
\sigma, \mathcal{E} \models_{\mathcal{A}} \texttt{true} & \text{for all } \sigma \text{ and } \mathcal{E} \\
\sigma, \mathcal{E} \models_{\mathcal{A}} a & \Leftrightarrow \sigma(a) = \texttt{true} \\
\sigma, \mathcal{E} \models_{\mathcal{A}} \varphi & \Leftrightarrow \varphi[\sigma, \mathcal{E}] = \texttt{true} \\
\sigma, \mathcal{E} \models_{\mathcal{A}} \phi_1 \wedge \phi_2 & \Leftrightarrow \sigma, \mathcal{E} \models_{\mathcal{A}} \phi_1 \text{ and } \sigma, \mathcal{E} \models_{\mathcal{A}} \phi_2 \\
\sigma, \mathcal{E} \models_{\mathcal{A}} \neg\phi & \Leftrightarrow \sigma, \mathcal{E} \not\models_{\mathcal{A}} \phi \\
\sigma, \mathcal{E} \models_{\mathcal{A}} z.\phi & \Leftrightarrow \sigma, \mathcal{E}[z \mapsto 0] \models_{\mathcal{A}} \phi \\
\sigma, \mathcal{E} \models_{\mathcal{A}} [\phi_1 \; \exists\mathcal{U} \; \phi_2]_{\sqsupseteq\lambda} & \Leftrightarrow \mathit{Prob}(\{\omega \,|\, \omega \in \mathit{Path}^{A}_{\mathit{ful}}(\sigma) \;\&\; \omega, \mathcal{E} \models_{\mathcal{A}} \phi_1 \; \mathcal{U} \; \phi_2\}) \sqsupseteq \lambda \\
& \quad \text{for some } A \in \mathcal{A} \\
\sigma, \mathcal{E} \models_{\mathcal{A}} [\phi_1 \; \forall\mathcal{U} \; \phi_2]_{\sqsupseteq\lambda} & \Leftrightarrow \mathit{Prob}(\{\omega \,|\, \omega \in \mathit{Path}^{A}_{\mathit{ful}}(\sigma) \;\&\; \omega, \mathcal{E} \models_{\mathcal{A}} \phi_1 \; \mathcal{U} \; \phi_2\}) \sqsupseteq \lambda \\
& \quad \text{for all } A \in \mathcal{A} \\
\omega, \mathcal{E} \models_{\mathcal{A}} \phi_1 \; \mathcal{U} \; \phi_2 & \Leftrightarrow \text{there exists } i \in \mathbb{N}, \text{ and } 0 \leq t \leq t_i \text{ such that} \\
& \quad \omega(i) + t, \mathcal{E} + \mathcal{D}_\omega(i) + t \;\models_{\mathcal{A}}\; \phi_2, \text{ and for all } j \in \mathbb{N} \\
& \quad \text{and } t' \in \mathbb{R} \text{ such that } t' \leq t_j \text{ and } (j, t') \prec (i, t), \\
& \quad \omega(j) + t', \mathcal{E} + \mathcal{D}_\omega(j) + t' \;\models_{\mathcal{A}}\; \phi_1 \vee \phi_2
\end{array}
$$

5 Probabilistic Timed Graphs

This section introduces *probabilistic timed graphs* as a modelling framework for real-time systems with probability. This formalism is derived from timed graphs [2], a variant of timed automata for which model checking of TCTL properties can be performed. Here, we extend timed graphs with discrete probability distributions over edges, so that the choice of the next location of the graph is now probabilistic, in addition to nondeterministic, in nature. Furthermore, we incorporate *invariant conditions* [14] into the probabilistic timed graph in order to enforce upper bounds on the time at which certain probabilistic choices are made.

Definition 11 (Probabilistic Timed Graph). *A probabilistic timed graph is a tuple $G = (\mathcal{S}, L, s_{init}, \mathcal{X}, inv, prob, \langle\tau_s\rangle_{s\in\mathcal{S}})$ where*

- *a finite set $\mathcal{S}$ of nodes,*
- *a function $L : \mathcal{S} \longrightarrow 2^{\mathrm{AP}}$ assigning to each node of the graph the set of atomic propositions that are true in that node,*
- *a start node $s_{init} \in \mathcal{S}$,*
- *a finite set $\mathcal{X}$ of clocks,*
- *a function $inv : \mathcal{S} \longrightarrow \mathrm{AF}_{\mathcal{X}}$ assigning to each node an invariant condition,*
- *a function $prob : \mathcal{S} \to \mathcal{P}_{fn}(\mu(\mathcal{S} \times 2^{\mathcal{X}}))$ assigning to each node a (finite non-empty) set of discrete probability distributions on $\mathcal{S} \times 2^{\mathcal{X}}$,*
- *a family of functions $\langle\tau_s\rangle_{s\in\mathcal{S}}$ where for any $s \in \mathcal{S}$: $\tau_s : prob(s) \longrightarrow \mathrm{AF}_{\mathcal{X}}$ assigns to each $p_s \in prob(s)$ an enabling condition.*

For simplicity, the invariant and enabling conditions are subject to the following assumption: if, in some state in the execution of G, allowing any amount of time to elapse would violate the invariant condition of the current node, then the enabling condition of at least one probability distribution is satisfied. [1]

[1] Another solution is to identify an additional discrete probability distribution $p_s^{inv} \in \mu(\mathcal{S} \times 2^{\mathcal{X}})$ with each $s \in \mathcal{S}$, which becomes enabled in s at the points for which progression of any amount of time would violate the node's invariant $inv(s)$.

The system starts in node s_{init} with all of its clocks initialized to 0. The values of all the clocks increase uniformly with time. At any point in time, if the system is in node s and the invariant condition will not be violated by letting time advance, then the system can either (a) remain in its current node and let time advance, or (b) make a *state transition* if there exists a distribution $p_s \in prob(s)$ whose corresponding enabling condition $\tau_s(p_s)$ is satisfied by the current values of the clocks. Alternatively, if the invariant condition will be violated by letting time advance then the system must make a state transition. State transitions are instantaneous and consist of the following two steps performed in succession: firstly, the system makes a *nondeterministic choice* between the set of distributions $p_s \in prob(s)$ whose corresponding enabling condition $\tau_s(p_s)$ is satisfied by the current values of the clocks. [2] Secondly, supposing that the probability distribution p_s is chosen, the system then makes a *probabilistic transition* according to p_s; that is, for any $s' \in \mathcal{S}$ and $C \subseteq \mathcal{X}$, the probability the system will make a state transition to node s', and reset all the clocks in C to 0, is given by $p_s(s', C)$.

