AALBORG UNIVERSITY
INSTITUTE FOR ELECTRONIC SYSTEMS
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

TITLE:
Formal Design of Reliable Real Time Systems

THEME:
Reliable Real Time Systems

PROJECT PERIOD:
1 Feb 1995 - 29 May 1995

PARTICIPANTS:
Heidi Gregersen
Henrik E. Jensen

SUPERVISOR:
Hans Hüttel

COPIES: 7

PAGES: 77

ABSTRACT:

This thesis presents formal techniques to be used when describing and developing reliable real time systems.

Three different probabilistic real time logics, used to specify about reliable real time processes in terms of timed probabilistic graphs, are presented. Algorithms for verifying implementations with respect to specifications in the developed logics are presented.

An algorithm to construct timed probabilistic graphs from specifications in one of the logics is presented.

Finally, the construction of characteristic formulas for timed probabilistic acyclic graphs is presented.

Preface

The present report is our Master's thesis, reporting work carried out during the spring of 1995, leading to the degree of M.Sc. in Computer Science at the Department of Mathematics and Computer Science at Aalborg University.

The work presented in this report assumes a basic knowledge of operational semantics and modal logic.

We would like to thank our supervisor Hans Hüttel for always being willing to listen and discuss new ideas.

Aalborg, May 29, 1995.

Heidi Gregersen

Henrik E. Jensen

Contents

Chapter 1

Introduction

The use of computers in many of the artifacts that we use in our daily lives is still rapidly increasing. Everything, from simple toys to advanced control processes i airplanes, medical equipment and nuclear plants is controlled by computer units which in this way are becoming integral parts in continuously changing environments.

Computer programs embedded in systems of the above kind are what we call *reactive* or *real time* systems. That is, the programs must be able to react to environment changes in real time. Furthermore, systems of the above kind are often required to be tolerant towards fault and therefore we extend the notion of these systems to be *reliable* real time systems. Usually such systems are both parallel and distributed in nature.

Obviously, correctness is of vital importance for reliable real time systems and as a consequence the need for using formal models in the development methods for developing the above systems is obvious.

1.1 Software development

The development of software is traditionally split into three activities. An analysis of the problem is performed. The requirements to the program are written in a *specification language* and a system, satisfying the specified requirements, is implemented in an *implementation language*.

If an implementation could be generated automatically from a given specification, many of the problems that software developers face might be solved. Unfortunately this approach is very difficult to apply in practice. The reason is that the gap between the desired degree of expressiveness in the specification language and the technical need for a somewhat specific implementation language is in general too large.

Due to the above, a fourth activity is introduced in the development process. This is the *verification* activity. In traditional development methods this activity is very often performed as a *test* of the implemented program. Though, when considering reliable real time systems the test approach can be too unreliable as it is not possible to perform tests for all possible sequences of program behaviour, due to the real time environment. Therefore, using a formal model in all development activities also allows for a formal verification strategy which enables us to actually verify all possible behaviours of a formal model of a reliable real time system.

1.2 Specification and Verification

The essential requirement to a formal specification language is that it has to be expressive enough to describe interesting properties within the problem area. Moreover, the language has to be formal enough to be part of a formal verification method. In other words there should be a proper balance between expressiveness and formalism.

The natural way to develop specifications for a system is to develop partial specifications by developing sub-specifications of the system. As the development process progresses, more and more requirements evolve and these are conjuncted to the already existing requirements.

When considering reactive systems, an obvious specification language respecting the above requirements is *modal logic*. In this language it is simple to construct partial specifications, as conjunctions are part the logic language. Furthermore, modal logic specifications are loose in the sense that they allow for abstracting away unimportant details.

In this thesis we develop three kinds of modal logics to use for specifying about reliable real time systems. The three logics vary in their expressiveness and level of abstraction, which makes them useful in different kinds of settings.

We furthermore develop a model of reliable real time systems in which processes are modelled as Timed Probabilistic Graphs (TPG's) having a semantic description in terms of an extended notion of traditional labelled transition systems.

To verify whether a logical specification is satisfied by some model, is known as *model checking*. In this thesis we develop model checking algorithms for specifications in two of the developed logics with respect to implementations in terms of TPG's.

1.3 Construction

One way to develop reliable real time systems is by making a logical specification of the system and then have programmers implement the system. But programmers will often make mistakes when programming the system. Of course we can use the verification approach to ensure correctness of the system, but having an algorithm which given a specification will be able to automatically construct a satisfying model is obviously a big advantage. In this way the programmers concern will only be to make the correct specifications.

In this thesis we develop an algorithm for model construction, given a logical specification in *one* of the developed specification formalisms.

1.4 Related work

Our model of reliable real time systems in the form of TPG's can be seen as a probabilistic extension of *Timed Automata* as described by Alur and Dill [AD90]. The probabilistic extension is performed using a *generative* semantic of probabilities. In [vGSST90] probabilistic models are categorized in the three main groups: *generative*, *reactive* and *stratified* models. In the generative approach the probability distribution is on the total set of actions possible in some state, whereas in the reactive approach a probability distribution is considered for *each* action. This last approach is a natural extension of Milners notion "button-pushing"-experiment and the approach is e.g. used by Larsen and Skou in [LS89].

Our modal logic specification formalisms can all be seen as extensions of the modal logic HML [Mil89]. The extensions with respect to probabilistic modalities is inspired by the approach in [LS89] although we work upon a generative semantic model.

Different logics are developed in which it is possible to express time properties. In [EMSS89] CTL (Computation Tree Logic) is extended with the possibility to express properties of quantitative discrete time. Also Alur, Courcoubetis and Dill [ACD90] extends CTL with time properties, but in their logic TCTL formulas are interpreted over models with continouos time domain. In [Han91] a logic including both probabilities and quantitative time is presented. In this logic formulas are interpreted over a discrete time domain.

The logics we develop in this thesis are interpreted over a real time domain and we are able to express quantitative time properties as the ones presented in both [ACD90] and [Han91]. Furthermore, using an approach similar to the one described in [LLW95] we introduce a notion of *formula clocks* which enables a generalized interval time logic.

The model checking approach used in the thesis is a proof systematic approach using the *region technique* of Alur and Dill [ACD90] to obtain a finite representation of the real time domain.

The method used for model construction is based on traditional tableau methods applied for modal logic [HC68].

1.5 Thesis outline

In Chapter 2 we present our formal semantic model for reliable real time systems. This is the notion of timed probabilistic transition systems. Furthermore, we define the syntax of timed probabilistic graphs (TPG's) upon the basic model.

In Chapter 3 we present our three different modal logics used to specify properties of reliable real time systems.

In Chapter 4 we present our model checking algorithms for specifications in the developed logics and implementations in the form of TPG's. We have developed model checking algorithms for two of the three logics, whereas we present ideas on model checking the third logic.

In Chapter 5 we present our model construction algorithm for specifications in *one* of the developed logics. We show that a certain kind of bounded satisfiability is decidable.

In Chapter 6 we describe the construction of characteristic formulas for TPG's.

Chapter 2

Background

In this chapter we present the definitions that will be used throughout this thesis. The chapter is organized as follows. In Section 2.1 we introduce the notion of timed probabilistic transition systems as the fundamental model of reliable real time systems. In Section 2.2 we define the notion of timed probabilistic graphs which we use as the syntax of timed probabilistic processes.

2.1 Timed Probabilistic Transition Systems

Timed probabilistic transition systems are an extension of traditional labelled transition systems with respectively time and probabilities. The time extension is based on the same model-theoretic choices and assumptions that make up the semantics of Wang's TCCS [Yi91] and the basis of our probabilistic extension is a generative interpretation of probabilities.

In Section 2.1.1 we briefly introduce the most important design principles of the timed and probabilistic extensions, respectively. In Section 2.1.2 we define the notion of a timed probabilistic transition system and impose certain restrictions on the model to obtain several nice features. Finally, we define a notion of strong timed probabilistic bisimulation and weak timed probabilistic bisimulation between processes.

2.1.1 Fundamental design principles

The time extension of traditional labelled transition systems presented in this thesis follows the well known two-phase approach: Actions are considered to take no time and time is allowed to pass in between actions. Hence, we have two separate types of transitions according to actions and time respectively. Like Wang we consider a dense time domain; the positive reals.

We have two types of *events* that can cause transitions, namely *actions* and *delays*. The action set is partitioned into two disjoint sets, Act_{urg} and Act_{lazy}. In

TCCS Wang uses τ-actions and the maximal progress assumption in an essential manner in order to be able to express time-outs which cause the disabling of certain choices. As we wish to preserve this expressiveness, we keep the special status of urgent actions. Urgent actions compete with the remaining actions at the moment of their enabling, and beyond that point one of the urgent actions is forced to be taken. Unlike τ-actions we allow actions in Act_{urg} to be not just internal synchronization between pairs of processes.

Formally, the set of events is defined as follows.

Definition 2.1 *The set of events $\mathcal{M}$ is defined by $\mathcal{M} = Act \cup \mathcal{D}$ where*

- $Act = Act_{urg} \cup Act_{lazy}$ *is the set of actions.*
- $\mathcal{D} = \{\epsilon(d) | d \in \mathbf{R}^+\}$ *is the set of delays.*

Probabilistic models are extensions of traditional labelled transition systems with probabilities attached to the action transitions. The probabilistic extension that we consider is a *generative* extension. In [vGSST90] van Glabbeek et.al. categorize probabilistic models in three main groups. These are the *reactive, generative*, and *stratified* models.

In the *reactive* model the sum of probabilities of transitions from a given state labelled with *one* certain action is either 1 or 0. Hence, it is not possible to relate probabilities of transitions labelled with *different* actions. The basic idea is Milner's "button-pushing" experiment [Mil89] where an external observer of a system chooses to observe a certain action i.e. the probability distribution is on the different transitions labelled with exactly the chosen action.

The *generative* approach is a generalization of the reactive approach [vGSST90] where there is a probability distribution on the *complete set* of enabled actions in a given state.

Finally, the *stratified* model is a refined generative model that has a *branching structure* on the probabilistic transitions. In the generative model there is a flat structure on the probabilistic transitions. To obtain a relatively general and simple model we choose the generative approach in the following.

2.1.2 Timed probabilistic transition system

Having briefly discussed the basic design principles of the timed and probabilistic extensions we can now define the notion of a timed probabilistic transition system.

Definition 2.2 *A timed probabilistic transition system is a tuple*

$\langle S, s_0, \pi, \longrightarrow \rangle$

where

- *S is a set of states.*
- *$s_0 \in S$ is the initial state.*
- *$\pi : S \times Act \times S \to [0,1]$ is a probabilistic transition function, satisfying for each $s \in S$*

$$\sum_{a,s'} \pi(s,a,s') \in \{0,1\}$$

- *$\longrightarrow \subseteq S \times \mathcal{D} \times \mathcal{S}$ is a timed transition relation.*

Let Proc represent the set of all timed probabilistic processes.

Notation: For a probabilistic timed transition system as above whenever $\langle s,t,s'\rangle \in \longrightarrow$, we write $s \xrightarrow{\epsilon(t)} s'$. Similarly, whenever $\pi(s,a,s') = p$ for $p > 0$ we write $s \xrightarrow{a,p} s'$ or, sometimes, $s \xrightarrow{a} s'$ or $s \xrightarrow{a}$ when the precise value of p and/or s' is not important. For a set $S \subset Proc$ a state $s \in Proc$ and an action $a \in Act$, we let $\pi(s,a,S) = \sum_{s' \in S} \pi(s,a,s')$. We often write $s \xrightarrow{a} S$ when $\pi(s,a,S) > 0$. Similarly if $p \xrightarrow{\epsilon(t)} s'$ for some $s' \in S$ we write $p \xrightarrow{\epsilon(t)} S$. Finally, we let $\pi(s,a) = \sum_{s'} \pi(s,a,s')$.

We impose the following restrictions on the timed transition relation, thereby obtaining some nice features of our model.

- By requiring that if $s \not\xrightarrow{a}$ for any $a \in Act_{urg}$ then $s \xrightarrow{\epsilon(t)}$ we obtain *transition liveness*.
- By requiring that whenever $s \xrightarrow{a}$ for some $a \in Act_{urg}$ then $s \not\xrightarrow{\epsilon(t)}$ we obtain *maximal progress*.
- By requiring that whenever $s \xrightarrow{\epsilon(t)} s'$, and $s \xrightarrow{\epsilon(t)} s''$ then $s' = s''$ we obtain *time determinacy*.
- By requiring that whenever $s \xrightarrow{\epsilon(c+d)} s''$ then for some s', $s \xrightarrow{\epsilon(c)} s' \xrightarrow{\epsilon(d)} s''$ we obtain *time continuity*.

Our strongest notion of equivalence between timed probabilistic processes is that of strong timed probabilistic bisimulation. This equivalence can be seen as a timed probabilistic version of traditional strong bisimulation [Mil89, Par80].

Definition 2.3 *An equivalence relation $\mathcal{B} \subseteq Proc \times Proc$ is a strong timed probabilistic bisimulation whenever $(P,Q) \in \mathcal{B}$ implies that for all $S \in Proc/\mathcal{B}$*

- *for all $a \in Act$, $\pi(P, a, S) = \pi(Q, a, S)$ and*
- *for all $t \in \mathbf{R}_{\geq 0}$, $P \xrightarrow{\epsilon(t)} S$ iff $Q \xrightarrow{\epsilon(t)} S$*

Two processes P and Q are strong timed probabilistic bisimular, denoted $P \sim_{tp} Q$, iff there exists a strong timed probabilistic bisimulation containing the pair (P, Q).

Example 2.1 Below we describe two processes P and Q, which are strong timed probabilistic bisimular. $\mathcal{T}$ describes a partitioning in equivalence classes.

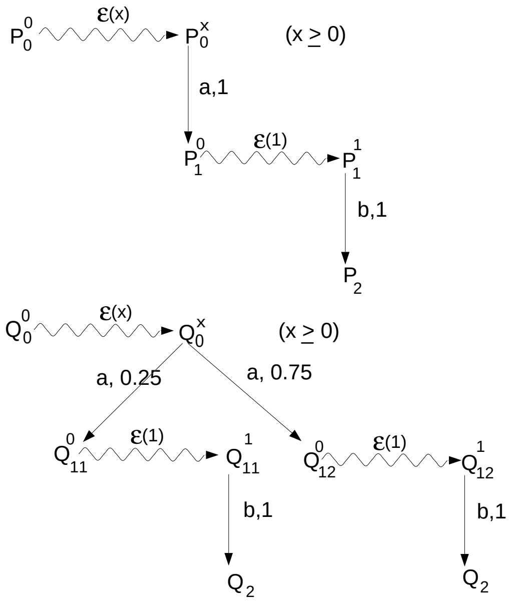


Figure 2.1: *This figure illustrates the behaviour of the processes P and Q.*

$$P = \langle S, s_0, \pi, \longrightarrow \rangle \text{ and } Q = \langle N, n_0, \pi, \longrightarrow \rangle$$

where:

$$\begin{array}{l} S = \{P_0^d, P_1^e, P_2 | d, e \in \mathbf{R}_{\geq 0}, 0 \leq e \leq 1\} \\ s_0 = P_0^0 \end{array}$$

$$\begin{array}{ll} \pi(P_0^d, a, P_1^0) = 1 & P_0^d \xrightarrow{\epsilon(t)} P_0^{d+t} \\ \pi(P_1^1, b, P_2) = 1 & P_1^e \xrightarrow{\epsilon(t)} P_1^{e+t}; e + t \leq 1 \\ & P_2 \xrightarrow{\epsilon(t)} P_2 \end{array}$$

and

$$
\begin{array}{ll}
N = \{Q_0^f, Q_{11}^g, Q_{12}^g, Q_2 | f, g \in \mathcal{R}_{\geq 0}, 0 \leq g \leq 1\} & \\
n_0 = Q_0^0 & \\
\pi(Q_0^f, a, Q_{11}^0) = 0.25 & Q_0^f \xrightarrow{\epsilon(t)} Q_0^{f+t} \\
\pi(Q_0^f, a, Q_{12}^0) = 0.75 & Q_{11}^g \xrightarrow{\epsilon(t)} Q_{11}^{g+t}; g + t \leq 1 \\
\pi(Q_{11}^1, b, Q_2) = 1 & Q_{12}^g \xrightarrow{\epsilon(t)} Q_{12}^{g+t}; g + t \leq 1 \\
\pi(Q_{12}^1, b, Q_2) = 1 & Q_2 \xrightarrow{\epsilon(t)} Q_2
\end{array}
$$

$$\mathcal{T} = \{(P_0^0, Q_0^0), (P_0^d, Q_0^d), (P_1^e, Q_{11}^e), (P_1^e, Q_{12}^e), (P_2, Q_2)\}$$

□

Similar to [Mil89], we can define a notion of weak bisimilarity between processes, where internal actions are abstracted away. As we have a generalized notion of urgent actions, we split this set of actions into two disjoint sets such that $Act_{urg} = Act_{urg}^{com} \cup Act_{urg}^{obs}$, where Act_{urg}^{com} represents internal communication and Act_{urg}^{obs} the observable urgent actions. We now define a notion of weak probabilistic transitions as follows.

Definition 2.4 *Let π be a probabilistic transition function. We define the τ-closure of π as $\pi^* : Proc \times Proc \longrightarrow [0, 1]$ as*

$$\pi^*(P, Q) = p \quad \mathit{if} \quad \sum_S \prod_{i=1}^{m-1} \pi(P_i^{(n)}, \tau, P_{i+1}^{(n)}) = p$$

where

$$
\begin{array}{rcl}
S & = & \{n, m, \{P_1^{(n)}, \ldots, P_m^{(n)}\} \mid P_1^{(n)} = P, P_m^{(n)} = Q, \\
 & & \forall i \in [1, .., m-1].\ \pi(P_i^{(n)}, \tau, P_{i+1}^{(n)}) \neq 0 \ \mathit{or} \ \pi(P_i^{(n)}, \epsilon, P_{i+1}^{(n)}) = 1\}
\end{array}
$$

Definition 2.5 *Let τ range over the actions in Act_{urg}^{com} and let $\alpha \in Act \backslash Act_{urg}^{com}$. We define a weak probabilistic transition function $\pi_w : Proc \times Act \times Proc \rightarrow [0, 1]$ as:*

$$
\begin{array}{ll}
1 & \pi_w(P, \tau, Q) = p \ \mathit{if} \ \pi^*(P, Q) = p \\
2 & \pi_w(P, \alpha, Q) = p \ \mathit{if} \sum_S \pi^*(P_1, R_1) \cdot \pi(R_1, \alpha, R_2) \cdot \pi^*(R_2, Q) = p
\end{array}
$$

where

$$S = \{(R_1, R_2) \mid \pi^*(P, R_1) \neq 0, \pi(R_1, \alpha, R_2) \neq 0, \pi^*(R_2, Q) \neq 0\}$$

NOTATION: Whenever $\pi_w(P, \tau, Q) = p$ or $\pi_w(P, \alpha, Q) = p'$ for $p, p' > 0$ we write $P \overset{\tau,p}{\Longrightarrow} Q$, $P \overset{\alpha,p'}{\Longrightarrow} Q$ respectively.