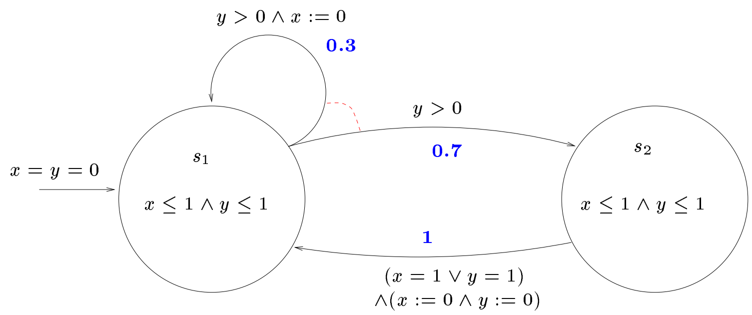

Figure 1. The probabilistic timed graph G_1.

Example. An example of a probabilistic timed graph is given in Figure 1. Control of G_1 initially resides in node s_1, with the system clocks, x and y, each set to 0. Node s_1 has two outgoing edges, both of which have the same enabling condition, $(y > 0)$, and are defined with respect to the probability distribution p_{s_1}, as denoted by the dashed arc connecting the edges at their source. The bold numerals labelling the edges refer to the probabilities of the edges being taken, while assignment labels such as $x := 0$ refer to clock resets. Therefore, the diagram states that when the value of y exceeds 0 and a nondeterministic choice has been made to take an edge according to p_{s_1}, with probability 0.3 control returns to s_1 with the value of x reset to 0, and with probability 0.7 control switches to node s_2. More formally, $p_{s_1}(s_1, \{x\}) = 0.3$ and $p_{s_1}(s_2, \emptyset) = 0.7$. Also note that the invariant condition of s_1, which is shown within the body of the node, states that the probabilistic timed graph cannot allow time to pass if doing

[2] In the case in which we have the special probability distribution, p_s^{inv}, then this distribution *must* be taken at this point.

so would take the value of either x or y above 1; in such a case, a probabilistic choice over the outgoing edges would be forced. The behaviour of the system when control resides in s_2 takes a similar form.

Obtaining a Probabilistic Timed Structure from a Probabilistic Timed Graph. This section will now show that the behaviour of a probabilistic timed graph can be formally stated in terms of a probabilistic timed structure. First, the following notation must be introduced. A *system clock valuation* for the set of clocks $\mathcal{X}$ is a function $\nu : \mathcal{X} \longrightarrow \mathrm{I\!R}$. Let $\Gamma(\mathcal{X})$ denote the set of all system clock valuations for all the clocks of $\mathcal{X}$. The standard notation for clock valuations, as introduced in section 3, is used for system clock valuations ν.

Let $\varphi \in \mathrm{AF}_{\mathcal{X}}$ and $\nu \in \Gamma(\mathcal{X})$. Then $\varphi[\nu]$ is the boolean value obtained by replacing each occurrence of a clock $x \in \mathcal{X}$ in φ by $\nu(x)$. If $\varphi[\nu] = \mathtt{true}$ then we say that ν *satisfies* φ.

Definition 12 (State of a Probabilistic Timed Graph). *A* state *of G is a tuple $\langle s, \nu \rangle$, where $s \in \mathcal{S}$ and $\nu \in \Gamma(\mathcal{X})$ such that ν satisfies $inv(s)$.*

To uniquely identify each node of the probabilistic timed graph, we let a_s be an atomic proposition that is true only in node s. Formally, we extend the set of atomic propositions to $\mathrm{AP}' = \mathrm{AP} \cup \{a_s \mid s \in \mathcal{S}\}$, and the labelling function to $L' : \mathcal{S} \to 2^{\mathrm{AP}'}$, where $L'(s) = L(s) \cup \{a_s\}$ for all $s \in \mathcal{S}$.

We now define a probabilistic timed structure to formally define the behaviour of a probabilistic timed graph. Note that this definition also allows us to interpret PTCTL formulae with atomic propositions from AP' over a probabilistic timed graph.

Definition 13. *For any probabilistic timed graph G, let $\mathcal{M}^G = (\Sigma^G, Tr^G, End^G)$ be the probabilistic timed structure defined as follows:*

- *Σ^G is the set of states of G. For a given state of G, $\langle s, \nu \rangle$, then the corresponding state of $\mathcal{M}^G$ obtained by letting $\langle s, \nu \rangle(a) = \mathtt{true}$, if $a \in L'(s)$, for all $a \in \mathrm{AP}'$, and $\mathtt{false}$ otherwise, and letting $\langle s, \nu \rangle(x) = \nu(x)$ for all $x \in \mathcal{X}$.*
- *Take any $\langle s, \nu \rangle \in \Sigma^G$. Then $(t, p) \in Tr^G(\langle s, \nu \rangle)$, where $t \in \mathrm{I\!R}$ and $p \in \mu(\mathcal{S} \times \Gamma(\mathcal{X}))$, if and only if there exists $p_s \in prob(s)$ such that*
  1. *the clock valuation $\nu + t$ satisfies $\tau_s(p_s)$,*
  2. *$(\nu + t')$ satisfies the invariant condition $inv(s)$ for all $0 \leq t' \leq t$,*
  3. *for any $\langle s', \nu' \rangle$:*

$$p(\langle s', \nu' \rangle) = \sum_{\substack{C \subseteq \mathcal{X} \, \& \\ (\nu+t)[C \mapsto 0] = \nu'}} p_s(s', C) \ .$$

- *End^G comprises of states $\langle s, \nu \rangle$, for which, for any $t \in \mathrm{I\!R}$, $\nu + t$ satisfies $inv(s)$.*

It is now possible to define the set $\mathcal{A}^G$ of adversaries of $\mathcal{M}^G$ using Definition 4.

6 Model Checking Probabilistic Timed Graphs

Note that, because all clocks are real-valued, the state space of a probabilistic timed graph is infinite. However, it was noted in [3] that the space of clock valuations of a timed graph can be partitioned into a finite set of *clock regions*, each containing a finite or infinite number of valuations which, as noted by [2], satisfy the same TCTL formulae. Combination of this partitioning with the transition systems of a timed graph induces a structure called a *region graph*, which can be used for model checking. This section will show that a similar construction can be used for model checking probabilistic timed graphs against PTCTL formulae.

Equivalence of Clock Valuations.

Definition 14. *For any $x \in \mathcal{X}$ let k_x be the largest constant the system clock x is compared to in any of the invariant or enabling conditions.*

Furthermore, for any $\nu \in \Gamma(\mathcal{X})$ and $x \in \mathcal{X}$, x is relevant *for ν if $\nu(x) \leq k_x$.*

Definition 15. *For any $t \in \mathbb{R}$, $\lfloor t \rfloor$ denotes its integral part. Then, for any $t, t' \in \mathbb{R}$, t and t'* agree on their integral parts *if and only if:*

1. *$\lfloor t \rfloor = \lfloor t' \rfloor$,*
2. *both t and t' are integers or neither is an integer.*

Definition 16 (Clock equivalence). *For clock valuations ν and ν' in $\Gamma(\mathcal{X})$, $\nu \cong \nu'$ if and only if the following conditions are satisfied:*

1. *$\forall x \in \mathcal{X}$ either $\nu(x)$ and $\nu'(x)$ agree on their integral parts, or x is not relevant for both ν and ν',*
2. *$\forall x, x' \in \mathcal{X}$ that are relevant for ν, then $\nu(x) - \nu(x')$ and $\nu'(x) - \nu'(x')$ agree on their integral parts.*

Lemma 1. *Let $\nu, \nu' \in \Gamma(\mathcal{X})$ such that $\nu \cong \nu'$. Then the following conditions hold:*

(*a*) *$\nu[C \mapsto 0] \cong \nu'[C \mapsto 0]$ for all $C \subseteq \mathcal{X}$,*
(*b*) *for any $x \in \mathcal{X}$, x is relevant for ν if and only if x is relevant for ν',*
(*c*) *for any atomic formula $\varphi \in \mathrm{AF}_{\mathcal{X}}$, ν satisfies φ if and only if ν' satisfies φ.*

Proof. The proof follows from the definition of $\cong$. □

Let $[\nu]$ denote the equivalence class to which ν belongs, and we refer to elements such as $\langle s, [\nu] \rangle$ as *regions*.