We define weak time transitions as follows:

Definition 2.6 *Let τ range over the actions in Act_{urg}^{com}. We define a weak timed transition function $\Longrightarrow \subseteq Proc \times (\mathcal{D} \cup [0,1]) \times Proc$ as:*
$P \overset{\epsilon(d),p}{\Longrightarrow} Q$ *if*

$$\begin{aligned} \textstyle\sum\{\!| \, p' \mid \quad & \exists n, \exists R_1, \ldots, R_n. \\ & P \overset{\tau,p_1}{\Longrightarrow} R_1 \overset{\epsilon(d_1)}{\longrightarrow} R_2 \overset{\tau,p_2}{\Longrightarrow} \cdots \overset{\tau,p_n-1}{\Longrightarrow} R_{n-1} \overset{\epsilon(d_{n-1})}{\longrightarrow} R_n \overset{\tau,p_n}{\Longrightarrow} Q \textit{ and} \\ & p' = \textstyle\prod_{j \leq n} p_i \textit{ and } d = \textstyle\sum_{i \leq n-1} d_i \, |\!\} = p \end{aligned}$$

Notation: For a set $S \subset Proc$ and $P \in Proc$ we write

$$P \overset{\epsilon(d),p}{\Longrightarrow} S \text{ if } \sum\{p' \mid P \overset{\epsilon(d),p'}{\Longrightarrow} Q \wedge Q \in S\} = p$$

Given the definitions of weak transitions, we now can define our notion of weak timed probabilistic bisimulation.

Definition 2.7 *An equivalence relation $\mathcal{S} \subseteq Proc \times Proc$ is a weak timed probabilistic bisimulation whenever $(P, Q) \in \mathcal{S}$ implies for all $S \in Proc/\mathcal{S}$, $\alpha \in Act \setminus Act_{urg}^{com}$ and $d \in \mathbf{R}_{\geq 0}$*

1. $\pi_w(P, \alpha, S) = \pi_w(Q, \alpha, S)$
2. $P \overset{\epsilon(d),p}{\Longrightarrow} S \Leftrightarrow Q \overset{\epsilon(d),p}{\Longrightarrow} S$

Two processes P and Q are weak timed probabilistic bisimular, denoted $P \approx_{tp} Q$, iff there exists a weak timed probabilistic bisimulation containing the pair (P, Q)

2.2 Timed Probabilistic Graphs

This section defines Timed Probabilistic Graphs, the syntax we have chosen for definition of timed probabilistic processes.

For modelling real time processes, Alur and Dill [AD90] have proposed timed automata. Timed graphs as presented in [God94] are a variant of these. In [God94] it has been shown that timed graphs provides a more expresive model than some proposed real time process algebras and therefore we use an extension of timed graphs as our syntactic notion of reliable real time processes.

A timed graph is a labelled transition graph associated with a finite set of clocks. Any edge of the graph is labelled with an *action* from an alphabet, an *enabling condition* which states when the action can be performed and a subset of the clocks to be reset when performing the action. Figure 2.2 gives an example of a timed graph with clock set $\{x, y\}$.

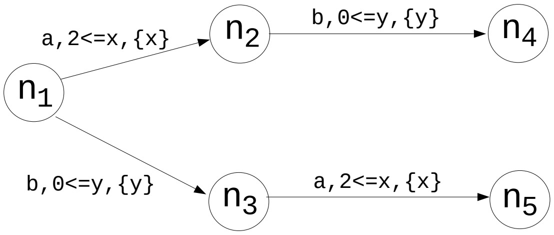


Figure 2.2: *A timed graph with clocks x and y*

2.2.1 Syntax and semantics

Timed probabilistic graphs (TPG's) can be seen as a probabilistic extension of timed graphs where edges, beside actions, are also labelled with weights. In the sequel we define the notion of timed probabilistic graphs.

Clocks of a timed probabilistic graph are associated with an assignment of real values. This assigment is called a *time assignment* and is defined as follows.

Definition 2.8 *For a finite set of clocks C, we define a time assignment $\gamma \in \mathbf{R}^C$, that for each $c \in C$ assigns a value $\gamma(c) \in \mathbf{R}_{\geq 0}$.*

Notation: For $\gamma, \gamma' \in \mathbf{R}^C$, and $C' \subseteq C$, $\gamma[C' \leftarrow \gamma']$ represents a time assignment $\gamma'' \in \mathbf{R}^{C''}$ such that

$$\gamma''(c) = \begin{cases} \gamma'(c) & \text{if } c \in C' \\ \gamma(c) & \text{otherwise} \end{cases}$$

If for all $c \in C$, $\gamma(c) = r$ we write r for the constant assignment. For $t \in \mathbf{R}_{\geq 0}$ og $\gamma \in \mathbf{R}^C$ we write $\gamma + t$ for the assignment $(\gamma + t)(c) = \gamma(c) + t$. Finally, we let $\lfloor \gamma(c) \rfloor$ represent the integral part of $c \in C$ and we let $\{\gamma(c)\}$ represent the fractional part of c.

To each action we associate a condition which states when it is legal to perform the action. It is legal to perform an action when the values of the clocks respect the specified condition on the clocks. These conditions are called *enabling conditions* and defined formally as follows.

Definition 2.9 *An enabling condition b over a set of clocks C is given by the following syntax*

$$b ::= c_1 + x \sim c_2 + y \mid b_1 \wedge b_2 \mid \neg b$$

where $c_1, c_2 \in C$; x,$y \in \mathbf{N}$ and $\sim \in \{\leq|<|\geq|>|=\}$. Let $\mathcal{B}(C)$ denote the set of enabling conditions over C. By the subset $\mathcal{B}'(C) \subset \mathcal{B}(C)$ we denote the set of enabling conditions over C where $\sim$ has been replaced by $\sim' \in \{\leq|\geq|=\}$.

In enabling conditions we only allow integer constants. This restriction is vital for later decidability results. We could have allowed rational constants but for the sake of simplicity we stick to the integers. The reason for only allowing non-strict comparison of clock values, $\sim'$ is that we need to ensure transition liveness when writing enabling conditions for urgent actions.

The enabling conditions can be seen as predicates on time assignments. Below the semantics of enabling conditions is given as the set of time assignments for which the condition holds.

Definition 2.10 *Every enabling condition b describes a set of* time assignments *satisfying b. This is described by the semantic function $[\![\ ]\!] : \mathcal{B}(C) \to 2^{\mathbf{R}^C}$ defined by*

$$\begin{array}{lcl} [\![c_1 + x \sim c_2 + y]\!] & = & \{\gamma \mid \gamma(c_1) + x \sim \gamma(c_2) + y\} \\ [\![b_1 \wedge b_2]\!] & = & [\![b_1]\!] \cap [\![b_2]\!] \\ [\![\neg b]\!] & = & \mathbf{R}^{\mathbf{C}} \setminus [\![b]\!] \end{array}$$

Notation: We write $b(\gamma)$ meaning $\gamma \in [\![b]\!]$. Similarly we write $\neg b(\gamma)$ whenever $\gamma \notin [\![b]\!]$.

In enabling conditions associated with a TPG there is a maximal constant.

Definition 2.11 *For an enabling condition b we let $max(b)$ denote the* maximal constant *in b,*

$$\begin{array}{lcl} max(c_1 + x \sim c_2 + y) & = & max\{x, y\} \\ max(b_1 \wedge b_2) & = & max\{max(b_1), max(b_2)\} \\ max(\neg b) & = & max(b) \end{array}$$

For A, a finite set of enabling conditions we define the maximal constant as

$$max(A) = max\{max(b) \mid b \in A\}$$

The extension of timed graphs with respect to probabilities can be done either by assigning weights to the actions or by assigning probabilities to the actions. If we assign weights we do not have to ensure that they normalize to the value 1 at all times. This would be the case if we chose to assign probabilities to the actions.

We choose the first approach, viz. to extend with respect to probabilities by assigning to each action, a weight which we can interpret as a probability via the notion of normalization. Now given all of the above, we can define TPG's as follows.

Definition 2.12 *A timed probabilistic graph (TPG) is a tuple $\langle N, n_0, C, \longrightarrow \rangle$ where*

- *N is a finite set of nodes.*
- *$n_0 \in N$ is the initial node.*

- *C is a finite set of clocks.*
- $\longrightarrow \subseteq (N \times Act_{lazy} \times \mathbf{N} \times \mathcal{B}(C) \times 2^C \times N) \cup (N \times Act_{urg} \times \mathbf{N} \times \mathcal{B}'(C) \times 2^C \times N)$ *is a weighted timed transition function.*

Notation: For $(n, a, w, b, C', m) \in \longrightarrow$ we often write $n \overset{a,w,b,C'}{\longrightarrow} m$, where

- a denotes an action causing the transition.
- w is the weight of the transition.
- b is the enabling condition of the transition.
- C' is a finite set of clocks to be reset concurrent with the transition.

Finally, $W_{n,\gamma} \overset{\text{def}}{=} \sum\{\!|\ w \mid \exists a', b, r, m.\ n \overset{a',w,b,r}{\longrightarrow} m \wedge b(\gamma)\ |\!\}$ denotes the total weight of edges from n enabled at γ.

Figure 2.3 gives an example of a TPG.

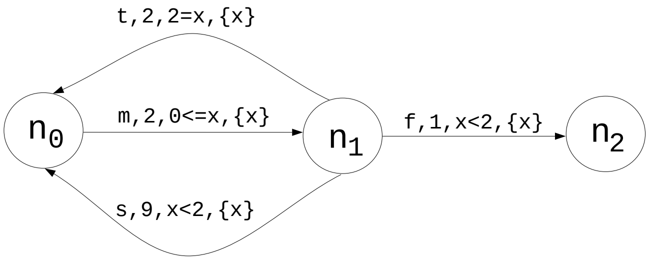


Figure 2.3: *A TPG, describing a faulty medium. The medium can receive a message (m) with weight 2, thereby reaching a state where it can either send the message (s) to an receiver with weight 9 or fail with weight 1. The medium may also timeout (t) with weight 2, if it has not failed or send the message within 2 time units.*

Informally, the interpretation of a TPG is as follows: The TPG is started in the initial node with all the clocks set to zero. All the clocks start at the same time and progress with the same speed. If an enabling condition on an edge holds, the action described by this edge can be performed and the TPG thereby changes state. The execution of the action takes no time and simultaneously with the execution of the action the clocks mentioned in the set C' are reset.

The formal semantics of a TPG is described by a timed probabilistic transition system known as the *Configuration Graph* of the TPG.

The states of the configuration graph, also called configurations, are the nodes of the TPG together with particular time assignments. The initial node will be the initial node of the TPG associated with the time assigment assigning

0 to all clocks in the clock set. Because of the dense time domain, there are infinitely many time assignments associated with each node of the TPG, so the state space of the configuration graph will be infinite.

The probabilistic transition function describes the probabilities of individual actions. As the edges of the TPG are labelled with weights and not probabilities, a normalization is necessary. The probability of reaching a state $\langle m, \gamma' \rangle$ from another state $\langle n, \gamma \rangle$ by performing an a-action is found as follows: We sum the weights on edges from n to m labelled with an a-action enabled at γ and where γ' is the time assignment obtained by resetting the clocks in the reset set. This sum is then normalized with respect to $W_{n,\gamma}$.

Assuming the TPG of figure 2.3 is at state $\langle n_1, x = 0 \rangle$ the probability of an s-action is 0.9.

Now, the formal definition of the configuration graph is as follows.

Definition 2.13 *Let $G = \langle N, n_0, C, \longrightarrow \rangle$ be a TPG. Then its Configuration Graph, $\mathcal{C}[\![G]\!]$ is a timed probabilistic transition system with*

- *state set $N \times \mathbf{R}^C_{\geq 0}$.*
- *initial state $\langle n_0, \mathbf{0} \rangle$ where $\mathbf{0} = \lambda C.0$*
- *probabilistic transition function*

  $$\pi : (N \times \mathbf{R}^C_{\geq 0}) \times Act \times (N \times \mathbf{R}^C_{\geq 0}) \rightarrow [0,1]$$

  defined for $a \in Act$ by

  $$\pi(\langle n, \gamma \rangle, a, \langle m, \gamma' \rangle \overset{\text{def}}{=} \begin{cases} \frac{\sum \{\!| w \mid \exists b, C'. n \overset{a,w,b,C'}{\longrightarrow} m \wedge b(\gamma) \wedge \gamma' = \gamma[C' \leftarrow 0] |\!\}}{W_{n,\gamma}}; & \textit{if } W_{n,\gamma} \neq 0 \\ 0; & \textit{otherwise} \end{cases}$$

- *timed transition relation $\longrightarrow \subseteq (N \times \mathbf{R}^C_{\geq 0}) \times (N \times \mathbf{R}^C_{\geq 0})$ defined by*

  $$\langle n, \gamma \rangle \xrightarrow{\epsilon(t)} \langle n, \gamma' \rangle$$

  where $\gamma' = \gamma + t$, and for all $t' < t$ and $a \in Act_{urg}$, $\pi(\langle n, \gamma + t' \rangle, a) = 0$.

The following defines strong timed probabilistic bisimulation on TPG's.

Definition 2.14 *Two TPG's G and H are said to be strong timed probabilistic bisimular iff their configuration graphs are; i.e.*

$$G \sim_{tp} H \quad \textit{iff} \quad \mathcal{C}[\![G]\!] \sim_{tp} \mathcal{C}[\![H]\!]$$

2.2.2 Properties of TPG's

In this section we show that TPG's preserve the properties described in Section 2.1.2.

Theorem 2.1 (time determinancy)

Let $G = \langle N, n_0, C, \longrightarrow \rangle$ be a TPG and let n be a node in G. Then

$$\langle n, \gamma \rangle \xrightarrow{\epsilon(t)} \langle n, \gamma' \rangle \text{ and } \langle n, \gamma \rangle \xrightarrow{\epsilon(t)} \langle n, \gamma'' \rangle \text{ implies } \gamma' = \gamma''(= \gamma + t)$$

Proof: Follows directly as a consequence of the definition of the timed transition relation in Definition 2.13. □

Theorem 2.2 (maximal progress)

Let $G = \langle N, n_0, C, \longrightarrow \rangle$ be a TPG and let n be a node in G. Then

$$\langle n, \gamma \rangle \xrightarrow{a,p} \text{ for } a \in Act_{urg} \text{ implies } \forall y > 0.\, \langle n, \gamma \rangle \overset{\epsilon(y)}{\not\longrightarrow}$$

Proof: As a consequence of the definition of $\xrightarrow{\epsilon(t)}$, a timed transition is only possible if no *urgent* actions are enabled. □

Theorem 2.3 (transition liveness)

Let $G = \langle N, n_0, C, \longrightarrow \rangle$ be a TPG and let n be a node in G. Then

$$\forall a \in Act_{urg}.\, \langle n, \gamma \rangle \overset{a,p}{\not\longrightarrow} \text{ implies } \exists t.\, \langle n, \gamma \rangle \xrightarrow{\epsilon(t)}$$

Proof: Follows from Definition 2.13. □

Theorem 2.4 (time continuity)

Let $G = \langle N, n_0, C, \longrightarrow \rangle$ be a TPG and let n be a node in G. Then

$$\langle n, \gamma \rangle \xrightarrow{\epsilon(t+d)} \langle n, \gamma'' \rangle \text{ iff } \exists \gamma'.\langle n, \gamma \rangle \xrightarrow{\epsilon(t)} \langle n, \gamma' \rangle \xrightarrow{\epsilon(d)} \langle n, \gamma'' \rangle$$

Proof: We only show one direction, the other is equally simple. Suppose that $\langle n, \gamma \rangle \xrightarrow{\epsilon(t+d)} \langle n, \gamma'' \rangle$ for some t and d. Then according to Definition 2.13, $\forall r < t + d.\pi(\langle n, \gamma + r, a) = 0$. Hence, $\forall r < t.\, \pi(\langle n, \gamma + r, a) = 0$ and $\forall r' < d.\pi(\langle n, \gamma + t + r', a) = 0$. From Definition 2.13 we then have that $\langle n, \gamma \rangle \xrightarrow{\epsilon(t)} \langle n, \gamma + t \rangle \xrightarrow{\epsilon(d)} \langle n, \gamma + t + d \rangle$ □

Chapter 3

Real Timed Probabilistic Logic

In this chapter we define three different logics for expressing properties about reliable real time systems in the form of real time probabilistic transition systems. These logics are called *Real Time Probabilistic Logic* (RTPL), RTPL_{weak} and RTPL_{until}, respectively. The logic RTPL is an extension of traditional HML [HM85] extended with time and probability operators. The timing extensions are based on the operators defined by Larsen, Laroussinie and Weise in [LLW95] where formulas can contain multiple formula clocks. In the logic RTPL_{weak} we replace the strong action and delay modalities of RTPL with weak versions. That is we allow τ-abstraction. Finally in the logic RTPL_{until} we define operators similar to the *until* operators defined in CTL [CES83] but extended with quantitative time and probabilities.

In Section 3.1 we define the syntax and semantics for RTPL. We introduce the notion of *extended configuration graphs*, and interpret formulas over these. We also prove that RTPL characterizes strong timed probabilistic bisimulation. In Section 3.2 we define the syntax for RTPL_{weak} and give the semantics for the new operators. We prove that RTPL_{weak} characterizes weak timed probabilistic bisimulation. In Section 3.3 we extend RTPL to RTPL_{until}. Furthermore we discuss some of the interesting properties which we are able to express in RTPL_{until}. Finally, we show that RTPL_{until} characterizes strong timed probabilistic bisimulation.