We now extend the concept of clock equivalence to formula clocks. Let $(\nu, \mathcal{E}) : \mathcal{X} \cup \mathcal{Z} \to \mathbb{R}$ be the clock valuation that assigns a real value to each of the system and formula clocks, and let $\Gamma^*(\mathcal{X} \cup \mathcal{Z})$ be the set of all such valuations for G. For a $(\nu, \mathcal{E}) \in \Gamma^*(\mathcal{X} \cup \mathcal{Z})$, and $C \subseteq \mathcal{X} \cup \mathcal{Z}$, we use the notation $(\nu, \mathcal{E})[C \mapsto 0]$

in the usual way. For some $t \in \mathbb{R}$, $(\nu + t, \mathcal{E} + t)$ denotes the clock valuation for which all clocks c in $\mathcal{X} \cup \mathcal{Z}$ take the value $(\nu, \mathcal{E})(c) + t$.

The equivalence relation for such a valuation is defined with respect to a particular PTCTL formula ϕ. For each formula clock $z \in \mathcal{Z}$, we let k_z be the largest constant that z is compared to in the atomic formulae of ϕ, and extend the notion of relevance of Definition 15 to formula clocks in the natural way. Let $\mathcal{E}'$ be the restriction of $\mathcal{E}$ over the clocks of $\mathcal{Z}$ that are referred to in ϕ. We can then extend the equivalence relation from $\cong$ to $\cong^*$ simply by taking $(\nu, \mathcal{E}')$ instead of ν and $\mathcal{X} \cup \mathcal{Z}$ instead of $\mathcal{X}$; the definition of equivalence classes of the form $[\nu, \mathcal{E}']$ then follows in an obvious manner. Furthermore, Lemma 1 holds for $\cong^*$. Because our construction of the equivalence classes will always be with respect to a particular ϕ, we henceforth write $\mathcal{E}$ for $\mathcal{E}'$. An element of the form $\langle s, [\nu, \mathcal{E}] \rangle$ is called an *augmented region*.

Let α be an equivalence class of the form $[\nu, \mathcal{E}]$. Then $\alpha[C \mapsto 0]$ denotes the equivalence class obtained from α by setting all of the clocks in C to 0, and let clock $c \in \mathcal{X} \cup \mathcal{Z}$ *be relevant for* α if $(\nu, \mathcal{E})(c) \leq k_c$, where $(\nu, \mathcal{E})$ is some clock valuation such that $(\nu, \mathcal{E}) \in \alpha$.

The Region Graph. We now define an edge relation over the augmented regions to obtain the *region graph*. The non-probabilistic region construction of [2] results in a state-labelled transition system, which can be model checked using well-established methods. However, in our case the region graph takes the form of a concurrent probabilistic system [5] (and is also equivalent to the probabilistic-nondeterministic systems of [7]), for which there exist model checking techniques for temporal logics with probability bounds.

First, we require some preliminary definitions.

Definition 17 (Satisfaction of formulae). *Let α be an equivalence class of $\Gamma^*(\mathcal{X} \cup \mathcal{Z})$ and $\varphi \in \mathrm{AF}_{\mathcal{X} \cup \mathcal{Z}}$ be an atomic formula. Then α* satisfies *φ if and only if, for any $(\nu, \mathcal{E}) \in \alpha$, the value of φ after substituting each occurrence of $x \in \mathcal{X}$ with $\nu(x)$, and each occurrence of $z \in \mathcal{Z}$ with $\mathcal{E}(z)$, is* `true`. *(Note that the value of φ will be the same for all $(\nu, \mathcal{E}) \in \alpha$, by Lemma 1(c).)*

Definition 18 (Successor Region). *Let α and β be distinct equivalence classes of $\Gamma^*(\mathcal{X} \cup \mathcal{Z})$. The equivalence class β is said to be the* successor *of α if and only if, for each $(\nu, \mathcal{E}) \in \alpha$, there exists a positive $t \in \mathbb{R}$ such that $(\nu + t, \mathcal{E} + t) \in \beta$, and $(\nu + t', \mathcal{E} + t') \in \alpha \cup \beta$ for all $t' \leq t$. We then denote the equivalence class β by $succ(\alpha)$.*

The successor relation can be extended to augmented regions in the following way: $\langle s', \beta \rangle$ is the successor region *of $\langle s, \alpha \rangle$ if $s' = s$ and $\beta = succ(\alpha)$.*

Definition 19 (End Class). *Let α be an equivalence class of $\Gamma^*(\mathcal{X} \cup \mathcal{Z})$. The class α is an* end class *if and only if for all $c \in \mathcal{X} \cup \mathcal{Z}$, c is not relevant for α. Furthermore, for any $s \in \mathcal{S}$, $\langle s, \alpha \rangle$ is an* end region.

We now define a region graph which captures both the probabilistic transitions in G and the movement to new regions due to the passage of time.

Definition 20 (Region Graph). *The* region graph $R(G, \phi)$ *is defined to be the graph* $(V^*, Steps^*, End^*)$. *The vertex set* V^* *is the set of augmented regions, and the set* $End^* \subseteq V^*$ *comprises of the set of end regions. The edge function* $Steps^* : V^* \longrightarrow \mathcal{P}_{fn}(\mu(V^*))$ *includes two types of transitions:* [3]

passage of time: *if* α *is not an end class and the invariant condition* $inv(s)$ *is satisfied by* $succ(\alpha)$, *then* $p^{s,\alpha}_{succ} \in Steps^*(\langle s, \alpha \rangle)$ *where for any* $\langle s', \beta \rangle \in V^*$:

$$p^{s,\alpha}_{succ}(\langle s', \beta \rangle) = \begin{cases} 1 \text{ if } \langle s', \beta \rangle = \langle s, succ(\alpha) \rangle \\ 0 \text{ otherwise.} \end{cases}$$

state transitions of G: $p^{s,\alpha}_{p_s} \in Steps^*(\langle s, \alpha \rangle)$ *if there exists* $p_s \in prob(s)$ *and* α *satisfies the enabling condition* $\tau_s(p_s)$ *such that for any* $s' \in \mathcal{S}$ *and equivalence class* β:

$$p^{s,\alpha}_{p_s}(\langle s', \beta \rangle) = \sum_{\substack{C \subseteq \mathcal{X} \,\&\, \\ [C \mapsto 0]\alpha = \beta}} p_s(s', C).$$