3.1 The logic RTPL

In the section we define the syntax and semantics of the logic RTPL.

Definition 3.1 *Let K be a finite set of clocks and k an integer. Furthermore let $p \in \mathbf{Q}_{\geq 0}$ be a probability parameter. Then* RTPL *is the set of formulas over*

K and k, given by the abstract syntax :

$$\begin{aligned} F \quad ::= \quad & tt \mid \neg F \mid F_1 \wedge F_2 \mid \langle\alpha\rangle_{\geq p} F \mid \langle\alpha\rangle_{>p} F \mid \forall\!\!\!\forall F \mid c_1 \text{ in } F \\ & \mid c_1 + x \sim c_2 + y \end{aligned}$$

where $\alpha \in Act$; $c_1, c_2 \in K$; $x, y \in \{0, 1, \ldots, k\}$; $\sim \in \{=, \leq, \geq, <, >\}$ and $p \in [0, 1]$.

Notation: We will use the following abbreviations:

$$\begin{aligned} & ff \equiv \neg tt,\ F_1 \vee F_2 \equiv \neg(\neg F_1 \wedge \neg F_2),\ [\alpha]F \equiv \neg\langle\alpha\rangle_{>0}\neg F,\ \exists\!\!\!\exists F \equiv \neg\forall\!\!\!\forall\neg F, \\ & F_1 \Rightarrow F_2 \equiv \neg F_1 \vee F_2 \end{aligned}$$

A formula F is said to be *closed* if every formula clock c occuring in F is in the scope of an "c in" operator. In the sequel we will *only* be considering closed formulas.

Formula clocks are abstract clocks that are part of the general language (RTPL) independent of the specific automata in concern. They can be thought of as "stop watches" used by an observer of a system to measure if certain time properties are satisfied. The stop watches are independent of the system being observed and can be reset at any time by the observer. Because of this, we have to interpret RTPL formulas over configuration graphs extended with the relevant formula clocks. We denote these structures as *extended configuration graphs.*

Definition 3.2 *Let $G = \langle N, n_0, C, \longrightarrow\rangle$ be a TPG and let $\mathcal{C}[\![G]\!] = \langle S, s_0, \pi, \longrightarrow\rangle$ be the configuration graph for G. Consider a formula clock set, K. Let $\mu \in \mathbf{R}^K_{\geq 0}$ be a time assignment on the set of formula clocks and let $C^+ = C \cup K$. The extended configuration graph $E\mathcal{C}[\![G, K]\!]$ for G with respect to K is a timed probabilistic transition system defined as $E\mathcal{C}[\![G, K]\!] = \langle N^+, n_0^+, \pi^+, \longrightarrow^+\rangle$ where*

- *N^+ is the state set, $N^+ = N \times \mathbf{R}^{C+}_{\geq 0}$*
- *$n_0^+ = \langle n_0, \mathbf{0}\rangle^+$ where $\mathbf{0} = \lambda C^+.0$ is the initial state.*
- *π^+ is the probabilistic transition function $\pi^+ : N^+ \times Act \times N^+ \to [0, 1]$ defined for $a \in Act$ by*

$$\forall \mu \in \mathbf{R}^K.\, \pi^+(\langle n, \gamma\mu\rangle^+, a, \langle n', \gamma'\mu\rangle^+) = p \ \textit{ iff } \ \pi(\langle n, \gamma\rangle, a, \langle n', \gamma'\rangle) = p$$

- *$\longrightarrow^+$ is a timed transition relation $\longrightarrow^+ \subseteq N^+ \times N^+$ defined by*

$$\forall \mu \in \mathbf{R}^K.\, \langle n, \gamma\mu\rangle^+ \xrightarrow{\epsilon(d)} \langle n, \gamma + d\mu + d\rangle^+,\ d \in \mathbf{R}_{\geq 0} \ \textit{ iff } \ \langle n, \gamma\rangle \xrightarrow{\epsilon(d)} \langle n, \gamma + d\rangle$$

Notation: Whenever $\pi^+(\langle n, \gamma\mu\rangle^+, a, \langle n', \gamma'\mu\rangle^+) = p$ for $p > 0$ we write $\langle n, \gamma\mu\rangle^+ \overset{a,p}{\longrightarrow}{}^+ \langle n', \gamma'\mu\rangle^+$

The formula (c_1 in F) introduces a formula clock c_1 and initializes it to 0; i.e. an extended state satisfies the formula in case the modified state with c_1 being reset to 0 satisfies F. Introduced formula clocks occur in formulas of the type $(c_1 + x \sim c_2 + y)$, which are satisfied by an extended state, provided the values of the named fomula clocks satisfy the required relationship. The other important connectives in RTPL are the *modalities*. Informally, $\forall\!\!\!\forall F$ holds in an extended state if all delay transitions lead to an extended state satisfying F. Thus $\forall\!\!\!\forall$ denotes universal quantification over delay transitions. Whereas the abbreviation $\exists\!\!\!\exists$ denotes existential quantification over delay transitions. The $\langle\alpha\rangle_{\geq p}F$ operator expresses the possibility of performing an α action with probability $\geq p$. Similarly $\langle\alpha\rangle_{>p}F$. Whereas $[\alpha]F$, as mentioned earlier, is an abbreviation for $\neg\langle\alpha\rangle_{>0}\neg F$. Now we can define the semantics of RTPL.

Definition 3.3 *Let $G = \langle N, n_0, C, \longrightarrow\rangle$ be a TPG. Let $\langle n, \gamma\mu\rangle^+$ be a state in a extended configuration graph of G. Let $\sqsupseteq$ denote either $\geq$ or $>$. We now define the satisfaction relation $\models$ as follows.*

$$\begin{array}{lll}
\langle n, \gamma\mu\rangle^+ \models tt & & \\
\langle n, \gamma\mu\rangle^+ \models \neg F & \overset{\text{def}}{\Leftrightarrow} & \langle n, \gamma\mu\rangle^+ \not\models F \\
\langle n, \gamma\mu\rangle^+ \models F \wedge G & \overset{\text{def}}{\Leftrightarrow} & \langle n, \gamma\mu\rangle^+ \models F \textit{ and } \langle n, \gamma\mu\rangle^+ \models G \\
\langle n, \gamma\mu\rangle^+ \models \langle\alpha\rangle_{\sqsupseteq p}F & \overset{\text{def}}{\Leftrightarrow} & \sum\{\!|\ \pi^+(\langle n, \gamma\mu\rangle^+, \alpha, \langle n', \gamma'\mu\rangle^+) \mid \langle n', \gamma'\mu\rangle^+ \models F\ |\!\} \sqsupseteq p \\
\langle n, \gamma\mu\rangle^+ \models \forall\!\!\!\forall F & \overset{\text{def}}{\Leftrightarrow} & \forall d \in \mathbf{R}_{\geq 0}. \langle n, \gamma + d\mu + d\rangle^+ \models F \\
\langle n, \gamma\mu\rangle^+ \models c_1 \textbf{ in } F & \overset{\text{def}}{\Leftrightarrow} & \langle n, \gamma\mu'\rangle^+ \models F \textit{ where } \mu' = \mu[c_1 \leftarrow 0] \\
\langle n, \gamma\mu\rangle^+ \models c_1 + x \sim c_2 + y & \overset{\text{def}}{\Leftrightarrow} & \mu(c_1) + x \sim \mu(c_2) + y
\end{array}$$

We say that a TPG, G satisfies a closed RTPL formula F and write $G \models F$ when $\langle n_0, \gamma_0\mu\rangle^+ \models F$ for any μ.

From RTPL we can obtain traditional real time interval modalities as presented in [JG95] as derived operators. The derived operators are defined as

$$\begin{array}{ll}
\exists[l, u]\ F \overset{\text{def}}{=} x \text{ in } \exists\!\!\!\exists((x \geq l \wedge x \leq u) \wedge F) & \forall[l, u]\ F \overset{\text{def}}{=} x \text{ in } \forall\!\!\!\forall((x \geq l \wedge x \leq u) \Rightarrow F) \\
\exists]l, u]\ F \overset{\text{def}}{=} x \text{ in } \exists\!\!\!\exists((x > l \wedge x \leq u) \wedge F) & \forall[l, u]\ F \overset{\text{def}}{=} x \text{ in } \forall\!\!\!\forall((x > l \wedge x \leq u) \Rightarrow F) \\
\exists[l, u[\ F \overset{\text{def}}{=} x \text{ in } \exists\!\!\!\exists((x \geq l \wedge x < u) \wedge F) & \forall[l, u]\ F \overset{\text{def}}{=} x \text{ in } \forall\!\!\!\forall((x \geq l \wedge x < u) \Rightarrow F) \\
\exists]l, u[\ F \overset{\text{def}}{=} x \text{ in } \exists\!\!\!\exists((x > l \wedge x < u) \wedge F) & \forall[l, u]\ F \overset{\text{def}}{=} x \text{ in } \forall\!\!\!\forall((x > l \wedge x < u) \Rightarrow F)
\end{array}$$

Example 3.1 The TPG G shown in Figure 3.1 will satisfy the following formula F:

$$F = c_1 \text{ in } (c_2 \text{ in } (0 \leq c_1 \wedge \langle a\rangle_{\geq 1}\ (0 \leq c_2 \wedge 2 > c_2 \wedge \langle b\rangle_{\geq 1} tt)))$$

□

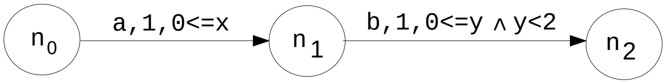


Figure 3.1: *TPG G satisfying the formula F.*

Theorem 3.1 *Two timed probabilistic processes* $\langle n, \gamma\mu\rangle^+$, $\langle m, \eta\mu\rangle^+$ *are strong timed probabilistic bisimular iff no formula* $F \in RTPL$ *with formula clock set* K *can distinguish between them. That is* $\forall \mu \in \mathbf{R}^K$

$$\langle n, \gamma\mu\rangle^+ \sim_{tp} \langle m, \eta\mu\rangle^+ \Leftrightarrow (\forall F \in \text{RTPL}. \langle n, \gamma\mu\rangle^+ \models F \Leftrightarrow \langle m, \eta\mu\rangle^+ \models F)$$

Proof:
$\Rightarrow$: It will be enough to show that: $\langle n, \gamma\mu\rangle^+ \sim_{tp} \langle m, \eta\mu\rangle^+ \wedge \langle n, \gamma\mu\rangle^+ \models F \Rightarrow \langle m, \eta\mu\rangle^+ \models F$. We proceed by induction in the structure of the formulas with the following IH: $\langle n', \gamma'\mu\rangle^+ \sim_{tp} \langle m', \eta'\mu\rangle^+ \wedge \langle n', \gamma'\mu\rangle^+ \models F' \Rightarrow \langle m', \eta'\mu\rangle^+ \models F'$ where F' is a subformula of F.

Cases:
$\underline{F = tt}$: Trivial since all processes satisfy tt.

$\underline{F = \neg F'}$: Assume $\langle n, \gamma\mu\rangle^+ \sim_{tp} \langle m, \eta\mu\rangle^+$ and $\langle n, \gamma\mu\rangle^+ \models F$. According to the definition of $\models$ we then have that $\langle n, \gamma\mu\rangle^+ \not\models F'$. Since F' is a subformula F then by IH we have that $\langle m, \eta\mu\rangle^+ \not\models F'$. Then by definition of $\models$ we have that $\langle m, \eta\mu\rangle^+ \models F$.

$\underline{F = F_1 \wedge F_2}$: Assume $\langle n, \gamma\mu\rangle^+ \sim_{tp} \langle m, \eta\mu\rangle^+$ and $\langle n, \gamma\mu\rangle^+ \models F$. Then by definition of $\models$ we have that $\langle n, \gamma\mu\rangle^+ \models F_1$ and $\langle n, \gamma\mu\rangle^+ \models F_2$. Since F_1 and F_2 both are subformulas of F by IH we get $\langle m, \eta\mu\rangle^+ \models F_1$ and $\langle m, \eta\mu\rangle^+ \models F_2$ and the definition of $\models$ now gives us $\langle m, \eta\mu\rangle^+ \models F_1 \wedge F_2$.

$\underline{F = \langle a\rangle_{\geq p} F'}$: Assume $\langle n, \gamma\mu\rangle^+ \sim_{tp} \langle m, \eta\mu\rangle^+$ and $\langle n, \gamma\mu\rangle^+ \models F$. Let $S = \{\langle n', \gamma'\mu\rangle^+ \mid \langle n, \gamma\mu\rangle^+ \overset{a,p}{\longrightarrow}{}^+ \langle n', \gamma'\mu\rangle^+ \wedge \langle n', \gamma'\mu\rangle^+ \models F'\}$. Then, by definition of $\models$, we now have $\pi^+(\langle n, \gamma\mu\rangle^+, a, S) \geq p$, where $p > 0$. Let $S' = \cup\{T \in Proc/\sim_{tp} \mid T \cap S \neq \emptyset\}$. We know that $\forall \langle n'', \gamma''\mu\rangle^+ \in S'. \exists \langle n', \gamma'\mu\rangle^+ \in S. \langle n'', \gamma''\mu\rangle^+ \sim_{tp} \langle n', \gamma'\mu\rangle^+$. By IH this implies that $\forall \langle n'', \gamma''\mu\rangle^+ \in S'. \langle n'', \gamma''\mu\rangle^+ \models F'$. Since $S \subseteq S'$ we get : $\pi^+(\langle n, \gamma\mu\rangle^+, a, S') \geq \pi^+(\langle n, \gamma\mu\rangle^+, a, S) \geq p$. Because $\langle n, \gamma\mu\rangle^+ \sim_{tp} \langle m, \eta\mu\rangle^+$ we know by definition of $\sim_{tp}$ that $\pi^+(\langle n, \gamma\mu\rangle^+, a, S) = \pi^+(\langle m, \eta\mu\rangle^+, a, S)$ and by definition of $\models$ we must have $\langle m, \eta\mu\rangle^+ \models \langle a\rangle_{\geq p} F'$

$\underline{F = \langle a\rangle_{>p} F'}$: As for $F = \langle a\rangle_{\geq p} F'$

$\underline{F = \forall\!\!\!\forall F'}$: Assume that $\langle n, \gamma\mu\rangle^+ \sim_{tp} \langle m, \eta\mu\rangle^+$ and $\langle n, \gamma\mu\rangle^+ \models F$. We define $S = \{\langle n, \gamma'\mu'\rangle^+ \mid \exists d \in \mathbf{R}_{\geq 0}. \langle n, \gamma\mu\rangle^+ \overset{\epsilon(d)}{\longrightarrow}{}^+ \langle n, \gamma'\mu'\rangle^+\}$. According to

the definition of $\models$ we know that $\forall\langle n, \gamma'\mu'\rangle^+ \in S.\langle n, \gamma'\mu'\rangle^+ \models F'$. Now let $S' = \cup\{T \in Proc/\!\sim_{tp} |\; T \cap S \neq \emptyset\}$.
$(*)$ We know that $\forall\langle n, \gamma''\mu''\rangle^+ \in S'.\exists\langle n, \gamma'\mu'\rangle^+ \in S.\langle n, \gamma''\mu''\rangle^+ \sim_{tp} \langle n, \gamma'\mu'\rangle^+$. Then by IH this implies that $\langle n, \gamma''\mu''\rangle^+ \models F'$. Since $S \subseteq S'$ this implies $\forall d \in \mathbf{R}_{\geq 0} : \langle n, \gamma\mu\rangle^+ \overset{\epsilon(d)}{\longrightarrow}^+ S'$ and because $\langle n, \gamma\mu\rangle^+ \sim_{tp} \langle m, \eta\mu\rangle^+$ then, according to the definition of $\sim_{tp}$ we have that $\forall T \in Proc/\!\sim_{tp} .\forall d \in \mathbf{R}_{\geq 0}.\langle n, \gamma\mu\rangle^+ \overset{\epsilon(d)}{\longrightarrow}^+ T \Leftrightarrow \langle m, \eta\mu\rangle^+ \overset{\epsilon(d)}{\longrightarrow}^+ T$ and then we have $\forall d \in \mathbf{R}_{\geq 0}.\langle m, \eta\mu\rangle^+ \overset{\epsilon(d)}{\longrightarrow}^+ S'$. This together with $(*)$ gives that $\langle m, \eta\mu\rangle^+ \models \forall F'$

$F = c_1$ in F' : Assume that $\langle n, \gamma\mu\rangle^+ \sim_{tp} \langle m, \eta\mu\rangle^+$ and $\langle n, \gamma\mu\rangle^+ \models F$. Then by definition of $\models$ we have that $\langle n, \gamma\mu'\rangle^+ \models F'$ where $\mu' = \mu[c_1 \leftarrow 0]$. Since F' is a subformula of F then by IH we have: $\langle m, \eta\mu'\rangle^+ \models F'$ with $\mu' = \mu[c_1 \leftarrow 0]$. Then, by definition of $\models$ we must have that $\langle m, \eta\mu\rangle^+ \models F$.

$F = c_1 + x \sim c_2 + y$: Trival since we only consider closed formulaes.

$\Leftarrow$: We have to show that $\mathcal{B} = \{(\langle n, \gamma\mu\rangle^+, \langle m, \eta\mu\rangle^+) \mid \forall F \in RTPL.\langle n, \gamma\mu\rangle^+ \models F \Leftrightarrow \langle m, \eta\mu\rangle^+ \models F\}$ is a timed probabilistic bisimulation. Consider an arbitrary pair $(\langle n, \gamma\mu\rangle^+, \langle m, \eta\mu\rangle^+) \in \mathcal{B}$ then by the definition of $\sim_{tp}$ we have to consider two cases.
1: Let $S_0 \in Proc/\mathcal{B}$ and let $\pi^+(\langle n, \gamma\mu\rangle^+, a, S_0) = p_0$.
Furthermore let $\langle m, \eta\mu\rangle^+ \overset{a,p_1}{\longrightarrow}^+ \langle m'\eta'\mu\rangle^+_1, \ldots, \langle m, \eta\mu\rangle^+ \overset{a,p_k}{\longrightarrow}^+ \langle m', \eta'\mu\rangle^+_k$ be the a-transitions of $\langle m, \eta\mu\rangle^+$.

The a-derivatives of $\langle m, \eta\mu\rangle^+$ can be ordered into sets s.t. $\langle m', \eta'\mu\rangle^+_1, \ldots, \langle m', \eta'\mu\rangle^+_i \in S_0$ and $\langle m', \eta'\mu\rangle^+_{i+1}, \ldots, \langle m', \eta'\mu\rangle^+_k \notin S_0$. Then we have

$$\begin{array}{lll} \langle m', \eta'\mu\rangle^+_{i+1} \notin S_0 & \Rightarrow & \exists F_{i+1}.\forall\langle n', \gamma'\mu\rangle^+ \in S_0.\ \langle n', \gamma'\mu\rangle^+ \models F_{i+1} \wedge \langle m', \eta'\mu\rangle^+_{i+1} \not\models F_{i+1} \\ \vdots & & \vdots \\ \langle m', \eta'\mu\rangle^+_k \notin S_0 & \Rightarrow & \exists F_k.\forall\langle n', \gamma'\mu\rangle^+ \in S_0.\ \langle n', \gamma'\mu\rangle^+ \models F_k \wedge \langle m', \eta'\mu\rangle^+_k \not\models F_k \end{array}$$

that is, we can find formulas $F_{i+1}, \ldots, F_k$ s.t. $\forall\langle n', \gamma'\mu\rangle^+ \in S_0.\ \langle n', \gamma'\mu\rangle^+ \models F_j$ but $\langle m', \eta'\mu\rangle^+_j \not\models F_j$ for $i + 1 \leq j \leq k$. Now assume $\sum_{j=1}^{i} p_j < p_0$ where $\pi^+(\langle m, \eta\mu\rangle^+, a, \langle m, \eta\mu\rangle^+_j) = p_j$. Then $\langle n, \gamma\mu\rangle^+ \models \langle a\rangle_{\geq p_0} \bigwedge_{j=i+1}^{k} F_j$ but $\langle m, \eta\mu\rangle^+ \not\models \langle a\rangle_{\geq p_0} \bigwedge_{j=i+1}^{k} F_j$ contradicting that $(\langle n, \gamma\mu\rangle^+, \langle m, \eta\mu\rangle^+) \in \mathcal{B}$. Thus $p_0 \leq \sum_{j=1}^{i} p_i$ and therefore $\pi^+(\langle m, \eta\mu\rangle^+, a, S_0) = p'$ with $p' \geq \sum_{j=1}^{i} p_i \geq p_0$. By symmetry of $\mathcal{B}$ it follows that $p_0 \geq p'$ hence $p_0 = p'$.

2: Let $S_0 \in Proc/\mathcal{B}$ and let $\langle n, \gamma\mu\rangle^+ \overset{\epsilon(d)}{\longrightarrow}^+ S_0$.
We will now show that $\langle m, \eta\mu\rangle^+ \overset{\epsilon(d)}{\longrightarrow}^+ S_0$.
1 : Assume $\langle m, \eta\mu\rangle^+ \overset{\epsilon(d)}{\not\longrightarrow}^+$ then we have for some atomic formula $(c_1 + x \sim c_2 + y)$ that $\langle n, \gamma\mu\rangle^+ \models \exists(c_1 + x \sim c_2 + y)$ and $\langle m, \eta\mu\rangle^+ \not\models \exists(c_1 + x \sim c_2 + y)$. This contradicts $(\langle n, \gamma\mu\rangle^+, \langle m, \eta\mu\rangle^+) \in \mathcal{B}$ which we assumed.

2 : Assume $\langle m, \eta\mu\rangle^+ \xrightarrow{\epsilon(d)}{}^+ \langle m, \eta + d\mu + d\rangle^+$ where $\langle m, \eta + d\mu + d\rangle^+ \notin S_0$ ($\langle m, \eta + d\mu + d\rangle^+$ is unique due to time determinism); then there exists F' such that $\forall \langle n, \gamma + d\mu + d\rangle^+ \in S_0$ $\langle n, \gamma + d\mu + d\rangle^+ \models F'$ but $\langle m, \eta + d\mu + d\rangle^+ \not\models F'$ thus $\langle m, \eta\mu\rangle^+ \not\models \exists F'$ but $\langle n, \gamma\mu\rangle^+ \models \exists F'$. This again gives a contradiction to $(\langle n, \gamma\mu\rangle^+, \langle m, \eta\mu\rangle^+) \in \mathcal{B}$ which we assumed.

1+2 give us that $\langle m, \eta\mu\rangle^+ \xrightarrow{\epsilon(d)}{}^+ S_0$

1+2 give us that $\mathcal{B}$ is in fact a timed probabilistic bisimulation. □

3.2 The logic RTPL$_{weak}$

In the previous chapter we defined weak timed probabilistic bisimulation. We would like to have a logic in which we can express weak properties that is, a logic which abstract from urgent actions of timed probabilistic processes and hence characterizes weak timed probabilistic bisimulation.

In this section we present the logic RTPL$_{weak}$ and we prove that this logic characterizes weak timed probabilistic bisimulation. We now define the syntax of RTPL$_{weak}$.

Definition 3.4 *Let K be a finite set of clocks and k an integer. Furthermore let $p \in \mathbf{Q}_{\geq 0}$ be a probability parameter. Then* RTPL$_{weak}$ *is the set of fomulae over K and k generated by the abstract syntax :*

$$\begin{array}{rcl} F & ::= & tt \mid ff \mid F_1 \wedge F_2 \mid F_1 \vee F_2 \mid \langle\!\langle \alpha \rangle\!\rangle_{\geq p} F \mid \langle\!\langle \alpha \rangle\!\rangle_{> p} F \mid [\![\alpha]\!] F \\ & & \mid \forall\!\!\!\forall_{\sqsupseteq p} F \mid \exists\!\!\!\exists_{\sqsupseteq p} F \mid c_1 \textbf{ in } F \mid c_1 + x \sim c_2 + y \end{array}$$

where $\alpha \in Act$; $c_1, c_2 \in K$; $x, y \in \{0, 1, \ldots, k\}$; $\sim \in \{=, \leq, \geq, <, >\}$; $p \in [0,1]$ and $\sqsupseteq \in \{>, \geq\}$.

Since we interpret formulas over extended states we have to define weak transition functions on the extended configuration graph.

Definition 3.5 *The weak transition functions are defined as*

1 $\pi_w^+(\langle n, \gamma\mu\rangle^+, \alpha, \langle n', \gamma'\mu\rangle^+) = p$ *iff* $\pi_w(\langle n, \gamma\rangle, \alpha, \langle n', \gamma'\rangle) = p$

2 $\langle n, \gamma\mu\rangle^+ \overset{\epsilon(d),p}{\Longrightarrow}{}^+ \langle n, \gamma + d\mu + d\rangle^+$ *iff* $\langle n, \gamma\rangle \overset{\epsilon(d),p}{\Longrightarrow} \langle n, \gamma + d\rangle$

Notation: Whenever $\pi_w^+(\langle n, \gamma\mu\rangle^+, a, \langle n', \gamma'\mu\rangle^+) = p$ for $p > 0$ we write $\langle n, \gamma\mu\rangle^+ \overset{a,p}{\Longrightarrow}{}^+ \langle n', \gamma'\mu\rangle^+$

Now we can define the semantics of RTPL$_{weak}$.

Definition 3.6 *Let $G = \langle N, n_0, C, \longrightarrow\rangle$ be a TPG. Let $\langle n, \gamma\mu\rangle^+$ be a state in an extended configuration graph of G. Let $\sqsupseteq$ denote either $\geq$ or $>$. We now define the weak satisfaction relation $\models_w$ as follows.*

$$
\begin{array}{lcl}
\langle n,\gamma\mu\rangle^+ \models_w tt & & \\
\langle n,\gamma\mu\rangle^+ \models_w F \vee G & \overset{\text{def}}{\Leftrightarrow} & \langle n,\gamma\mu\rangle^+ \models_w F \textit{ or } \langle n,\gamma\mu\rangle^+ \models_w G \\
\langle n,\gamma\mu\rangle^+ \models_w F \wedge G & \overset{\text{def}}{\Leftrightarrow} & \langle n,\gamma\mu\rangle^+ \models_w F \textit{ and } \langle n,\gamma\mu\rangle^+ \models_w G \\
\langle n,\gamma\mu\rangle^+ \models_w \langle\langle\alpha\rangle\rangle_{\sqsupseteq p} F & \overset{\text{def}}{\Leftrightarrow} & \sum\{\!| \ \pi_w^+(\langle n,\gamma\mu\rangle^+, \alpha, \langle n',\gamma'\mu\rangle^+) \mid \langle n',\gamma'\mu\rangle^+ \models_w F \ |\!\} \sqsupseteq p \\
\langle n,\gamma\mu\rangle^+ \models_w [\![\alpha]\!] F & \overset{\text{def}}{\Leftrightarrow} & \forall \langle n',\gamma'\mu\rangle^+ . \langle n,\gamma\mu\rangle^+ \overset{a,p}{\Longrightarrow}^+ \langle n',\gamma'\mu\rangle^+ \Rightarrow \langle n',\gamma'\mu\rangle^+ \models_w F \\
\langle n,\gamma\mu\rangle^+ \models_w \forall\!\!\!\forall_{\sqsupseteq p} F & \overset{\text{def}}{\Leftrightarrow} & \forall d \in \mathbf{R}_{\geq 0} . \sum\{\!| \ p' \mid \langle n,\gamma\mu\rangle^+ \overset{\epsilon(d),p'}{\Longrightarrow}^+ \langle n',\gamma'\mu + d\rangle^+ \\
 & & \quad \textit{and } \langle n',\gamma'\mu + d\rangle^+ \models_w F \ |\!\} \sqsupseteq p \\
\langle n,\gamma\mu\rangle^+ \models_w \exists\!\!\!\exists_{\sqsupseteq p} F & \overset{\text{def}}{\Leftrightarrow} & \exists d \in \mathbf{R}_{\geq 0} . \sum\{\!| \ p' \mid \langle n,\gamma\mu\rangle^+ \overset{\epsilon(d),p'}{\Longrightarrow}^+ \langle n',\gamma'\mu + d\rangle^+ \\
 & & \quad \textit{and } \langle n',\gamma'\mu + d\rangle^+ \models_w F \ |\!\} \sqsupseteq p \\
\langle n,\gamma\mu\rangle^+ \models_w c_1 \textbf{ in } F & \overset{\text{def}}{\Leftrightarrow} & \langle n,\gamma\mu'\rangle^+ \models_w F \textit{ where } \mu' = \mu[c_1 \leftarrow 0] \\
\langle n,\gamma\mu\rangle^+ \models_w c_1 + x \sim c_2 + y & \overset{\text{def}}{\Leftrightarrow} & \mu(c_1) + x \sim \mu(c_2) + y
\end{array}
$$

We will now prove that RTPL$_{weak}$ characterizes weak timed probabilistic bisimulation.

Theorem 3.2 *Two timed probabilistic processes $\langle n,\gamma\mu\rangle^+$, $\langle m,\eta\mu\rangle^+$ are weak timed probabilistic bisimular iff no formula $F \in RTPL_{weak}$ with formula clock set K can distinguish between them. That is for all $\mu \in \mathbf{R}^K$*

$$\langle n,\gamma\mu\rangle^+ \approx_{tp} \langle m,\eta\mu\rangle^+ \Leftrightarrow (\forall F \in \mathrm{RTPL}_{weak} . \langle n,\gamma\mu\rangle^+ \models_w F \Leftrightarrow \langle m,\eta\mu\rangle^+ \models_w F)$$

Proof: $\Rightarrow$: It will be enough to show that: $\langle n,\gamma\mu\rangle^+ \approx_{tp} \langle m,\eta\mu\rangle^+ \wedge \langle n,\gamma\mu\rangle^+ \models_w F \Rightarrow \langle m,\eta\mu\rangle^+ \models_w F$. We proceed by induction in the structure of the formulas with the following IH: $\langle n',\gamma'\mu\rangle^+ \approx_{tp} \langle m',\eta'\mu\rangle^+ \wedge \langle n',\gamma'\mu\rangle^+ \models_w F' \Rightarrow \langle m',\eta'\mu\rangle^+ \models_w F'$ where F' is a subformula of F. We will only show the cases $\vee$, $\langle\langle\alpha\rangle\rangle_{\sqsupseteq p}$, $\forall\!\!\!\forall_{\sqsupseteq p}$, $\exists\!\!\!\exists_{\sqsupseteq p}$

Cases:

$\underline{F = F_1 \vee F_2}$: Assume $\langle n,\gamma\mu\rangle^+ \approx_{tp} \langle m,\eta\mu\rangle^+$ and $\langle n,\gamma\mu\rangle^+ \models_w F$. Then by definition of $\models_w$ we have that $\langle n,\gamma\mu\rangle^+ \models_w F_1$ or $\langle n,\gamma\mu\rangle^+ \models_w F_2$. Assume $\langle n,\gamma\mu\rangle^+ \models_w F_1$, now F_1 is a subformula of F and by IH we get $\langle m,\eta\mu\rangle^+ \models_w F_1$. Then the definition of $\models_w$ give us $\langle m,\eta\mu\rangle^+ \models F_1 \vee F_2$. The case where we assume $\langle n,\gamma\mu\rangle^+ \models_w F_2$ follow the above approach.

$\underline{F = \langle\langle a\rangle\rangle_{\geq p} F'}$: Assume $\langle n,\gamma\mu\rangle^+ \approx_{tp} \langle m,\eta\mu\rangle^+$ and $\langle n,\gamma\mu\rangle^+ \models_w F$. Let $S = \{\langle n',\gamma'\mu\rangle^+ \mid \langle n,\gamma\mu\rangle^+ \overset{a,p}{\Longrightarrow}^+ \langle n',\gamma'\mu\rangle^+ \wedge \langle n',\gamma'\mu\rangle^+ \models_w F'\}$. Then by definition of $\models_w$ we now have $\pi_w^+(\langle n,\gamma\mu\rangle^+, a, S) \geq p$, where $p > 0$. Let $S' \subseteq \cup\{T \in Proc/\approx_{tp} \mid T \cap S \neq \emptyset\}$. We know that $\forall \langle n'',\gamma''\mu\rangle^+ \in S' . \exists\, \langle n',\gamma'\mu\rangle^+ \in$

$S.\langle n'', \gamma''\mu\rangle^+ \approx_{tp} \langle n', \gamma'\mu\rangle^+$. By IH this implies that $\forall\langle n'', \gamma''\mu\rangle^+ \in S'.\langle n'', \gamma''\mu\rangle^+ \models_w F'$. Since $S \subseteq S'$ we get : $\pi_w^+(\langle n, \gamma\mu\rangle^+, a, S') \geq \pi_w^+(\langle n, \gamma\mu\rangle^+, a, S) \geq p$. Because $\langle n, \gamma\mu\rangle^+ \approx_{tp} \langle m, \eta\mu\rangle^+$ we know by definition of $\approx_{tp}$ that $\pi_w^+(\langle n, \gamma\mu\rangle^+, a, S) = \pi_w^+(\langle m, \eta\mu\rangle^+, a, S)$ then by definition of $\models_w$ we must have $\langle m, \eta\mu\rangle^+ \models_w \langle\!\langle a\rangle\!\rangle_{\geq p} F'$

$\underline{F = \langle\!\langle a\rangle\!\rangle_{>p} F'}$: As for $F = \langle\!\langle a\rangle\!\rangle_{\geq p} F'$

$\underline{F = \text{₩}_{\sqsupseteq p} F'}$: Assume that $\langle n, \gamma\mu\rangle^+ \approx_{tp} \langle m, \eta\gamma\rangle^+$ and $\langle n, \gamma\mu\rangle^+ \models_w F$.

Let $S = \{\langle n', \gamma'\mu'\rangle^+ \mid \exists p' \in [0,1], \exists\, d \in \mathbf{R}_{\geq 0}.\, \langle n, \gamma\mu\rangle^+ \overset{\epsilon(d),p'}{\Longrightarrow}{}^+ \langle n', \gamma'\mu'\rangle^+$ and $\langle n', \gamma'\mu'\rangle^+ \models_w F'\}$.

Now, let $S' = \cup\{T \in Proc/{\approx_{tp}} \mid T \cap S \neq \emptyset\}$.

(*) We know that $\forall\langle n'', \gamma''\mu''\rangle^+ \in S'.\, \exists\langle n', \gamma'\mu'\rangle^+ \in S.\, \langle n'', \gamma''\mu''\rangle^+ \approx_{tp} \langle n', \gamma'\mu'\rangle^+$.

By IH : $\forall\langle n'', \gamma''\mu''\rangle^+ \in S'.\, \langle n'', \gamma''\mu''\rangle^+ \models_w F'$.

Now, as $S \subseteq S'$ we know : $\forall d \in \mathbf{R}_{\geq 0}.\, \langle n, \gamma\mu\rangle^+ \overset{\epsilon(d),p}{\Longrightarrow}{}^+ S'$ where $p = \sum\{\!| \, p' \mid \langle n, \gamma\mu\rangle^+ \overset{\epsilon(d),p'}{\Longrightarrow}{}^+ \langle n', \gamma'\mu'\rangle^+$ and $\langle n', \gamma'\mu'\rangle^+ \in S \,|\!\}$.

Now, from the definition of $\approx_{tp}$ we know: $\forall d \in \mathbf{R}_{\geq 0}.\, \langle m, \eta\mu\rangle^+ \overset{\epsilon(d),p}{\Longrightarrow}{}^+ S'$.

This together with (*) gives that $\langle m, \eta\mu\rangle^+ \models_w \text{₩}_{\sqsupseteq p} F'$.