Definition 21 (Path on the Region Graph). *Given an augmented region* $\langle s, \alpha \rangle$, *a* $\langle s, \alpha \rangle$*-path is a finite or infinite path of the form:*

$$\omega^* = \langle s_0, \alpha_0 \rangle \xrightarrow{p^{s_0,\alpha_0}} \langle s_1, \alpha_1 \rangle \xrightarrow{p^{s_1,\alpha_1}} \langle s_2, \alpha_2 \rangle \xrightarrow{p^{s_2,\alpha_2}} \cdots$$

where $\langle s_0, \alpha_0 \rangle = \langle s, \alpha \rangle$, $s_i \in \mathcal{S}$, α_i *is an equivalence class of* $\Gamma^*(\mathcal{X} \cup \mathcal{Z})$ *and* $p^{s_i,\alpha_i} \in Steps^*(\langle s_i, \alpha_i \rangle)$ *such that* $p^{s_i,\alpha_i}(\langle s_{i+1}, \alpha_{i+1} \rangle) > 0$.

We define adversaries on the region graph $R(G, \phi)$ as follows:

Definition 22 (Adversaries on the Region Graph). *An adversary* A^* *on the region graph is a function* A^* *mapping every finite path* ω^* *of* $R(G, \Phi)$ *to a distribution* p *such that* $p \in Steps^*(last(\omega^*))$.

We can then define the sets of paths $Path^*_{fin}$ and $Path^*_{ful}$, and those associated with an adversary, $Path^{A^*}_{fin}$ and $Path^{A^*}_{ful}$, as before. Note that end regions take the role of end states in the definition of the finite paths of $Path^*_{ful}$ and $Path^{A^*}_{ful}$.

With each adversary A^* we can associate a Markov chain. If A^* is an adversary of the region graph $R(G, \phi)$, then $MC^{A^*} = (Path^{A^*}_{fin}, \mathbf{P}^{A^*})$ is a Markov chain where, for the augmented regions $\langle s, \alpha \rangle$, $\langle s', \alpha' \rangle$, and $last(\omega^*) = \langle s, \alpha \rangle$:

$$\mathbf{P}^{A^*}(\omega^*, \omega'^*) = \begin{cases} p^{s,\alpha}(\langle s', \alpha' \rangle) \text{ if } A^*(\omega^*) = p^{s,\alpha} \text{ and } \omega'^* = \omega^* \xrightarrow{p^{s,\alpha}} \langle s', \alpha' \rangle \\ 0 \qquad\qquad\qquad\quad \text{otherwise} \end{cases}$$

Definition 23 (Divergent Adversaries on the Region Graph). *An adversary* A^* *is* divergent *if and only if for all infinite paths* $\omega^* \in Path^{A^*}_{ful}$, *there exist infinitely many* $n \in \mathbb{N}$ *such that one of the following holds:*

[3] If the model includes the distributions p^{inv}_s then we need to add an extra condition in the definition.

1. $\omega^*(n)$ *is an end region,*
2. $\omega^*(n+1)$ *is the successor region of* $\omega^*(n)$.

Let $\mathcal{A}^*_{div}$ *be the set of divergent adversaries on the region graph.*

Such divergent adversaries on the region graph $R(G,\phi)$ correspond to an infinite number of adversaries on the underlying probabilistic timed structure $\mathcal{M}^G$, some of which will be divergent in the sense of Definition 5. Conversely, for any divergent adversary of $\mathcal{M}^G$, there exists a corresponding divergent adversary on $R(G,\phi)$. We observe that the notion of divergent paths of $R(G,\phi)$ induced by adversaries in $\mathcal{A}^*_{div}$ differs from that of *fair paths* of the region graph as presented in [2] because of our assumption of weakly monotonic time.

As in, for example, [5, 7], we define the function $Prob^*$ as the unique measure on the σ-algebra $\mathcal{F}^*_{Path}$.

Model Checking. A method for model checking probabilistic timed graphs against PTCTL formulae will now be presented. This approach takes the form of three steps: construction of the region graph as a finite state representation of the probabilistic timed graph in question, obtaining a formula of an extension of the probabilistic logic PBTL, and then resolving this new formula on the region graph.

First, we turn our attention to the structure over which the PBTL formula will be resolved. Formulae of PBTL are interpreted over 'PBTL-structures', which are concurrent probabilistic systems extended with a vertex labelling function. As our region graph $R(G,\phi)$ is a concurrent probabilistic system, adding an appropriately defined labelling function will convert it into a PBTL-structure which we will call a *labelled region graph*. We define AF_ϕ as the set of minimal atomic formulae appearing in the given PTCTL formula ϕ. For every atomic formula in $\varphi \in \mathrm{AF}_\phi$, we extend the set AP with the atomic proposition a_φ. We denote the resulting set of atomic propositions by AP^*.

Definition 24 (Labelled Region Graph). *For a given region graph* $R(G,\phi)$, *we define its associated* labelled region graph *by* $(R(G,\phi), L^*)$, *where the vertex labelling function,* $L^* : V^* \to 2^{\mathrm{AP}^*}$, *is defined by the following. For a given* $\langle s, [\nu, \mathcal{E}]\rangle$, *we let:*

$$L^*(\langle s,[\nu,\mathcal{E}]\rangle) = \{a \in L(s)\} \cup \{a_\varphi \mid [\nu,\mathcal{E}] \textit{ satisfies } \varphi,\ \varphi \in \mathrm{AF}_\phi\}$$

Next, we present an adjusted syntax of PBTL. Note that we omit PBTL's 'bounded until' operator, because an equivalent, dense time concept can be defined by nesting a PTCTL until operator within a freeze quantifier, and its 'next step' operator, which has no analogue in the case of dense real-time. However, we extend PBTL with a freeze quantifier expression.

Definition 25 (Syntax of PBTL). *The syntax of* PBTL *is defined as follows:*

$$\Phi ::= \mathtt{true} \ \mid\ a \ \mid\ \Phi \wedge \Phi \ \mid\ \neg\Phi \ \mid\ z.\Phi \ \mid\ [\Phi\, \exists\, \mathcal{U}\, \Phi]_{\sqsupseteq\lambda} \ \mid\ [\Phi\, \forall\, \mathcal{U}\, \Phi]_{\sqsupseteq\lambda}$$

where $a \in \mathrm{AP}^*$ *is an atomic proposition,* $z \in \mathcal{Z}$, $\lambda \in [0,1]$, *and* $\sqsupseteq$ *is either* $\geq$ *or* $>$.