$\underline{F = \exists\!\!\boxplus_{\sqsupseteq p} F'}$: As for the case $F = \text{₩}_{\sqsupseteq p} F'$

$\Leftarrow$: We have to show that $\mathcal{B} = \{(\langle n, \gamma\mu\rangle^+, \langle m, \eta\mu\rangle^+) \mid \langle n, \gamma\mu\rangle^+ \models_w F \Leftrightarrow \langle m, \eta\mu\rangle^+ \models_w F\}$ is a weak timed probabilistic bisimulation. Consider an arbitrary pair $(\langle n, \gamma\mu\rangle^+, \langle m, \eta\mu\rangle^+) \in \mathcal{B}$, then by the definition of $\approx_{tp}$ we have to consider two cases.
1: Let $S_0 \in Proc/\mathcal{B}$ and let $\pi_w^+(\langle n, \gamma\mu\rangle^+, a, S_0) = p_0$.
Furthermore let $\langle m, \eta\mu\rangle^+ \overset{a,p_1}{\Longrightarrow}{}^+ \langle m'\eta'\mu\rangle_1^+, \ldots, \langle m, \eta\mu\rangle^+ \overset{a,p_k}{\Longrightarrow}{}^+ \langle m', \eta'\mu\rangle_k^+$ be the weak transitions from $\langle m, \eta\mu\rangle^+$. The a-descendants of $\langle m, \eta\mu\rangle^+$ can be ordered s.t. $\langle m', \eta'\mu\rangle_1^+, \ldots, \langle m', \eta'\mu\rangle_i^+ \in S_0$ and $\langle m', \eta'\mu\rangle_{i+1}^+, \ldots, \langle m', \eta'\mu\rangle_k^+ \notin S_0$. Then we have

$$\begin{array}{lll}
\langle m', \eta'\mu\rangle_{i+1}^+ \notin S_0 & \Rightarrow & \exists F_{i+1}.\forall\langle n', \gamma'\mu\rangle^+ \in S_0\ \langle n', \gamma'\mu\rangle^+ \models_w F_{i+1} \wedge \langle m', \eta'\mu\rangle_{i+1}^+ \not\models_w F_{i+1} \\
\vdots & & \vdots \\
\langle m', \eta'\mu\rangle_k^+ \notin S_0 & \Rightarrow & \exists F_k.\forall\langle n', \gamma'\mu\rangle^+ \in S_0\ \langle n', \gamma'\mu\rangle^+ \models F_k \wedge \langle m', \eta'\mu\rangle_k^+ \not\models_w F_k
\end{array}$$

that is, we can find formulas $F_{i+1}, \ldots, F_k$ such that $\forall\langle n', \gamma'\mu\rangle^+ \in S_0\ \langle n', \gamma'\mu\rangle^+ \models_w F_j$ but $\langle m', \eta'\mu\rangle_j^+ \not\models_w F_j$ for $i+1 \leq j \leq k$. Now assume $\sum_{j=1}^{i} p_j < p_0$ where $\pi_w^+(\langle m, \eta\mu\rangle^+, a, \langle m, \eta\mu\rangle_j^+) = p_j$. Then $\langle n, \gamma\mu\rangle^+ \models_w \langle\!\langle a\rangle\!\rangle_{\geq p_0} \bigwedge_{j=i+1}^{k} F_j$ but $\langle m, \eta\mu\rangle^+ \not\models_w \langle\!\langle a\rangle\!\rangle_{\geq p_0} \bigwedge_{j=i+1}^{k} F_j$ contradicting that $(\langle n, \gamma\mu\rangle^+, \langle m, \eta\mu\rangle^+) \in \mathcal{B}$. Thus $p_0 \leq \sum_{j=1}^{i} p_i$ and therefore $\pi_w^+(\langle m, \eta\mu\rangle^+, a, S_0) = p'$ with $p' \geq \sum_{j=1}^{i} p_i \geq p_0$. By symmetry of $\mathcal{B}$ it follows that $p_0 \geq p'$ hence $p_0 = p'$.

2: Let $S_0 \in Proc/\mathcal{B}$ and let $\langle n, \gamma\mu\rangle^+ \overset{\epsilon(d),p}{\Longrightarrow}{}^+ S_0$.

This follows the above approach.

1+2 give us that $\mathcal{B}$ is in fact a weak timed probabilistic bisimulation. □

3.3 The logic RTPL$_{until}$

Different logics make it possible, at the same time, to express properties about global time and probabilities. An example of such a property is: The process P will with probability greater than $\frac{1}{2}$ within 5 time units perform an a-action, thereby reaching a state satisfying the formula F.

An example of such a logic is TPCTL [Han91]. TPCTL is an extension of the logic CTL and is interpreted over Labelled Concurrent Markov Chains (LCMC). In these LCMC there is a strict partitioning of non-deterministic states and probabilistic states, and of non-deterministic transitions and probabilistic transitions. Furthermore there is a strict alternation between action tansitions and probabilistic transitions. The passage of time is modelled with special discrete χ transitions.

Since this model has both non-deterministic and probabilistic choice, one gets two types of quantification in TPCTL formulas: Probabilistic quantification and non-deterministic quantification. In TPCTL it is possible to express properties such as:

$$F_1 \, EU^{\leq t}_{\geq p} \, F_2 \text{ and } F_1 \, AU^{\leq t}_{\geq p} \, F_2$$

Intuitively $F_1 \, AU^{\leq t}_{\geq p} \, F_2$ holds for a LCMC, if for all non-deterministic choices in the LCMC, at least with probability p both F_2 will become true within t time units and F_1 will be true until F_2 becomes true. And intuitively $F_1 \, EU^{\leq t}_{\geq p} \, F_2$ holds for a LCMC, if there exist a non-deterministic choice in the LCMC such that there is at least a probability p that F_2 will become true within t time units and that F_1 will be true until F_2 becomes true.

Properties as the above are very important properties to be able to express about reliable real time systems, and we would of course like to be able to express similar properties. Therefore we extend RTPL with operators similar to the above. We now present the syntax for RTPL$_{until}$.

Definition 3.7 *Let K be a finite set of clocks and k an integer. Furthermore let $t \in \mathbf{N}_{\geq 0}$ be a time parameter and $p \in \mathbf{Q}_{\geq 0}$ a probability parameter. Then the set of RTPL$_{until}$ fomulae over K and k is given by the abstract syntax :*

$$\begin{aligned} F \quad ::= \quad & tt \mid \neg F \mid F_1 \wedge F_2 \mid \langle \alpha \rangle_{\geq p} F \mid \langle \alpha \rangle_{> p} F \mid \forall\!\!\forall F \mid c_1 \textbf{ in } F \\ & \mid c_1 + x \sim c_2 + y \mid F_1 \, EU^{\leq t}_{\geq p} \, F_2 \mid F_1 \, EU^{\leq t}_{> p} \, F_2 \\ & \mid F_1 \, AU^{\leq t}_{\geq p} \, F_2 \mid F_1 \, AU^{\leq t}_{> p} \, F_2 \end{aligned}$$

where $\alpha \in Act$; $c_1, c_2 \in K$; $x, y \in \{0, 1, \ldots, k\}$; $\sim \in \{=, \leq, \geq, <, >\}$; $t \in \mathbf{N}_{\geq 0}$ and $p \in [0, 1]$.

Since RTPL$_{until}$ is an extension of RTPL we will be able to use the abbreviations defined in Section 3.1.

Intuitively, $F_1\, EU^{\leq t}_{\geq p}\, F_2$ holds for a timed probabilistic process, if there exists a way of resolving all non-deterministic choices in the process, such that the probability is at least p that both F_2 will become true within t time units and that F_1 will be true from now on until F_2 becomes true, i.e. , U is an analogue to the *strong until* in CTL. Similarly $F_1\, AU^{\leq t}_{\geq p}\, F_2$ holds, if for all non-deterministic choices, the probability is at least p that both F_2 will become true within t time units and that F_1 will be true from now on until F_2 becomes true. The formulas $F_1\, EU^{\leq t}_{>p}\, F_2$ and $F_1\, AU^{\leq t}_{>p}\, F_2$ have analogous meanings.

We interpret the non-deterministic choices of a timed probabilistic transition system as follows: In a given state a timed probabilistic process will either let time pass or perform an action with a certain probability. The actual behaviour of the process depends on the well known *two-phase principle:* time is considered to pass in between actions and thus action transitions and delay transitions do not compete probabilistically. Therefore the non-deterministic choice can be perceived as the choice, in a given state, between different delay transitions i.e. the choice is a certain way to *resolve* the amount of time a process will delay. We will now formally define what is meant by a *resolved probabilistic transition system.*

Definition 3.8 *Let $P = \langle S, s_0, \pi, \longrightarrow\rangle$ be a timed probabilistic transition system. A resolved probabilistic transition system for P is a purely probabilistic transition system $R = \langle T, t_0, \pi_R\rangle$ where*

- *$T = T_{Act} \cup T_{\mathcal{D}}$ is the state set*
- *$t_0 \in T_{\mathcal{D}}$ is the initial state, $t_0 = s_0$*
- *$\pi_R : T \times (Act \cup \mathcal{D}) \times T \rightarrow [0,1]$ is a probabilistic transition function, defined for $m \in Act \cup \mathcal{D}$ and $t_1, t_2 \in T$ by*

  $$\pi_R(t_1, m, t_2) = \begin{cases} \pi(t_1, m, t_2) & \textit{if } m \in Act,\ t_1 \in T_{Act},\ t_2 \in T_{\mathcal{D}} \\ 1 & \textit{if } t_1 \in T_{\mathcal{D}},\ t_2 \in T_{Act},\ m = d(t_1) \\ 0 & \textit{otherwise} \end{cases}$$

  where $d : T_{\mathcal{D}} \rightarrow \mathbf{R}_{\geq 0}$ is a function that for each $t_1 \in T_{\mathcal{D}}$ resolves the unique $m = d(t_1)$ s.t. given $t_2 \in T_{Act}$ we have

  $$\pi_R(t_1, \epsilon(m), t_2) = 1$$

By Res(P) we denote the set of resolved probabilistic transition systems for timed probabilistic transition system P.

We impose the following restrictions on the probabilistic transition function π_R.

- For all states $t \in T$ and events $m_1 \in$ Act, $m_2 \in \mathcal{D}$ we require

  $$\neg(\pi_R(t_1, m_1) > 0 \text{ and } \pi_R(t_1, m_2) > 0)$$

- Strict alternation between *delay transitions* and *action transitions.*

$$\begin{array}{l}\text{If } t_1 \in T_{Act},\ m \in \text{Act and } \pi_R(t_1, m, t_2) > 0 \text{ then } t_2 \in T_{\mathcal{D}} \\ \text{If } t_1 \in T_{\mathcal{D}},\ m \in \mathcal{D} \text{ and } \pi_R(t_1, m, t_2) > 0 \text{ then } t_2 \in T_{Act}\end{array}$$

A particular computation or *path* of a resolved process can now be defined as follows.

Definition 3.9 *Let $R = \langle T, t_0, \pi_R\rangle$ be a resolved probabilistic transition system. A resolved path σ in R is a possibly infinite alternating sequence of consecutive transitions in R*

$$\sigma = t_0 \overset{\epsilon(d_0),1}{\longrightarrow} t_1 \overset{a_1,p_1}{\longrightarrow} t_2 \overset{\epsilon(d_1),1}{\longrightarrow} t_3 \overset{a_2,p_2}{\longrightarrow} \ldots$$

where $t_i \in T$, $d_i = d(t_i)$ and $\longrightarrow \in \pi_R$. By Paths(R) we denote the set of resolved paths in R. By Paths$_{fin}$(R) we denote the set of finite segments of Paths(R) where for $\sigma \in Paths(R)$ we have that $\sigma' \in Paths_{fin}(R)$ is the finite segment of σ s.t. for some $, n \in \mathbf{N}_{\geq 0}$ we have that

$$\sigma' = t_0 \overset{\epsilon(d_0),1}{\longrightarrow} t_1 \overset{a_1,p_1}{\longrightarrow} t_2 \ldots \overset{\epsilon(d_{n-2}),1}{\longrightarrow} t_{n-1} \overset{a_{n-1},p_{n-1}}{\longrightarrow} t_n$$

Given a finite path of a resolved probabilistic transition system, we define two functions that accumulates the delays and probabilities respectively, in the path and a third function that gives the last state in a path.

Definition 3.10 *Let $\sigma \in Path_{fin}(R)$ in a resolved probabilistic transition system R. We define a function time: Paths$_{fin}$(R) $\rightarrow \mathbf{R}_{\geq 0}$ as follows*

$$time(\sigma) = \sum \{\!| \ d \mid \exists t, t'.\, t \overset{\epsilon(d)}{\longrightarrow} t' \in \sigma \ |\!\}$$

Definition 3.11 *Let $\sigma \in Path_{fin}(R)$ in a resolved probabilistic transition system R. We define a probability function prob: Paths$_{fin}$(R) $\rightarrow \mathbf{R}_{\geq 0}$ as follows:*

$$prob(\sigma) = \prod \{\!| \ p \mid \exists t, t', a.\, t \overset{a,p}{\longrightarrow} t' \in \sigma \ |\!\}$$

If $\sigma = t_0 \in T_{\mathcal{D}}$ then prob(σ) = 1

Definition 3.12 *Let $\sigma \in Path_{fin}(R)$ in a resolved probabilistic transition system R we define a function term: Paths$_{fin}$(R) $\rightarrow T$ as follows*

$$term(\sigma) = t_n$$

We now define the semantics of RTPL$_{until}$

Definition 3.13 *Let $\langle n, \gamma\mu\rangle^+$ be a state in the extended configuration graph E and let $Res(E,\langle n, \gamma\mu\rangle^+)$ denote the set of resolved probabilistic transition systems of E starting in state $\langle n, \gamma\mu\rangle^+$. Let $\sqsupseteq$ denote either $\geq$ or $>$. We now define the satisfaction relation $\models_u$ as follows.*

$$
\begin{array}{lll}
\langle n, \gamma\mu\rangle^+ \models_u tt & & \\
\langle n, \gamma\mu\rangle^+ \models_u \neg F & \overset{\text{def}}{\Leftrightarrow} & \langle n, \gamma\mu\rangle^+ \not\models_u F \\
\langle n, \gamma\mu\rangle^+ \models_u F \wedge G & \overset{\text{def}}{\Leftrightarrow} & \langle n, \gamma\mu\rangle^+ \models_u F \textit{ and } \langle n, \gamma\mu\rangle^+ \models_u G \\
\langle n, \gamma\mu\rangle^+ \models_u \langle\alpha\rangle_{\sqsupseteq p} F & \overset{\text{def}}{\Leftrightarrow} & \sum\{\!|\ \pi^+(\langle n, \gamma\mu\rangle^+, \alpha, \langle n', \gamma'\mu\rangle^+) \mid \langle n', \gamma'\mu\rangle^+ \models_u F\ |\!\} \sqsupseteq p \\
\langle n, \gamma\mu\rangle^+ \models_u \forall\!\!\!\forall F & \overset{\text{def}}{\Leftrightarrow} & \forall d \in \mathbf{R}_{\geq 0}.\langle n, \gamma + d\mu + d\rangle^+ \models_u F \\
\langle n, \gamma\mu\rangle^+ \models_u c_1 \textbf{ in } F & \overset{\text{def}}{\Leftrightarrow} & \langle n, \gamma\mu'\rangle^+ \models_u F \textit{ where } \mu' = \mu[c_1 \leftarrow 0] \\
\langle n, \gamma\mu\rangle^+ \models_u c_1 + x \sim c_2 + y & \overset{\text{def}}{\Leftrightarrow} & \mu(c_1) + x \sim \mu(c_2) + y \\
\langle n, \gamma\mu\rangle^+ \models_u F_1\, EU^{\leq t}_{\sqsupseteq p} F_2 & \overset{\text{def}}{\Leftrightarrow} & \exists R \in Res(E, \langle n, \gamma\mu\rangle^+). \\
& & \sum\{\!|\ prob(\sigma_1) \mid \exists \sigma \in Paths(R), \exists \sigma_2 : \sigma = \sigma_1\sigma_2, \\
& & \quad time(\sigma_1) \leq t, \\
& & \quad term(\sigma_1) \models_u F_2, \\
& & \quad \forall s \in \sigma_1 \setminus term(\sigma_1).\ s \models_u F_1\ |\!\} \sqsupseteq p \\
\langle n, \gamma\mu\rangle^+ \models_u F_1\, AU^{\leq t}_{\sqsupseteq p} F_2 & \overset{\text{def}}{\Leftrightarrow} & \forall R \in Res(E, \langle n, \gamma\mu\rangle^+). \\
& & \sum\{\!|\ prob(\sigma_1) \mid \exists \sigma \in Paths(R), \exists \sigma_2 : \sigma = \sigma_1\sigma_2, \\
& & \quad time(\sigma_1) \leq t, \\
& & \quad term(\sigma_1) \models_u F_2, \\
& & \quad \forall s \in \sigma_1 \setminus term(\sigma_1).\ s \models_u F_1\ |\!\} \sqsupseteq p
\end{array}
$$

Theorem 3.3 *Two timed probabilistic processes $\langle n, \gamma\mu\rangle^+$, $\langle m, \eta\mu\rangle^+$ are strong timed probabilistic bisimular iff no formula $F \in \mathrm{RTPL}_{until}$ with formula clock set K can distinguish between them. That is for all $\mu \in \mathbf{R}^K$:*

$$\langle n, \gamma\mu\rangle^+ \sim_{tp} \langle m, \eta\mu\rangle^+ \Leftrightarrow (\forall F \in RTPL_{until}.\langle n, \gamma\mu\rangle^+ \models_u F \Leftrightarrow \langle m, \eta\mu\rangle^+ \models_u F)$$

Proof:

$\Rightarrow$: It will be enough to show that: $\langle n, \gamma\mu\rangle^+ \sim_{tp} \langle m, \eta\mu\rangle^+ \wedge \langle n, \gamma\mu\rangle^+ \models_u F \Rightarrow \langle m, \eta\mu\rangle^+ \models_u F$. We proceed by induction in the structure of the formulas with the following IH: $\langle n', \gamma'\mu\rangle^+ \sim_{tp} \langle m', \eta'\mu\rangle^+ \wedge \langle n', \gamma'\mu\rangle^+ \models_u F' \Rightarrow \langle m', \eta'\mu\rangle^+ \models_u F'$ where F' is a subformula of F. We will only show the cases $F = F_1\, AU^{\leq t}_{\geq p} F_2$ and $F = F_1\, EU^{\leq t}_{\geq p} F_2$ since the rest is proven as in Theorem 3.1.