Definition 26 (Satisfaction Relation for PBTL). *Given a labelled region graph $(R(G,\phi), L^*)$ and a set $\mathcal{A}^*$ of adversaries on $R(G,\phi)$, then for any augmented region $\langle s,[\nu,\mathcal{E}]\rangle$ of $R(G,\phi)$, and* PBTL *formula Φ, the satisfaction relation $\langle s,[\nu,\mathcal{E}]\rangle \models_{\mathcal{A}^*} \Phi$ is defined inductively as follows:*

$$\begin{array}{lll}
\langle s,[\nu,\mathcal{E}]\rangle \models_{\mathcal{A}^*} \mathtt{true} & & \text{for all } \langle s,[\nu,\mathcal{E}]\rangle \\
\langle s,[\nu,\mathcal{E}]\rangle \models_{\mathcal{A}^*} a & \Leftrightarrow & a \in L^*(\langle s,[\nu,\mathcal{E}]\rangle) \\
\langle s,[\nu,\mathcal{E}]\rangle \models_{\mathcal{A}^*} \Phi_1 \wedge \Phi_2 & \Leftrightarrow & \langle s,[\nu,\mathcal{E}]\rangle \models_{\mathcal{A}^*} \Phi_1 \text{ and } \langle s,[\nu,\mathcal{E}]\rangle \models_{\mathcal{A}^*} \Phi_2 \\
\langle s,[\nu,\mathcal{E}]\rangle \models_{\mathcal{A}^*} \neg\Phi & \Leftrightarrow & \langle s,[\nu,\mathcal{E}]\rangle \not\models_{\mathcal{A}^*} \Phi \\
\langle s,[\nu,\mathcal{E}]\rangle \models_{\mathcal{A}^*} z.\Phi & \Leftrightarrow & \langle s,[\nu,\mathcal{E}[z \mapsto 0]]\rangle \models_{\mathcal{A}^*} \Phi \\
\langle s,[\nu,\mathcal{E}]\rangle \models_{\mathcal{A}^*} [\Phi_1 \,\exists\, \mathcal{U}\, \Phi_2]_{\sqsupseteq\lambda} & \Leftrightarrow & Prob^*(\{\omega \,|\, \omega \in Path^{A^*}_{ful}(\langle s,[\nu,\mathcal{E}]\rangle)\ \& \\
 & & \omega \models_{\mathcal{A}^*} \Phi_1\, \mathcal{U}\, \Phi_2\}) \sqsupseteq \lambda \text{ for some } A^* \in \mathcal{A}^* \\
\langle s,[\nu,\mathcal{E}]\rangle \models_{\mathcal{A}^*} [\Phi_1 \,\forall\, \mathcal{U}\, \Phi_2]_{\sqsupseteq\lambda} & \Leftrightarrow & Prob^*(\{\omega \,|\, \omega \in Path^{A^*}_{ful}(\langle s,[\nu,\mathcal{E}]\rangle)\ \& \\
 & & \omega \models_{\mathcal{A}^*} \Phi_1\, \mathcal{U}\, \Phi_2\}) \sqsupseteq \lambda \text{ for all } A^* \in \mathcal{A}^* \\
\omega \models_{\mathcal{A}^*} \Phi_1\, \mathcal{U}\, \Phi_2 & \Leftrightarrow & \text{there exists } i \in \mathbb{N}, \text{ such that } \omega(i) \models_{\mathcal{A}^*} \Phi_2, \\
 & & \text{and for all } j \in \mathbb{N} \text{ such that } 0 \le j < i \text{ and} \\
 & & \omega(j) \models_{\mathcal{A}^*} \Phi_1
\end{array}$$

Furthermore, a PBTL formula, Φ, can be *derived from* a PTCTL formula, ϕ, by applying the following rules inductively:

Subformula of ϕ_i	Subformula of Φ_i
$\mathtt{true}$	$\mathtt{true}$
a	a
φ	a_φ
$\phi_1 \wedge \phi_2$	$\Phi_1 \wedge \Phi_2$
$\neg\phi$	$\neg\Phi$
$z.\phi$	$z.\Phi$
$[\phi_1 \,\exists\, \mathcal{U}\, \phi_2]_{\sqsupseteq\lambda}$	$[\Phi_1 \,\exists\, \mathcal{U}\, \Phi_2]_{\sqsupseteq\lambda}$
$[\phi_1 \,\forall\, \mathcal{U}\, \phi_2]_{\sqsupseteq\lambda}$	$[\Phi_1 \,\forall\, \mathcal{U}\, \Phi_2]_{\sqsupseteq\lambda}$

Figure 2 presents the region construction of the probabilistic timed graph of Figure 1. As before, the probabilistic transitions are linked with an arc at their source vertex. In order for the reader to easily comprehend the behaviour of the region graph, each vertex has been labelled with a constraint that is satisfied by all of the clock valuations within that augmented region. Consider the following PTCTL formula:

$$\phi_1 = [(y=0)\exists\mathcal{U}[(x>0)\exists\mathcal{U}(y=0)]_{\geq 0.7}]_{\geq 1}.$$

ϕ_1 can be interpreted over this graph by first converting it into the equivalent PBTL formula:

$$\Phi_1 = [a_{(y=0)}\exists\mathcal{U}[a_{(x>0)}\exists\mathcal{U}a_{(y=0)}]_{\geq 0.7}]_{\geq 1}.$$

Φ_1 is satisfied by this region graph, and therefore we conclude that the probabilistic timed graph G_1 satisfies ϕ_1. Note that the following PTCTL formula, ϕ_2, is *not* satisfied by the region graph:

$$\phi_2 = [(y=0)\exists\mathcal{U}[(x>0)\exists\mathcal{U}(y=0)]_{>0.7}]_{\geq 1}$$

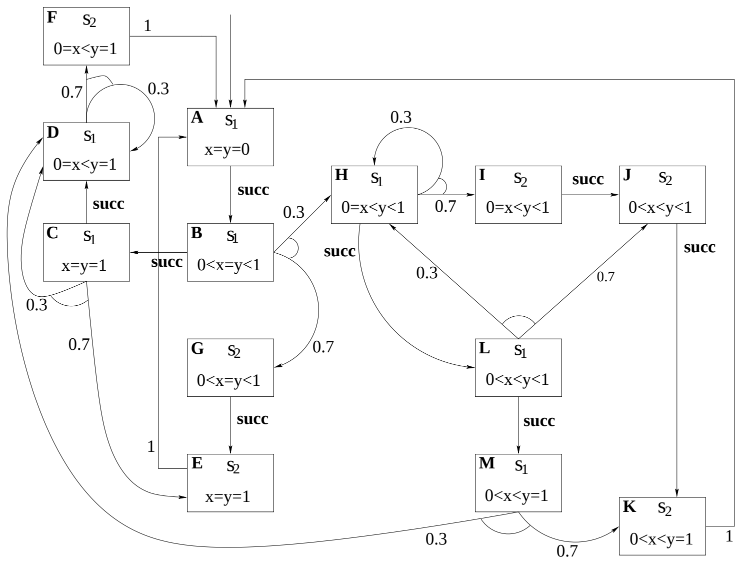

Figure 2. The region graph of the probabilistic timed graph G_1.

Proposition 1. (Correctness of the model checking procedure) *Given the probabilistic timed graph G, state $\langle s, \nu \rangle$ of $\mathcal{M}^G$ and formula clock valuation $\mathcal{E}$ satisfies the PTCTL formula ϕ if and only if vertex $\langle s, [\nu, \mathcal{E}] \rangle$ of $(R(G, \phi), L^*)$ satisfies the PBTL formula Φ, where Φ is derived from ϕ.*

Proof. Before considering any temporal or probabilistic operators, we must be convinced that subformulae comprising only of atomic propositions, atomic formulae, boolean connectives and freeze quantifiers are resolved satisfactorily. We proceed to show this by induction on the structure of ϕ.

If $\phi = \texttt{true}$, then ϕ will be true for all states of $\mathcal{M}^G$ and all formula clock valuations. Here ϕ derives $\Phi = \texttt{true}$, which is true for all vertices in $(R(G, \phi), L^*)$.

If $\phi = a$, where $a \in \mathrm{AP}$, then it is true for state $\langle s, \nu \rangle$ of $\mathcal{M}^G$ and all formula clock valuations if and only if $\langle s, \nu \rangle (a) = \texttt{true}$. We also know that $\langle s, \nu \rangle (a) = \texttt{true}$ if and only if $a \in L(s)$. By Definition 24, $a \in L^*(\langle s, [\nu, \mathcal{E}] \rangle)$ if $a \in L(s)$, so $\Phi = a$ is true for the vertex $\langle s, [\nu, \mathcal{E}] \rangle$.

If $\phi = \varphi$, where φ is a minimal atomic formula, then the state $\langle s, \nu \rangle$ of $\mathcal{M}^G$ and formula clock valuation $\mathcal{E}$ satisfies φ if $\varphi[\sigma, \mathcal{E}] = \texttt{true}$. Then, from Definition 24, $a_\varphi \in L^*(\langle s, [\nu, \mathcal{E}] \rangle)$. Because $\Phi = a_\varphi$, and Φ is derived from ϕ, both ϕ and Φ resolve to true in $\langle s, \nu \rangle, \mathcal{E}$ and $\langle s, [\nu, \mathcal{E}] \rangle$ respectively.

The cases of the boolean connectives, $\neg$ and $\wedge$, are self-evident.

If $\phi = z.\phi_1$, then, for a given state $\langle s, \nu \rangle$ and formula clock valuation $\mathcal{E}$ that satisfies ϕ, we know that the augmented region $\langle s, [\nu, \mathcal{E}] \rangle$ will also satisfy $z.\Phi_1$, by observing the following argument. By Definition 10

$$
\begin{aligned}
\langle s, \nu \rangle, \mathcal{E} \models_{\mathcal{A}^G_{div}} z.\phi &\Leftrightarrow \langle s, \nu \rangle, \mathcal{E}[z \mapsto 0] \models_{\mathcal{A}^G_{div}} \phi \\
&\Leftrightarrow \langle s, [\nu, \mathcal{E}[z \mapsto 0]] \rangle \models_{\mathcal{A}^*_{div}} \Phi \quad \text{by induction} \\
&\Leftrightarrow \langle s, [\nu, \mathcal{E}] \rangle \models_{\mathcal{A}^*_{div}} z.\Phi \qquad \text{by Definition 26}
\end{aligned}
$$

Now we show that $\langle s,\nu\rangle,\mathcal{E} \models_{\mathcal{A}^G_{div}} [\phi_1 \ \exists\mathcal{U}\ \phi_2]_{\sqsupseteq\lambda}$, if and only if $\langle s,[\nu,\mathcal{E}]\rangle \models_{\mathcal{A}^*_{div}} [\Phi_1 \ \exists\mathcal{U}\ \Phi_2]_{\sqsupseteq\lambda}$. Our presentation is split into three sections:

1. showing that, for a path ω of $\mathcal{M}^G$, a corresponding path of $(R(G,\phi),L^*)$, $[\omega]$, can be constructed. Furthermore, both of these paths have the same corresponding notion of divergence; that is, ω is divergent if and only if $[\omega]$ is divergent. It also follows that, given path ω^* of the region graph, we can construct ω such that $[\omega]=\omega^*$.

2. showing that the two paths ω and $[\omega]$ are associated with the same probability value.

3. showing $\omega,\mathcal{E} \models_{\mathcal{A}^G_{div}} \phi_1\mathcal{U}\phi_2$ if and only if $[\omega] \models_{\mathcal{A}^*_{div}} \Phi_1\mathcal{U}\Phi_2$, where the initial augmented state of $[\omega]$ comprises of $\mathcal{E}$.

1. Consider the following property, which shall henceforth be referred to as the *sequence property*. Take a particular node of G, $s\in\mathcal{S}$, and a clock valuation $(\nu,\mathcal{E})$, and consider the sequence of equivalence classes, $[\nu+d_1,\mathcal{E}+d_1],...,[\nu+d_k,\mathcal{E}+d_k]$, where each equivalence class satisfies $inv(s)$, for all $1\le i\le k$, $d_i\in\mathbb{R}$, and for all $1\le l<k$, $succ([\nu+d_l,\mathcal{E}+d_l])=[\nu+d_{l+1},\mathcal{E}+d_{l+1}]$. Then, for the time value $d\in\mathbb{R}$, where $d_1\le d\le d_k$, we know that $(\nu+d,\mathcal{E}+d)\cong^*(\nu+d_j,\mathcal{E}+d_j)$, for some $1\le j\le k$. We can then write $(\nu+d,\mathcal{E}+d)\in[\nu+d_j,\mathcal{E}+d_j]$.

The sequence property allows us to state the following. Consider the path ω of $\mathcal{M}^G$, such that:

$$\omega=\langle s_0,\nu_0\rangle \xrightarrow{t_0,p_0} \langle s_1,\nu_1\rangle \xrightarrow{t_1,p_1} \cdots$$

Take a particular $i\ge 0$. From $\langle s_i,\nu_i\rangle$, and letting t_i time units elapse, we may cross into a number of equivalence classes before making the next edge transition. We let m_i be this number. Let $v_{i0}=\langle s_i,[\nu_i,\mathcal{E}+\mathcal{D}_\omega(i)]\rangle$, and $v_{ij}=\langle s_i,[\nu_i+d_j,\mathcal{E}+\mathcal{D}_\omega(i)+d_j]\rangle$ for some $1\le j\le m_i$, and $d_j\in\mathbb{R}$. Then we can construct the finite path $[\omega_i]$ of the region graph, such that:

$$[\omega_i]=v_{i0} \xrightarrow{p^{v_{i0}}_{succ}} v_{i1} \xrightarrow{p^{v_{i1}}_{succ}} \cdots v_{im_i} \xrightarrow{p^{v_{im_i}}_{p_{s_i}}} v_{(i+1)0}\ .$$

Let $[\omega]=[\omega_0][\omega_1]\cdots$ be the concatenation of all such segments.