Cases:

$\underline{F = F_1\, AU^{\leq t}_{\geq p} F_2}$: Assume that $\langle n, \gamma\mu\rangle^+ \sim_{tp} \langle m, \eta\mu\rangle^+$ and $\langle n, \gamma\mu\rangle^+ \models_u F$. Then by definition of $\models_u$ we know that for all resolvents $R \in Res(E, \langle n, \gamma\mu\rangle^+)$ the following holds:

$$
\begin{array}{ll}
\sum\{\!|\ prob(\sigma_1) \mid & \exists \sigma \in Paths(R), \exists \sigma_2 : \sigma = \sigma_1\sigma_2, time(\sigma_1) \leq t, \\
& term(\sigma_1) \models_u F_2, \forall s \in \sigma_1 \setminus term(\sigma_1).\ s \models_u F_1\ |\!\} \sqsupseteq p
\end{array}
$$

Assume $R \in Res(E, \langle n, \gamma\mu\rangle^+)$. Now as $\langle n, \gamma\mu\rangle^+ \sim_{tp} \langle m, \eta\mu\rangle^+$ we know for all states $\langle n', \gamma'\mu'\rangle^+ \in R$ there exists states defining a resolved probabilistic transition system $Res(E, \langle m, \eta\mu\rangle^+)$ s.t.

$$\forall \langle n', \gamma'\mu'\rangle^+ \in R. \exists \langle m', \eta'\mu'\rangle^+ \in Res(E, \langle m, \eta\mu\rangle^+ . \langle n', \gamma'\mu'\rangle^+ \sim_{tp} \langle m', \eta'\mu'\rangle^+$$

Now the result follows from the induction hypothis.

$\underline{F = F_1\, EU^{\leq t}_{\geq p}\, F_2}$: As for $F = F_1\, AU^{\leq t}_{\geq p}\, F_2$

$\Leftarrow$: Exactly as for Theorem 3.1 □

3.3.1 Properties expressible in RTPL$_{until}$

In this section we describe some of the interesting properties expressible in RTPL$_{until}$, and we show that our until operators are in fact generalizations of analogous CTL operators.

In CTL there are, among others, the following four operators:

$AG\,F$ expresses:	*for all non-deterministic choices always F*,
$AF\,F$ expresses:	*for all non-deterministic choices eventually F*,
$EG\,F$ expresses:	*there exists a non-deterministic choice s.t. always F*,
$EF\,F$ expresses:	*there exists a non-deterministic choice s.t. eventually F.*

The same properties are expressible in HML extended with recursion as *Inv F*, *Even F*, *Saf F* and *Pos F* [Lar90], respectively. These properties are all qualitative properties and they are certainly relevant and convenient. In RTPL we can express these properties using a *weak until* operator $\mathcal{U}$.

Intuitively $F_1 \mathcal{U} F_2$ means that either F_2 holds at some time in the future and F_1 holds until then, or F_1 holds forever. To see how our $\mathcal{U}$ operators can be derived from our U operators consider the formula $F_1\, E\mathcal{U}^{\leq t}_{\geq p}\, F_2$. The intuitive meaning of this formula is that $F_1\, E\mathcal{U}^{\leq t}_{\geq p}\, F_2$ holds for a timed probabilistic transition system P if there exists a resolved probabilistic transition system $R \in Res(P)$ s.t. the sum of probabilities of the paths $\sigma \in Paths(R)$, for which $F_1\, U^{\leq t}\, F_2$ holds or for which F_1 hold continuously for at least t time units, is at least p. Thus it must not be the case that, for all resolved probabilistic transition systems there is a probability greater than $1-p$ for the duality of $F_1 \mathcal{U}^{\leq t} F_2$. So, we want to express this duality in terms of U.

We are looking for a formula which holds for all states where $F_1 \mathcal{U}^{\leq t} F_2$ does not hold. First of all it must be the case that for these "dual states" F_1 can not hold for all moments in time $\leq t$, i.e. $\neg F_1$ holds for some moment $x \leq t$. Furthermore F_2 is not allowed to hold at any time $x \leq t$ without $\neg F_2$ holding for all moments until x, i.e. there will always be a prefix of the "dual paths" where $\neg F_2$ holds. Consequently, we have

$$F_1\, E\mathcal{U}^{\leq t}_{\geq p}\, F_2 \;\equiv \neg(\neg F_2\, AU^{\leq t}_{>(1-p)}\, (\neg F_1 \wedge \neg F_2))$$

Definition 3.14 *We define the following derived operators.*

$$\begin{aligned}
F_1\, E\mathcal{U}^{\leq t}_{\geq p}\, F_2 &\equiv \neg(\neg F_2\, AU^{\leq t}_{>(1-p)}\, (\neg F_1 \wedge \neg F_2)) \\
F_1\, E\mathcal{U}^{\leq t}_{>p}\, F_2 &\equiv \neg(\neg F_2\, AU^{\leq t}_{\geq(1-p)}\, (\neg F_1 \wedge \neg F_2)) \\
F_1\, A\mathcal{U}^{\leq t}_{\geq p}\, F_2 &\equiv \neg(\neg F_2\, EU^{\leq t}_{>(1-p)}\, (\neg F_1 \wedge \neg F_2)) \\
F_1\, A\mathcal{U}^{\leq t}_{>p}\, F_2 &\equiv \neg(\neg F_2\, EU^{\leq t}_{\geq(1-p)}\, (\neg F_1 \wedge \neg F_2))
\end{aligned}$$

Now we can define the operators corresponding to the CTL operators mentioned earlier.

$$\begin{aligned}
AG\, F &\equiv F\, A\mathcal{U}^{<\infty}_{\geq 1}\, \mathit{ff} \\
AF\, F &\equiv tt\, AU^{<\infty}_{\geq 1}\, F \\
EG\, F &\equiv F\, E\mathcal{U}^{<\infty}_{\geq 0}\, \mathit{ff} \\
EF\, F &\equiv tt\, EU^{<\infty}_{>0}\, F
\end{aligned}$$

Example 3.2 Given a process P with transition graph as shown in Figure 3.2. P will satisfy the formula $F = ttEU^{\leq 3}_{\geq \frac{3}{8}} F_{P_1}$ where F_{P_1} is any formula satisfied by P_1.

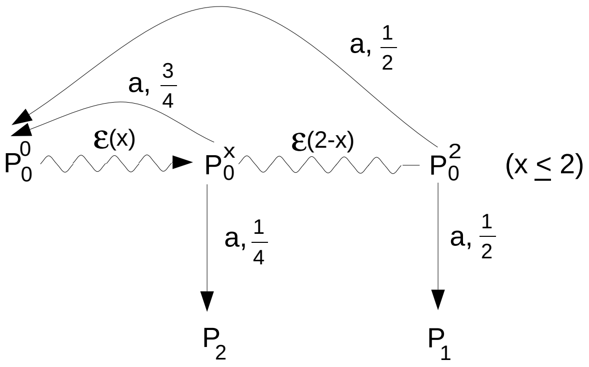


Figure 3.2: *Transition graph for the process P*

This can be seen by looking at a resolvent $R \in Res(P)$ shown at Figure 3.3 and a finite path $\sigma \in Paths(R)$ shown at Figure 3.4 As $Term(\sigma) = P_1$, $Prop(\sigma) = \frac{3}{8}$ and $time(\sigma) = 3$ the formula F is satisfied by the process P. □

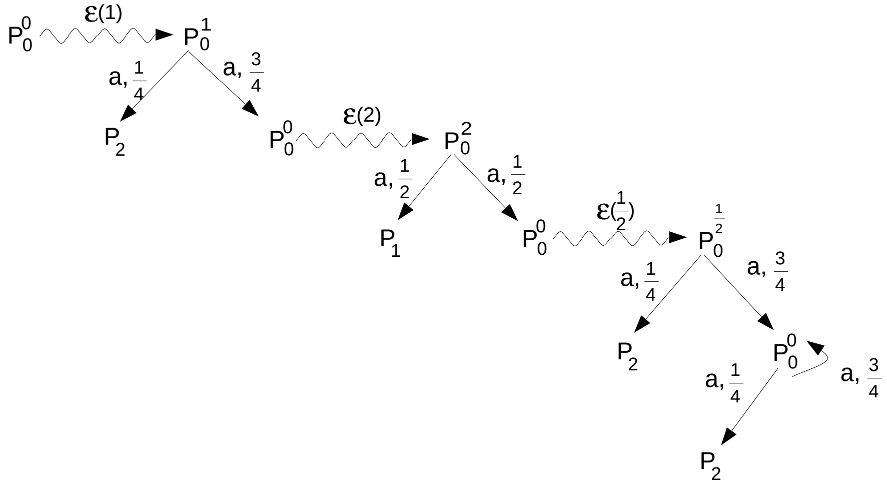


Figure 3.3: *Resolvent R for the proces P.*

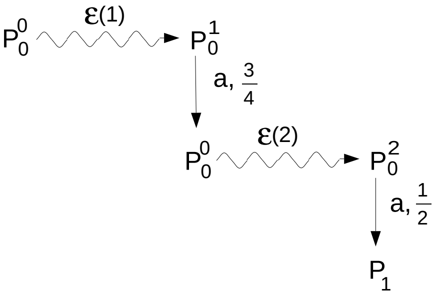


Figure 3.4: *Path σ in the resolvent R for the process P.*

Chapter 4

Model Checking

In this chapter we address the *satisfaction* problem, which in general terms concerns whether a given process in some implementation language satisfies a given specification in some logical specification language. Solving the satisfaction problem is known as the problem of *model checking*.

We address the model checking problems for specifications in the logics RTPL^{nf}, RTPL_{weak} and $\mathrm{RTPL}^{nf}_{until}$ in relation to implementations of timed probabilistic processes in terms of TPG's. That is: *Given a logical property F in one of the mentioned specification languages and given a TPG G, does G satisfy F, i.e. $G \models F$?*

The two logics RTPL^{nf}, $\mathrm{RTPL}^{nf}_{until}$ mentioned above are both fragments, chosen for the sake of simplicity, of the corresponding logics RTPL and RTPL_{until} where the negation of formulas is no longer syntactically possible. Instead we have added the explicit syntactic notion of: *ff*, $\vee$, $[\,]$ and $\exists$ s.t. formulas of $\mathrm{RTPL}^{nf}(\mathrm{RTPL}^{nf}_{until})$ are constructed from the syntax:

$$F \ ::= \mathcal{F} \mid f\!f \mid F_1 \vee F_2 \mid [\alpha]F \mid \exists F$$

where $\mathcal{F} \in \mathrm{RTPL}(\mathrm{RTPL}_{\mathrm{until}})$, $\alpha \in Act$.

The semantic interpretation of the new modalities is obvious. For $\langle n, \gamma\mu\rangle^{+}$ a state in some extended configuration graph $E = \langle N^{+}, n_0^{+}, \pi^{+}, \longrightarrow^{+}\rangle$ we let:

$$\begin{array}{lcl}
\langle n, \gamma\mu\rangle^{+} \models F \vee G & \overset{\text{def}}{\Leftrightarrow} & \langle n, \gamma\mu\rangle^{+} \models F \text{ or } \langle n, \gamma\mu\rangle^{+} \models G \\
\langle n, \gamma\mu\rangle^{+} \models [\alpha]F & \overset{\text{def}}{\Leftrightarrow} & \forall \langle n', \gamma'\mu\rangle^{+}.\ \langle n, \gamma\mu\rangle^{+} \xrightarrow{\alpha} \langle n', \gamma'\mu\rangle^{+} \Rightarrow \langle n', \gamma'\mu\rangle^{+} \models F \\
\langle n, \gamma\mu\rangle^{+} \models \exists F & \overset{\text{def}}{\Leftrightarrow} & \exists d \in \mathbf{R}_{\geq 0}.\ \langle n, \gamma + d\mu + d\rangle^{+} \models F
\end{array}$$

In this chapter we present two proof systems $\vdash$ and $\vdash_{weak}$ each forming the basis of a sound and complete model checking algorithm for specifications in respectively RTPL^{nf} and RTPL_{weak}. Furthermore, we present some not completely formalized ideas on model checking $\mathrm{RTPL}^{nf}_{until}$ and we propose a proof system $\vdash_u$ for satisfiability.

The fundamental technique used, to obtain decidability of the satisfaction problem, is the *region technique* of Alur and Dill [ACD90].

The problem of whether a TPG satisfies some formula is defined as whether the initial state of the configuration graph for the TPG, extended with the clock set of the formula, satisfies the formula.

Because of the infinite branching of time transitions in the configuration graph for a TPG, it is not possible to use the configuration graph directly as the basis of a model checking algorithm. Therefore, we use the region technique of Alur and Dill to provide an abstract semantics of TPG's in the form of finite labelled transition systems preserving the truth value of formulas from any of three logics.

In the following section we define the concept of regions and we use this concept to construct a finite representation of the configuration graph for a TPG. Furthermore, we define a representation of regions useful for implementation and for "talking" about regions. Then, in section 4.2 we present an extension of the finite representation of a configuration graph with respect to some formula clock set. This extension is similar to the notion of extended configuration graphs in the sense that, just as formulas are interpreted over extended configuration graphs so are formulas in the proof system $\vdash$ interpreted over the extended finite representation of configuration graphs. Finally, the proof system $\vdash$ is presented and shown to preserve the satisfiability of formulas. In section 4.3 we present the proof system $\vdash_{weak}$ following an approach similar to the one in section 4.2. Finally, in section 4.4 we present some ideas on model checking RTPL$_{until}$, and we propose a proof system $\vdash_u$ for satisfiability.

4.1 Regions

In this section we define a notion of regions applying equally well to any of the three logics with respect to which we want to model check some TPG.

The basic idea in the region technique is to partition the infinite state space of a configuration graph into finitely many symbolic states. The infinite state space is due to the infinitely many time assignments that clocks of a given TPG can be assigned. Therefore, the idea is to group these time assignments into finitely many equivalence classes denoted *regions*.

The requirement to the region partitioning is that, given a TPG G, two extended states $\langle n, \gamma\mu\rangle^+$, $\langle n', \gamma'\mu'\rangle^+$ in the extended configuration graph for G, where $\gamma\mu$ and $\gamma'\mu'$ are in the same region, satisfy the same formulas of RTPL^{nf}, RTPL$_{weak}$ and RTPL$^{nf}_{until}$.

In the definition of TPG's we have imposed a constraint on the set of possible enabling conditions on edges, namely that clocks of a TPG are always compared to integer constants or to their integral distance to other clocks using an operator $\sim \in \{\leq | < | \geq | > | =\}$. This restriction implies that actions of a TPG are enabled in one of two possible situations. Either when one or more clocks change integral part, corresponding to enabling conditions with operators $=, \geq$. Or when clocks with fractional part 0 change, corresponding to the enabling conditions with operator $>$.

Due to the above, two states of a TPG having the same integral part of all clocks and having the same set of clocks with fractional part 0, cannot be told apart by formulas of the type $\langle\alpha\rangle_{\sqsupseteq p}F$ from RTPL^{nf} or RTPL$^{nf}_{until}$. And neither can they be told apart by the weak modalities $\langle\!\langle\alpha\rangle\!\rangle_{\sqsupseteq p}F$ in RTPL$_{weak}$. Also the mutual ordering of fractional parts of all clocks will determine which clock will be the first to change its integral part and therefore the ordering will also determine which enabling conditions are satisfied and accordingly which formulas are being satisfied. Furthermore, the mutual integral distance between pairs of clocks can also determine whether some enabling conditions are satisfied or not, and accordingly whether some formulas are satisfied or not. Finally, whenever the value of a clock exceeds the maximal constant in any enabling condition, the actual value of the clock will be of no importance.

Furthermore, as it is only possible to compare formula clocks with integer constants in the atomic formulas of type $c_1 + x \sim c_2 + y$, we are assured that no formula can distinguish between equivalent time assignments using the delay modalities $\exists\!\!\exists$ or $\forall\!\!\!\forall$ in RTPL^{nf} and RTPL$^{nf}_{until}$ or using the weak delay modalities $\exists\!\!\exists\!\!\exists_{\sqsupseteq p}$, $\forall\!\!\!\forall\!\!\!\forall_{\sqsupseteq p}$ of RTPL$_{weak}$. Also, as we have restricted the upper bound t in formulas $F_1 EU^{\leq t}_{\sqsupseteq p} F_2$, $F_1 AU^{\leq t}_{\sqsupseteq p} F_2 \in$ RTPL$_{until}$ to be an integer, we are certain that states of a TPG having equivalent time assignments in the above sense can not be told apart.

The arguments above leads to the following formal definition of the equivalence relation over time assignments.

Definition 4.1 *Let $k \in \mathbf{N}$ and let C be a set of clocks. Then $\gamma, \gamma' \in \mathbf{R}^C$ are equivalent with respect to k, denoted $\gamma \stackrel{\bullet}{=} \gamma'$ iff:*

i) $\forall c \in C. \gamma(c) > k$ *iff* $\gamma'(c) > k$

ii) $\forall c \in C.$ *s.t.* $\gamma(c) \leq k. \lfloor\gamma(c)\rfloor = \lfloor\gamma'(c)\rfloor$ *and* $\{\gamma(c)\} = 0$ *iff* $\{\gamma'(c)\} = 0$

iii) $\forall c, c' \in C. \gamma(c) - \gamma(c') > k$ *iff* $\gamma'(c) - \gamma'(c') > k$

iv) $\forall c, c' \in C.$ *s.t.* $0 \leq \gamma(c) - \gamma(c') \leq k. \lfloor\gamma(c) - \gamma(c')\rfloor = \lfloor\gamma'(c) - \gamma'(c')\rfloor$ *and* $\{\gamma(c) - \gamma(c')\} = 0$ *iff* $\{\gamma'(c) - \gamma'(c')\} = 0$

Notation: For $\gamma \in \mathbf{R}^C$ let $[\gamma]$ represent the equivalence class for γ with respect to $\stackrel{\bullet}{=}$. The equivalence classes under $\stackrel{\bullet}{=}$ are called *regions*. We let ξ, ξ' range over $\mathbf{R}^C / \stackrel{\bullet}{=}$. Furthermore, we let $\mathcal{R}^C_k$ denote the set of all regions for a set C of clocks and the maximal constant k. Note that for any enabling condition b in $\mathcal{B}(C)$ with no constant greater than k, we have $b(\gamma) \Leftrightarrow b(\gamma')$, whenever γ and γ' belong to the same region in $\mathcal{R}^C_k$. Thus, for a region $\xi \in \mathcal{R}^C_k$ we can define $b(\xi)$ as the truth value of $b(\gamma)$ for any $\gamma \in \xi$. Let $C' \subseteq C$ we now define the following reset operator : $\xi[C' \leftarrow 0] = [\gamma[C' \leftarrow 0]]$ for all $\gamma \in \xi$. Furthermore, $\xi + n$ is well defined because for $\gamma, \gamma' \in \xi$ we have $\gamma + n \stackrel{\bullet}{=} \gamma' + n$

Any region has a unique successor region which is the next new region encountered when delaying all the clocks in the clock set. The successor region for a given region is defined as follows.

Definition 4.2 *Let* $[\gamma]$ *be a region and* $k \in \mathbf{N}$*, we then define the successor region* $succ([\gamma])$ *as the region* $[\gamma']$*, where:*

$$\gamma'(c) = \begin{cases} \gamma(c) + f; & \forall c.\gamma(c) \leq k \Rightarrow \{\gamma(c)\} > 0 \\ \gamma(c) + f/2; & \exists c.\gamma(c) \leq k \wedge \{\gamma(c)\} = 0 \end{cases}$$

where f *is defined as:*

$$f = 1 - max\,\{\{\gamma(c)\} \mid \gamma(c) \leq k\}$$

If $\{\{\gamma(c)\} \mid \gamma(c) \leq k\} = \emptyset$ *we define* $max\,\{\{\gamma(c)\} \mid \gamma(c) \leq k\} = 1$

Notation: We denote by ξ^i the i$^{\text{th}}$ successor region of ξ (i.e. $\xi^i = succ^i(\xi)$). From each region ξ, it is possible to reach a region ξ' s.t. $succ(\xi') = \xi'$, and we denote by l_ξ the required number of steps s.t. $\xi^{l_\xi} = succ(\xi^{l_\xi})$.

Example 4.1 For a clock set $C = \{c_1, c_2\}$ the figure 4.1 shows the set of regions defined by C and a maximal constant 1. There are five types of regions: corner points, vertical lines, horisontal lines, diagonals and open areas. Corner points, vertical lines and horisontal lines are called *boundary* regions, whereas the diagonals and the open areas are denoted *interior* regions.

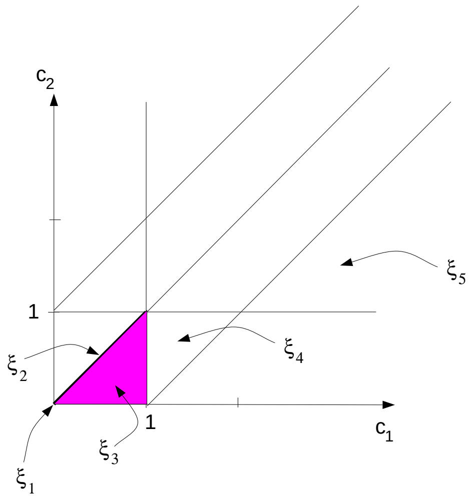


Figure 4.1: *Regions for clock set* $C = \{c_1, c_2\}$

The successor of a given region is the new region encountered by following 45° lines upwards to the right. In figure 4.1 $succ(\xi_1) = \xi_2$. In general, the successor of a boundary region will be an interior region and vice versa. □

Given a TPG G, we can now define the finite representation of the configuration graph for G. We call this representation the *State-Region* graph for G, denoted $SR[G]$.

Definition 4.3 *Let* $G = \langle N, n_0, C, \longrightarrow \rangle$ *be a TPG, and* k *the maximal constant in* G*. Then the state-region graph for* G *is defined as* $SR[G] = \langle\!\langle S, s_0, \twoheadrightarrow, \twoheadrightarrow_\omega \rangle\!\rangle$ *where*

- S *is a finite set of states,* $S = N \times \mathcal{R}_k^C$
- $s_0 \in S$ *is the initial state,* $s_0 = \langle\!\langle n_0, [\lambda c.0] \rangle\!\rangle \in S$
- $\twoheadrightarrow \subseteq S \times Act \times [0,1] \times S$ *is a probabilistic transition function, where there is a transition*
$$\langle\!\langle n, \xi \rangle\!\rangle \overset{a,p}{\twoheadrightarrow} \langle\!\langle n', \xi' \rangle\!\rangle$$
if there exists $n, n' \in N$ *s.t.* $n \overset{a,w,b,C'}{\longrightarrow} n'$ *and* p *is*
$$p = \pi(\langle\!\langle n, \xi \rangle\!\rangle, a, \langle\!\langle n', \xi' \rangle\!\rangle) \overset{\text{def}}{=} \begin{cases} \frac{\sum \{\!|w \mid \exists b, C'.\ n \overset{a,w,b,C'}{\longrightarrow} n' \wedge b(\xi) \wedge \xi' = \xi[C' \leftarrow 0]|\!\}}{W_{(n,\xi)}} & ;\ W_{(n,\xi)} \neq 0 \\ 0 & ;\ otherwise \end{cases}$$
where
$$W_{(n,\xi)} = \sum \{\!|\ w \mid \exists a, b, C', m.\ n \overset{a,w,b,C'}{\longrightarrow} m \wedge b(\xi) |\!\}$$
- $\twoheadrightarrow_\omega \subseteq S \times S$ *is a timed transition relation where*
$$\langle\!\langle n, \xi \rangle\!\rangle \twoheadrightarrow_\omega \langle\!\langle n, succ(\xi) \rangle\!\rangle$$
if
$$\sum \{\!|\ w \mid \exists a, b, C'.n \overset{a,w,b,C'}{\twoheadrightarrow} n' \wedge b(\xi) \wedge a \in Act_{urg}\ |\!\} = 0$$

Notation: By $\langle\!\langle n, \xi \rangle\!\rangle \overset{i}{\twoheadrightarrow}_\omega \langle\!\langle n, \xi^i) \rangle\!\rangle$ we mean i consecutive successor transitions.

We now present the Correspondence Lemma which establishes an agreement between the configuration graph for a TPG G and its finite representation in terms of the corresponding state-region graph $SR[G]$.

Lemma 4.1 Correspondence Lemma 1
Let $G = \langle N, n_0, C, \longrightarrow \rangle$ *be a TPG, and let* $SR[G] = \langle\!\langle S, s_0, \twoheadrightarrow, \twoheadrightarrow_\omega \rangle\!\rangle$ *be the state-region graph for* G*. Then*

1. *If* $\langle\!\langle n, \xi \rangle\!\rangle \overset{a,p}{\twoheadrightarrow} \langle\!\langle n', \xi' \rangle\!\rangle$ *then for each* $\gamma \in \xi$*, there exist a unique* $\gamma' \in \xi'$ *s.t.* $\langle n, \gamma \rangle \overset{a,p}{\longrightarrow} \langle n', \gamma' \rangle$ *in* $\mathcal{C}[\![G]\!]$
2. *For each* $\gamma, \gamma' \in \mathbf{R}^C$*, if* $\langle n, \gamma \rangle \overset{a,p}{\longrightarrow} \langle n', \gamma' \rangle$ *in* $\mathcal{C}[\![G]\!]$ *then there exists a transition* $\langle\!\langle n, [\gamma] \rangle\!\rangle \overset{a,p}{\twoheadrightarrow} \langle\!\langle n', [\gamma'] \rangle\!\rangle$ *in* $SR[G]$

3. If $\langle\!\langle n, \xi \rangle\!\rangle \twoheadrightarrow_\omega \langle\!\langle n, \xi' \rangle\!\rangle$ then for each $\gamma \in \xi$ there exists a uniqe $\gamma' \in \xi'$ s.t. $\langle n, \gamma \rangle \xrightarrow{\epsilon(m(\gamma))} \langle n, \gamma' \rangle$ in $\mathcal{C}[\![G]\!]$ and $\forall d \in [0, m(\gamma)]$ it holds that $\gamma + m(\gamma) \in \xi \cup \xi'$ where

$$m(\gamma) \overset{\text{def}}{=} \begin{cases} f; & \forall c.\gamma(c) \leq k \Rightarrow \{\gamma(c)\} > 0 \\ f/2; & \exists c.\gamma(c) \leq k \wedge \{\gamma(c)\} = 0 \end{cases}$$

and f is defined as

$$f = 1 - max\{\{\gamma(c)\} \mid \gamma(c) \leq k\}$$

4. If $\langle n, \gamma \rangle \xrightarrow{\epsilon(t)} \langle n, \gamma' \rangle$ in $\mathcal{C}[\![G]\!]$ then there exists a transition sequence $\langle\!\langle n, [\gamma] \rangle\!\rangle \twoheadrightarrow^*_\omega \langle\!\langle n, [\gamma'] \rangle\!\rangle$ in $SR[G]$

PROOF:

1. Assume $\langle\!\langle n, \xi \rangle\!\rangle \overset{a,p}{\twoheadrightarrow} \langle\!\langle n', \xi' \rangle\!\rangle$ and $\gamma \in \xi$. That is, $\pi(\langle\!\langle n, \xi \rangle\!\rangle, a, \langle\!\langle n', \xi' \rangle\!\rangle) = p$ where $p > 0$. Therefore there must be transitions in G, $n \xrightarrow{a,w,b,C'} n'$ s.t. $b(\xi)$ og $\xi' = \xi[C' \leftarrow 0]$. Since $b(\xi) = b(\gamma)$ we must have that $W_{(n,\xi)} = W_{(n,\gamma)}$. This causes $\pi(\langle n, \gamma \rangle, a, \langle n', \gamma' \rangle) = p$ where $p > 0$ i $\mathcal{C}[\![G]\!]$. This give that $\langle n, \gamma \rangle \xrightarrow{a,p} \langle n', \gamma' \rangle$ where $\gamma' \in \xi'$

2. Assume $\langle n, \gamma \rangle \xrightarrow{a,p} \langle n', \gamma' \rangle$. That is, $\pi(\langle n, \gamma \rangle, a, \langle n', \gamma' \rangle) = p$ where $p > 0$. Then there must be transitions in G, $n \xrightarrow{a,w,b,C'} n'$ such that $b(\gamma)$ and $\gamma' = \gamma[C' \leftarrow 0]$. Since $b(\gamma) = b([\gamma])$ we must have that $W_{(n,\gamma)} = W_{(n,[\gamma])}$. Therefore it must be the case that $\pi(\langle\!\langle n, [\gamma] \rangle\!\rangle, a, \langle\!\langle n', [\gamma'] \rangle\!\rangle) = p$ and thereby there exists $\langle\!\langle n, [\gamma] \rangle\!\rangle \overset{a,p}{\twoheadrightarrow} \langle\!\langle\ n', [\gamma'] \rangle\!\rangle$

3. Assume $\langle\!\langle n, \xi \rangle\!\rangle \twoheadrightarrow_\omega \langle\!\langle n, \xi' \rangle\!\rangle$ and $\gamma \in \xi$. This implies that the following holds for G: $\sum\{\!|\ w \mid \exists a, b, C'.\, n \xrightarrow{a,w,b,C'} n' \wedge b(\xi) \wedge a \in Act_{urg}\ |\!\} = 0$ and that succ(ξ)=ξ'. This must therefore also be the the case for $\mathcal{C}[\![G]\!]$ and there must be a time transition $\langle n, \gamma \rangle \xrightarrow{\epsilon(m(\gamma))} \langle n, \gamma' \rangle$ where $\gamma' = \gamma + m(\gamma)$. And for all $a \in Act_{urg}$ and $d < m(\gamma)$ it holds that $\pi(\langle n, \gamma \rangle, a, \langle n, \gamma + d \rangle) = 0$.

4. Assume $\langle n, \gamma \rangle \xrightarrow{\epsilon(t)} \langle n, \gamma + t \rangle$ in $\mathcal{C}[\![G]\!]$. This implies that the following holds for G $\sum\{\!|\ w \mid n \xrightarrow{a,w,b,C'} n' \wedge b(\gamma) \wedge a \in Act_{urg}\ |\!\} = 0$ and $\gamma' = \gamma + t$. This causes SR[G] to have $\langle\!\langle n, [\gamma] \rangle\!\rangle \overset{t_\xi}{\twoheadrightarrow}_\omega \langle\!\langle n, [\gamma + t] \rangle\!\rangle$, where $t_\xi \in \mathbf{N}$ is the number of regions to traverse in order to delay t in $SR[G]$

□

4.1.1 Representing regions

In this section we present a representation of regions useful in relation to implementation concerns and useful as an abstract language in which we can talk about regions.

According to our definition, a region is an equivalence class over time assignments. In definition 4.1 we can explicitly "read" the kind of information necessary to make a representation of a region over some clock set and some maximal constant. The information is as follows:

- The integral part of the clocks.
- Clocks whose value has exceeded the maximal constant.
- Clocks whose fractional part equals 0.
- The mutual ordering of the fractional parts of the clocks.
- The mutual distance between any pair of clocks.

We can now define 5 substructures each representing one of the items above, s.t. together the substructures define a complete representation of a region.

Let $\gamma \in \mathbf{R}^C_{\geq 0}$ be a time assignment over the clock set $C = \{c_1, \dots, c_l\}$ and let $\xi = [\gamma] \in \mathcal{R}^C_k$ denote a region over C with respect to the maximal constant k. We now define the substructures as follows.

Definition 4.4 *We define a list I_ξ denoting the integral parts of all clocks in ξ as follows:*

$$I_\xi \overset{\text{def}}{=} [n_1, \dots, n_l] \quad \textit{where} \quad \forall i.\, 1 \leq i \leq l.\, n_i \overset{\text{def}}{=} \begin{cases} \lfloor \gamma(c_i) \rfloor & \textit{if } \lfloor \gamma(c_i) \rfloor \leq k \\ k & \textit{otherwise} \end{cases}$$

Definition 4.5 *O_ξ is the subset of clocks in C whose value has exceeded k.*

$$O_\xi \overset{\text{def}}{=} \{c \in C \mid \gamma(c) > k\}$$

We define a set Z_ξ denoting the set of clocks having fractional part 0. When clocks exceed the maximal constant, their fractional part is no longer of any interest with respect to $\doteq$ and therefore we do not include clocks contained in O_ξ.

Definition 4.6 *The set Z_ξ is the set of clocks having fractional part 0.*

$$Z_\xi \overset{\text{def}}{=} \{c \in C \mid \{\gamma(c)\} = 0\} \backslash O_\xi$$

We define the list F_ξ of subsets of clocks denoting the ordering of the fractional parts of the clocks. Clocks having the same fractional parts are grouped in the same subset. Clocks contained in $O_\xi \cup Z_\xi$ are not part of F_ξ.

Definition 4.7 *The list F_ξ of subsets of clocks denoting the ordering of the fractional parts of the clocks is defined as.*

$$F_\xi = [C_1, C_2, \ldots, C_l]$$

$$\text{where} \begin{cases} C_i \subseteq C \backslash (O_\xi \cup Z_\xi) \text{ and} \\ C_i \cap C_j = \emptyset \text{ for } i \neq j \text{ and} \\ \bigcup_{i=1}^{l} C_i = C \backslash (O_\xi \cup Z_\xi) \text{ and} \\ \forall i.\ \forall c_1, c_2 \in C_i.\ \{\gamma(c_1)\} = \{\gamma(c_2)\} \text{ and} \\ \forall i < j.\ \forall c_1 \in C_i\ \forall c_2 \in C_j.\ \{\gamma(c_1)\} < \{\gamma(c_2)\} \end{cases}$$

We define the ordered list D_ξ containing for each pair of clocks in C the mutual integral distance between the clocks. If the distance between any pair of clocks exceeds the maximal constant k, the distance will be kept at k.

Definition 4.8 *The list D_ξ is defined as follows.*

$$D_\xi \stackrel{\text{def}}{=} [L_{(1,2)}, L_{(1,3)}, \ldots, L_{(l-1,l)}]$$

where $\forall i.\, 1 \leq i < l.\, \forall j.\, i < j \leq l.\ L_{(i,j)} \stackrel{\text{def}}{=} [c_i, c_j, x, y]$ *where:*

$$x = \begin{cases} |\lfloor \gamma(c_i) - \gamma(c_j) \rfloor| & \text{if } |\lfloor \gamma(c_i) - \gamma(c_j) \rfloor| \leq k \\ k & \text{otherwise} \end{cases}$$

$$y = \begin{cases} 0 & \text{if } |\{\gamma(c_i) - \gamma(c_j)\}| = 0 \text{ and } |\{\gamma(c_i) - \gamma(c_j)\}| \leq k \\ 1 & \text{otherwise} \end{cases}$$

Now, we are finally able to define the representation of the region ξ.

Definition 4.9 *The representation of the region ξ is*

$$\xi \stackrel{\text{def}}{=} (I_\xi,\ O_\xi,\ Z_\xi,\ F_\xi,\ D_\xi)$$

Example 4.2 Figure 4.1 page 34 illustrates five different regions over a clock-set $C = \{c_1, c_2\}$. For each of the regions the corresponding representation is illustrated below

$$\begin{aligned} \xi_1 &= ([0,0], \{\}, \{c_1, c_2\}, [\,], [[c_1, c_2, 0, 0]]) \\ \xi_2 &= ([0,0], \{\}, \{\}, [\{c_1, c_2\}], [[c_1, c_2, 0, 0]]) \\ \xi_3 &= ([0,0], \{\}, \{\}, [\{c_2\}, \{c_1\}], [[c_1, c_2, 0, 1]]) \\ \xi_4 &= ([1,0], \{c_1\}, \{\}, [\{c_2\}], [[c_1, c_2, 0, 1]]) \\ \xi_5 &= ([2,1], \{c_1, c_2\}, \{\}, [], [[c_1, c_2, 1, 1]]) \end{aligned}$$

□

In the sequel we will take a closer look at how the successor function defined in Definition 4.2 actually "works" by examining it in the context of the newly defined representation of regions.

Given a region ξ, the definition of $succ(\xi)$ is divided into two cases. One where the ξ is an interior region and one where ξ is boundary. In both cases the delay length from ξ to $succ(\xi)$ depends on the maximal fractional part of the clocks.

In the case where ξ is interior, $succ(\xi)$ will be boundary and the integral parts of the clocks with maximal fractional part in ξ are increased with 1, whereas the fractional parts for the same set of clocks are set to 0. The mutual distances between all pairs of clocks are unchanged.