This construction also works in the opposite direction. Let $\langle s,\alpha_1\rangle\to\cdots\to\langle s,\alpha_n\rangle$ be a path through the region graph such that all of its edges correspond to p^{s,α_i}_{succ} transitions. Now suppose that $\langle s,\alpha_n\rangle \xrightarrow{p^{s,\alpha}_{p_s}} \langle s',\alpha'\rangle$. Then, assuming that we have a partially constructed, finite path ω of $\mathcal{M}^G$ such that $|\omega|=i$, and $(\nu_i,\mathcal{E}+\mathcal{D}_\omega(i))\in\alpha_i$, then it follows that ω can be extended by the transition $\langle s_i,\nu_i\rangle \xrightarrow{(t_i,p_i)} \langle s_{i+1},\nu_{i+1}\rangle$, where $t_i\in\mathbb{R}$ and p_i is derived from p_s in the usual way. Therefore, we have a method for constructing infinite or finite paths of $\mathcal{M}^G$ from paths of $(R(G,\phi),L^*)$. We note that such paths may be finite if the region graph reaches an end class from which no transitions are enabled, and that paths induced by divergent adversaries of the region graph will guarantee the existence of corresponding time-divergent paths of $\mathcal{M}^G$. Then it follows that, given ω^* of $(R(G,\phi),L^*)$, we can construct ω of $\mathcal{M}^G$ such that $[\omega]=\omega^*$.

2. Now we must show that the probability value associated with the paths ω and $[\omega]$ are the same. Consider the segment of ω, $\omega_i=\langle s_i,\nu_i\rangle \xrightarrow{t_i,p_i} \langle s_{i+1},\nu_{i+1}\rangle$, and the segment of $[\omega]$:

$$[\omega_i] = v_{i0} \xrightarrow{p_{succ}^{v_{i0}}} v_{i1} \xrightarrow{p_{succ}^{v_{i1}}} \cdots v_{im_i} \xrightarrow{p_{p_{s_i}}^{v_{im_i}}} v_{(i+1)0} \ .$$

We wish to show that $\mathit{Prob}_{\mathit{fin}}(\omega_i) = \mathit{Prob}^*_{\mathit{fin}}([\omega_i])$. Consider the transition $v_{ij} \xrightarrow{p_{succ}^{v_{ij}}} v_{i(j+1)}$, for $1 \leq j < m_i$. Then, from the sequence property and the above construction of $[\omega]$, we know that $v_{i(j+1)}$ is a time successor of v_{ij}, and therefore $p_{succ}^{v_{ij}} = 1$. Therefore, our problem reduces to showing that:

$$\mathit{Prob}_{\mathit{fin}}(\langle s_i, \nu_i \rangle \xrightarrow{t_i, p_i} \langle s_{i+1}, \nu_{i+1} \rangle) = \mathit{Prob}^*_{\mathit{fin}}(v_{im_i} \xrightarrow{p_{p_{s_i}}^{v_{im_i}}} v_{(i+1)0}) \ .$$

By the definitions of Prob and Prob^*, this reduces to showing that:

$$p_i(\langle s_{i+1}, \nu_{i+1} \rangle) = p_{p_{s_i}}^{v_{im_i}}(\langle s_i, [\nu_{i+1}, \mathcal{E} + \mathcal{D}_\omega(i+1)] \rangle) \ .$$

Firstly, we note that $\nu_i + t_i \in [\nu_i + d_{m_i}]$. Because all of the system clock valuations in $[\nu_i + d_{m_i}]$ will enable the same probability distributions, we know that the same distributions are enabled in $\langle s_i, \nu_i + t_i \rangle$ and $v_{im_i} = \langle s_i, [\nu_i + d_{m_i}, \mathcal{E} + \mathcal{D}_\omega(i) + d_{m_i}] \rangle$. Furthermore, the move from $\langle s_i, \nu_i + t_i \rangle$ to $\langle s_{i+1}, \nu_{i+1} \rangle$ in $\mathcal{M}^G$, and from v_{im_i} to $v_{(i+1)0}$ in $R(G, \phi)$, will correspond to the choice of the *same* probability distribution of G. We denote this distribution by p_{s_i}. Recall from Definition 13 that:

$$p_i(\langle s_{i+1}, \nu_{i+1} \rangle) = \sum_{\substack{C \subseteq \mathcal{X} \ \& \\ (\nu_i + t_i)[C \mapsto 0] = \nu_{i+1}}} p_{s_i}(s_{i+1}, C) \ ,$$

and from the definition of the region graph:

$$p_{p_{s_i}}^{s_i, \alpha}(\langle s_{i+1}, \beta \rangle) = \sum_{\substack{C \subseteq \mathcal{X} \ \& \\ \alpha[C \mapsto 0] = \beta}} p_{s_i}(s_{i+1}, C) \ ,$$

where $\alpha = [\nu_i + d_{m_i}, \mathcal{E} + \mathcal{D}_\omega(i) + d_{m_i}]$ and $\beta = [\nu_{i+1}, \mathcal{E} + \mathcal{D}_\omega(i+1) + d_{i+1}]$. We know that, for any $C \subseteq \mathcal{X}$, $(\nu_i + t_i)[C \mapsto 0] \in [\nu_i + d_{m_i}][C \mapsto 0]$, and, trivially, that $\nu_{i+1} \in [\nu_{i+1}]$, and so the combinations of $C \subseteq \mathcal{X}$ used in both summations above will be the same. Therefore, the same probability values will be summed in the case of $\mathcal{M}^G$ and that of $R(G, \phi)$, and we can conclude that $p_i(\langle s_{i+1}, \nu_{i+1} \rangle) = p_{p_{s_i}}^{s_i, \alpha}(\langle s_{i+1}, \beta \rangle)$. We can repeat such a process for all $i \in \mathbb{N}$ and, by the definitions of Prob and Prob^*, show that the probability value associated with the paths ω and $[\omega]$ are the same.

3. Next we prove $\omega, \mathcal{E} \models_{\mathcal{A}^G_{div}} \phi_1 \mathcal{U} \phi_2$ if and only if $[\omega] \models_{\mathcal{A}^*_{div}} \Phi_1 \mathcal{U} \Phi_2$. If $\omega(i) = \langle s_i, \nu_i \rangle$ for all $i \in \mathbb{N}$, then $\omega, \mathcal{E} \models_{\mathcal{A}^G_{div}} \phi_1 \mathcal{U} \phi_2$

$\Leftrightarrow$ $\exists i \in \mathbb{N}$ and $0 \leq t \leq t_i$ such that $\omega(i) + t, \mathcal{E} + \mathcal{D}_\omega(i) + t \models_{\mathcal{A}^G_{div}} \phi_2$
and $\forall j \in \mathbb{N}$ and $t' \in \mathbb{R}$ such that $t' \leq t_j$ & $(j, t') \prec (i, t)$,
$\omega(j) + t', \mathcal{E} + \mathcal{D}_\omega(j) + t' \models_{\mathcal{A}^G_{div}} \phi_1 \vee \phi_2$
by Definition 10

$\Leftrightarrow$ $\exists i \in \mathbb{N}$ and $0 \leq t \leq t_i$ such that $\langle s_i, [\nu_i + t, \mathcal{E} + \mathcal{D}_\omega(i) + t] \rangle \models_{\mathcal{A}^*_{div}} \Phi_2$
and $\forall j \in \mathbb{N}$ and $t' \in \mathbb{R}$ such that $t' \leq t_j$ & $(j, t') \prec (i, t)$,
$\langle s_j, [\nu_j + t', \mathcal{E} + \mathcal{D}_\omega(j) + t'] \rangle \models_{\mathcal{A}^*_{div}} \Phi_1 \vee \Phi_2$
by induction

$\Leftrightarrow$ $\exists i' \in \mathbb{N}$ such that $[\omega](i') \models_{\mathcal{A}^*_{div}} \Phi_2$ and $[\omega](j') \models_{\mathcal{A}^*_{div}} \Phi_1 \vee \Phi_2 \ \forall j' \leq i'$
by construction of $[\omega]$

$\Leftrightarrow$ $[\omega] \models_{\mathcal{A}^*_{div}} \Phi_1 \mathcal{U} \Phi_2$ by Definition 26

It follows by the definition of adversaries, both on probabilistic timed structures and the region graph, and the construction of **1**, that for all $A \in \mathcal{A}_{div}^{G}$, there exists an adversary $[A] \in \mathcal{A}_{div}^{*}$ such that, for some $\mathcal{E}$,

$$Path_{ful}^{[A]}(\langle s, [\nu, \mathcal{E}] \rangle) = \{[\omega] \mid \omega \in Path_{ful}^{A}(\langle s, \nu \rangle)\}.$$

Conversely, given a path in the region graph, we can construct a path of $\mathcal{M}^G$ (see [2]). Using this construction, we can show that, for all adversaries $A^* \in \mathcal{A}_{div}^{*}$ of the region graph, there exists an adversary $A \in \mathcal{A}_{div}^{G}$ such that $[A] = A^*$.

From **2**, we know that the probability values associated with ω and $[\omega]$ are the same. Then we can conclude that:

$$Prob^*\{\omega^* \mid \omega^* \in Path_{ful}^{A^*}\} = Prob\{\omega \mid \omega \in Path_{ful}^{A}\}$$

for some $A \in \mathcal{A}_{div}^{G}$. □

Using the transformation presented above, we can obtain a PBTL formula, Φ, from the PTCTL formula, ϕ. Now we can use the model checking algorithm of [5] in order to verify whether the PBTL formula Φ holds in an initial state of the region graph, $\langle s_{init}, [\nu^0, \mathcal{E}] \rangle$, where, for all $x \in \mathcal{X}$, $(\nu^0, \mathcal{E})(x) = 0$ and $\mathcal{E}$ is an arbitrary formula clock valuation.

7 Conclusions

We conclude with a brief analysis of the complexity of our method. The time complexity of PBTL model checking is polynomial in the size of the system (measured by the number of states and transitions) and linear in the size of the formula [5] (see also the recent improvement [4]). Since the translation from PTCTL to the extended PBTL has no effect on the size of the formula, it follows that the model checking for PTCTL against probabilistic timed systems will be polynomial in the size of the region graph and linear in the size of the PTCTL formula. Note that the addition of probability distributions to timed automata does not significantly increase the size of the region graph over the size of the non-probabilistic region graph, and that the freeze quantifier formulae we have added to PBTL can be handled in a straightforward manner.

Future work could address the potential inefficiencies of this method. Model checking of real-time systems is expensive, with its complexity being exponential in the number of clocks and the magnitude of their upper bounds (denoted by k_c in our presentation). However, a number of techniques for combating this inefficiency have been developed (see [16]), and could be applied in this context.

Another potential avenue of research is the application of the methods of this paper to *hybrid automata*, a model for discrete-continuous systems which allows more general continuous dynamics than timed automata. In particular, it is known that certain classes of hybrid automata are reducible to timed automata [12], and other classes have finite bisimilarity quotients [15], both of which may be particularly adaptable to probabilistic extensions.

References

1. R. Alur, C. Courcoubetis, and D. Dill. Model-checking for probabilistic real-time systems. In *Automata, Languages and Programming: Proceedings of the 18th ICALP*, Lecture Notes in Computer Science 510, pages 115–126, 1991.
2. R. Alur, C. Courcoubetis, and D. Dill. Model-checking in dense real-time. *Information and Computation*, 104(1):2–34, 1993. Preliminary version appears in the Proc. of 5th LICS, 1990.
3. R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994. Preliminary version appears in Proc. 17th ICALP, 1990, LNCS 443.
4. C. Baier. Personal communication, 1998.
5. C. Baier and M. Kwiatkowska. Model checking for a probabilistic branching time logic with fairness. *Distributed Computing*, 11:125–155, 1998.
6. J. Bengtsson, K. Larsen, F. Larsson, P. Pettersson, W. Yi, and C. Weise. New generation of UPPAAL. In *Proceedings of the International Workshop on Software Tools for Technology Transfer*, Aalborg, Denmark, July 1998.
7. A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In *Foundations of Software Technology and Theoretical Computer Science*, volume 1026 of *Lecture Notes in Computer Science*, pages 499–513, 1995.
8. M. Bozga, C. Daws, O. Maler, A. Olivero, S. Tripakis, and S. Yovine. Kronos: a model-checking tool for real-time systems. In *Proc. of the 10th Conference on Computer-Aided Verification*, Vancouver, Canada, 28 June - 2 July 1998. Springer Verlag.
9. P. D'Argenio, J.-P. Katoen, T. Ruys, and J. Tretmans. Modeling and verifying a bounded retransmission protocol. In Z. Brezocnik and T. Kapus, editors, *Proc. of COST 247 International Workshop on Applied Formal Methods in System Design,* Maribor, Slovenia, Technical Report. University of Maribor, 1996.
10. H. Gregersen and H. E. Jensen. Formal design of reliable real time systems. Master's thesis, Department of Mathematics and Computer Science, Aalborg University, 1995.
11. H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6(5):512–535, 1994.
12. T. Henzinger, P. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata? *Journal of Computer and System Sciences*, 57(1):94–124, Aug. 1998.
13. T. Henzinger and O. Kupferman. From quantity to quality. In O. Maler, editor, *HART 97: Hybrid and Real-time Systems*, Lecture Notes in Computer Science 1201, pages 48–62. Springer-Verlag, 1997.
14. T. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine. Symbolic model checking for real-time systems. *Information and Computation*, 111(2):193–244, 1994. Special issue for LICS 92.
15. G. Lafferriere, G. Pappas, and S. Yovine. Decidable hybrid systems. Technical Report UCB/ERL M98/39, University of California at Berkeley, June 1998.
16. S. Yovine. Model checking timed automata. In G. Rozenberg and F. Vaandrager, editors, *Embedded Systems*, volume 1494 of *Lecture Notes in Computer Science*. Springer, 1998.