In the opposite case, $succ(\xi)$ is an interior region and none of the clocks change their integral parts. Also, none of the clocks will have a zero fractional part in $succ(\xi)$, but the mutual ordering of the fractional parts remains unchanged and likewise the mutual distances between any pairs of clocks.

Considering the representation $\xi = (I_\xi,\ O_\xi,\ Z_\xi,\ F_\xi,\ D_\xi)$ we can obtain $succ(\xi)$ simply by cyclically moving all clocks in the two substructures Z_ξ and F_ξ one step to the right. If ξ is an interior region, that is if $Z_\xi = \emptyset$, this implies that the maximal element of F_ξ is moved into the Z_ξ set and all integral parts of I_ξ corresponding to the clocks in the maximal element of F_ξ are increased by one.

If ξ is a boundary region, that is $Z_\xi \neq \emptyset$, then all elements in Z_ξ are moved to the right, thereby becoming the least element in F_ξ. Similarly all elements of F_ξ are moved one place to the right. If the value of any clocks exceeds the maximal constant, these clocks are added to the O_ξ set and accordingly they are removed from Z_ξ and F_ξ.

Following the steps described above, delaying one time unit will consist of a sequence of cycles moving all elements of Z_ξ and F_ξ one place to the right until the clocks initially situated in Z_ξ are there again. How many regions it takes to delay one time unit of course depends on the number of elements in the structures Z_ξ and F_ξ. This number is denoted $\parallel \xi \parallel$ and defined as:

$$\parallel \xi \parallel \overset{\text{def}}{=} |\, F_\xi \,| + \begin{cases} 1 & \text{if } Z_\xi \neq \emptyset \\ 0 & \text{if } Z_\xi = \emptyset \end{cases}$$

Example 4.3 Consider the region ξ_3 of Figure 4.1 page 34. To delay one time unit, we pass through regions as follows:

$$\begin{aligned}
& ([0,0],\{\},\{\},[\{c_2\},\{c_1\}],[[c_1,c_2,0,1]]) \\
\hookrightarrow\quad & ([1,0],\{\},\{c_1\},[\{c_2\}],[[c_1,c_2,0,1]]) \\
\hookrightarrow\quad & ([1,0],\{c_1\},\{\},[\{c_2\}],[[c_1,c_2,0,1]]) \\
\hookrightarrow\quad & ([1,1],\{c_1\},\{c_2\},[\,],[[c_1,c_2,0,1]]) \\
\hookrightarrow\quad & ([1,1],\{c_1,c_2\},\{\},[\,],[[c_1,c_2,0,1]])
\end{aligned}$$

□

4.2 The Proof System $\vdash$

In this section we present the proof system $\vdash$ forming the basis of the algorithm for model checking TPG's with respect to formulas of RTPL$^-$.

In section 4.1 we showed that given a TPG G we can construct a finite representation of the configuration graph $\mathcal{C}[\![G]\!]$. We will show that this finite representation, denoted $SR[G]$, has the property that if it is extended with respect to some formula clock set K and some maximal constant k_F, in a way defined below, then all extended states in $E\mathcal{C}[\![G]\!]$ corresponding to the same state in the extended $SR[G]$, will satisfy the same formulas of RTPL$^-$ as we can prove the mentioned extended state to "satisfy" in the proof system $\vdash$.

Consider some TPG G and consider some subset of RTPL$^-$ formulas with formula clock set K and maximal constant k_F. We now perform model checking with respect to the extended $SR[G]$ defined as follows.

Definition 4.10 *Let $G = \langle N, n_0, C, \longrightarrow\rangle$ be a TPG with maximal constant k_G. Consider the formula clock set K and maximal constant k_F. Also let $C^+ = C \cup K$ and $k = max(k_G, k_F)$. We let $\xi \in \mathcal{R}_k^{C^+}$ denote an extended region and whenever ξ is a region over C^+ we denote by $\xi\,|_C$ the set of time assignments in ξ restricted to the TPG clock set C. Similar, $\xi\,|_K$ denotes the set of time assignments in ξ restricted to the formula clock set K. Now the extended state-region graph for G with respect to K and k is defined as $ESR[G,K,k] = \langle\langle E, e_0, \twoheadrightarrow^+, \twoheadrightarrow_\omega^+\rangle\rangle^+$ where*

- *E is a finite set of states, $E = N \times \mathcal{R}_k^{C^+}$*
- *$e_0 \in S$ is the initial state, $e_0 = \langle\langle n_0, [\lambda c.0]\rangle\rangle^+ \in E$*
- *$\twoheadrightarrow^+ \subseteq E \times Act \times [0,1] \times E$ is a probabilistic transitions function where*

$$\langle\langle n,\xi\rangle\rangle^+ \overset{a,p}{\twoheadrightarrow}{}^+ \langle\langle n',\xi'\rangle\rangle^+ \text{ iff } \langle\langle n,\xi\,|_C\rangle\rangle \overset{a,p}{\twoheadrightarrow} \langle\langle n',\xi'\,|_C\rangle\rangle \wedge \xi'\,|_K = \xi\,|_K$$

- *$\twoheadrightarrow_\omega^+ \subseteq E \times E$ is a timed transition realation where*

$$\langle\langle n,\xi\rangle\rangle^+ \twoheadrightarrow_\omega^+ \langle\langle n, succ(\xi)\rangle\rangle^+ \text{ iff } \langle\langle n,\xi\,|_C\rangle\rangle \twoheadrightarrow_\omega \langle\langle n, succ(\xi\,|_C)\rangle\rangle$$

We can now present the proof system $\vdash$ over extended symbolic states. The statements of $\vdash$ are of the form $\vdash \langle\langle n,\xi\rangle\rangle^+ : F$, which means that it is proven in $\vdash$ that the extended symbolic state $\langle\langle n,\xi\rangle\rangle^+$ can be marked with the formula F of RTPL^{nf}. The proof system is as follows.

tt
$$\frac{}{\vdash \langle\!\langle n,\xi\rangle\!\rangle^{+} : tt}$$

$c_1 + x \sim c_2 + y$
$$\frac{}{\vdash \langle\!\langle n,\xi\rangle\!\rangle^{+} : c_1 + x \sim c_2 + y} \quad (c_1 + x \sim c_2 + y)(\xi)$$

c_1 in F
$$\frac{\vdash \langle\!\langle n,\xi[c_1 \leftarrow 0]\rangle\!\rangle^{+} : F}{\vdash \langle\!\langle n,\xi\rangle\!\rangle^{+} : c_1 \text{ in } F}$$

$F_1 \wedge F_2$
$$\frac{\vdash \langle\!\langle n,\xi\rangle\!\rangle^{+} : F_1 \quad \vdash \langle\!\langle n,\xi\rangle\!\rangle^{+} : F_2}{\vdash \langle\!\langle n,\xi\rangle\!\rangle^{+} : F_1 \wedge F_2}$$

$F_1 \vee F_2$
$$\frac{\vdash \langle\!\langle n,\xi\rangle\!\rangle^{+} : F_1}{\vdash \langle\!\langle n,\xi\rangle\!\rangle^{+} : F_1 \vee F_2} \qquad \frac{\vdash \langle\!\langle n,\xi\rangle\!\rangle^{+} : F_2}{\vdash \langle\!\langle n,\xi\rangle\!\rangle^{+} : F_1 \vee F_2}$$

$\langle a\rangle_{\geq p}F$
$$\frac{\{\vdash \langle\!\langle n',\xi'\rangle\!\rangle^{+} : F \mid \langle\!\langle n',\xi'\rangle\!\rangle^{+} \in S\}}{\vdash \langle\!\langle n,\xi\rangle\!\rangle^{+} : \langle a\rangle_{\geq p}F} \quad \pi^{+}(\langle\!\langle n,\xi\rangle\!\rangle^{+}, a, S) \geq p$$

$\langle a\rangle_{>p}F$
$$\frac{\{\vdash \langle\!\langle n',\xi'\rangle\!\rangle^{+} : F \mid \langle\!\langle n',\xi'\rangle\!\rangle^{+} \in S\}}{\vdash \langle\!\langle n,\xi\rangle\!\rangle^{+} : \langle a\rangle_{>p}F} \quad \pi^{+}(\langle\!\langle n,\xi\rangle\!\rangle^{+}, a, S) > p$$

$[a]F$
$$\frac{\{\vdash \langle\!\langle n',\xi'\rangle\!\rangle^{+} : F \mid \langle\!\langle n',\xi'\rangle\!\rangle^{+} \in S\}}{\vdash \langle\!\langle n,\xi\rangle\!\rangle^{+} : [a]F} \quad S = \{\langle\!\langle n',\xi'\rangle\!\rangle^{+} \mid \langle\!\langle n,\xi\rangle\!\rangle \overset{a,p}{\twoheadrightarrow}{}^{+} \langle\!\langle n',\xi'\rangle\!\rangle^{+}\}$$

$\forall\!\!\!\forall F$
$$\frac{\{\vdash \langle\!\langle n,\xi^{i}\rangle\!\rangle^{+} : F \mid \langle\!\langle n,\xi\rangle\!\rangle^{+} \overset{i}{\twoheadrightarrow}{}^{+}_{\omega} \langle\!\langle n,\xi^{i}\rangle\!\rangle^{+}, 0 \leq i \leq l_{\xi}\}}{\vdash \langle\!\langle n,\xi\rangle\!\rangle^{+} : \forall\!\!\!\forall F}$$

$\exists\!\!\!\exists F$
$$\frac{\vdash \langle\!\langle n,\xi^{i}\rangle\!\rangle^{+} : F}{\vdash \langle\!\langle n,\xi\rangle\!\rangle^{+} : \exists\!\!\!\exists F} \quad \begin{array}{l} \langle\!\langle n,\xi\rangle\!\rangle^{+} \overset{i}{\twoheadrightarrow}{}^{+}_{\omega} \langle\!\langle n,\xi^{i}\rangle\!\rangle^{+} \\ 0 \leq i \leq l_{\xi} \end{array}$$

In the sequel we proof that the proof system $\vdash$ is indeed sound and complete.

Theorem 4.1 Soundness

$$\vdash \langle\!\langle n,\xi\rangle\!\rangle^{+} : F \Rightarrow \forall \gamma\mu \in \xi.\ \langle n,\gamma\mu\rangle^{+} \models F$$

PROOF:
Proof by induction in the size of the proof for $\vdash \langle\!\langle$n,$\xi\rangle\!\rangle^{+}$: F. We assume that $\vdash \langle\!\langle n,\xi\rangle\!\rangle^{+} : F$ has been proven with a proof of size $n+1$. Now, depending on the proof rule last used, we must show that this implies $\langle n,\gamma\mu\rangle^{+} \models F$ for all $\gamma\mu \in \xi$.

We proceed with the following IH : $\vdash \langle\!\langle m, \eta \rangle\!\rangle^+ : G \Rightarrow \forall \gamma\mu \in \eta.\ \langle m, \gamma \rangle^+ \models G$ has been proven for a proof of size n

Basis : The size of the proof, $n = 0$.

$\underline{tt - rule}$: Trivial since all processes satisfy tt.

$\underline{(c_1 + x \sim c_2 + y) - rule}$: Follows from the region definition and the fact that x, y are integers and $\sim \in \{\leq, <, =, >, \geq\}$.

Step :

$\underline{\wedge - rule}$: Let $F = F_1 \wedge F_2$ and assume $\vdash \langle\!\langle n, \xi \rangle\!\rangle^+ : F$ has been proven by using the $\wedge$-rule with a proof of size $n+1$. Then the proof rule $\wedge$ gives, that it has been proven that $\vdash \langle\!\langle n, \xi \rangle\!\rangle^+ : F_1$ and $\vdash \langle\!\langle n, \xi \rangle\!\rangle^+ : F_2$ with proofs of size at most n. By IH it now follows that $\langle n, \gamma\mu \rangle^+ \models F_1$ and $\langle n, \gamma\mu \rangle^+ \models F_2$ and this together with the definition of $\models$ gives that $\langle n, \gamma\mu \rangle^+ \models F$

$\underline{\vee - rule}$: Let $F = F_1 \vee F_2$ and assume $\vdash \langle\!\langle n, \xi \rangle\!\rangle^+ : F$ has been proven by using the left $\vee$-rule with a proof of size $n+1$. Then by the left $\vee$ we know that $\vdash \langle\!\langle n, \xi \rangle\!\rangle^+ : F_1$ is proven with a proof of size at most n. According to IH, it now follows that $\langle n, \gamma\mu \rangle^+ \models F_1$. Then by the definition of $\models$ we have that $\langle n, \gamma\mu \rangle^+ \models F$. The case where the right $\vee$-rule is used is similar.

$\underline{\langle a \rangle_{\geq p} - rule}$: Assume that $F = \langle a \rangle_{\geq p} F'$ and $\vdash \langle\!\langle n, \xi \rangle\!\rangle^+ : F$ is proven with a proof of size $n+1$ using the $\langle a \rangle_{\geq p}$-rule. Then by the $\langle a \rangle_{\geq p}$-rule it has been proven with proofs of size at most n that $\vdash \langle\!\langle n', \xi' \rangle\!\rangle^+ : F'$ for all $\langle\!\langle n', \xi' \rangle\!\rangle^+ \in S$ where S is some set satifying $\pi^+(\langle\!\langle n, \xi \rangle\!\rangle^+, a, S) \geq p$. By IH it then follows that $\langle n', \gamma'\mu \rangle^+ \models F'$ for those $\gamma'\mu \in \xi'$ where $\langle\!\langle n', \xi' \rangle\!\rangle^+ \in S$. Now let $S^* = \{\langle n', \gamma'\mu \rangle^+ \mid \langle\!\langle n, [\gamma'\mu] \rangle\!\rangle^+ \in S\}$ it then holds that $\pi^+(\langle n, \gamma\mu \rangle^+, a, S^*) = \pi(\langle\!\langle n, \xi \rangle\!\rangle^+, a, S)$ for all $\gamma\mu \in \xi$. This together with the definition of $\models$ gives that $\langle n, \gamma\mu \rangle^+ \models F$

$\underline{\langle a \rangle_{> p} - rule}$: As for the $\langle a \rangle_{\geq p} - rule$

$\underline{[a] - rule}$: Let $F = [a]F'$ and assume $\vdash \langle\!\langle n, \xi \rangle\!\rangle^+ : F$ is proven using the $[a]$-rule with a proof of size $n+1$. Then by the proof rule $[a]$ it is proven with proofs of size at most n that $\vdash \langle\!\langle n', \xi' \rangle\!\rangle^+ : F'$ whenever $\langle\!\langle n, \xi \rangle\!\rangle^+ \overset{a,p}{\twoheadrightarrow}{}^{+} \langle\!\langle n', \xi' \rangle\!\rangle^+$. Then by IH and by Correspondence Lemma (1) it follows that for $\gamma\mu \in \xi$, $\langle n', \gamma'\mu \rangle^+ \models F'$ whenever $\langle n, \gamma\mu \rangle^+ \xrightarrow{a,p} \langle n', \gamma'\mu \rangle^+$. This together with the definition of $\models$ gives that $\langle n, \gamma\mu \rangle \models F$

$\underline{\forall\!\!\forall - rule}$: Let $F = \forall\!\!\forall F'$ and assume $\vdash \langle\!\langle n, \xi \rangle\!\rangle^{+} : F$ has been proven with a proof of size $n+1$. Then by the $\forall\!\!\forall$-rule we have proven with proofs of size at most n that

$$\{\vdash \langle\!\langle n, \xi^{i} \rangle\!\rangle^{+} : F' \mid \langle\!\langle n, \xi \rangle\!\rangle^{+} \overset{i}{\longrightarrow\!\!\!\!\twoheadrightarrow}{}^{+}_{\omega} \langle\!\langle n, \xi^{i} \rangle\!\rangle^{+}, 0 \leq i \leq l_{\xi}\}$$

Let S be the above set. Then by IH we can construct a set $S^{*} = \{\langle n, \gamma'\mu' \rangle^{+} \models F' \mid \gamma'\mu' \in \xi^{i} \wedge (\vdash \langle\!\langle n, \xi^{i} \rangle\!\rangle^{+} : F') \in S\}$. Now from the definition of $\longrightarrow\!\!\!\!\twoheadrightarrow^{+}_{\omega}$ we know that there is a transition $\longrightarrow\!\!\!\!\twoheadrightarrow^{+}_{\omega}$ in the *ESR*-graph iff there is a transition $\longrightarrow\!\!\!\!\twoheadrightarrow_{\omega}$ in the *SR*-graph. This together with Correspondence Lemma (3) and the definition of extended configuration graph now implies: $\langle n, \gamma\mu \rangle^{+} \xrightarrow{\epsilon(t)} \langle n, \gamma + t\mu + t \rangle^{+}$ for $t \geq 0$ and $\langle n, \gamma + t\mu + t \rangle^{+} \in S^{*}$. It now follows by definition of $\models$ that $\langle n, \gamma\mu \rangle^{+} \models F$

$\underline{\exists\!\!\exists - rule}$: As above.

$\underline{c_1 \text{ in } F - rule}$: Let $F = c_1$ in F' and assume $\vdash \langle\!\langle n, \xi \rangle\!\rangle^{+} : F$ has been proven in a proof of size $n+1$. Then by the proof rule we have that $\vdash \langle\!\langle n, \xi[c_1 \leftarrow 0] \rangle\!\rangle^{+} : F'$. By IH we can construct a set $S = \{\langle n, \gamma\mu' \rangle^{+} \models F' \mid \gamma\mu' \in \xi[c_1 \leftarrow 0]\}$. Now by definition of $\models$ we know: $\forall \gamma\mu \in \xi . \langle n, \gamma\mu \rangle^{+} \models F$ □

Theorem 4.2 Completeness

$$\langle n, \gamma\mu \rangle^{+} \models F \Rightarrow \vdash \langle\!\langle n, [\gamma\mu] \rangle\!\rangle^{+} : F$$

Proof: Proof by induction in the structure of the formula F with following IH: $\langle n, \gamma\mu \rangle^{+} \models F \Rightarrow \vdash \langle\!\langle n, [\gamma\mu] \rangle\!\rangle^{+} : F$ when F is a subformula of the actual formula.

Basis :

$\underline{F = tt}$: Trivial since $\vdash \langle\!\langle n, [\gamma] \rangle\!\rangle^{+} : tt$ is an axiom

$\underline{F = c_1 + x \sim c_2 + y}$: Follows from the definition of regions.

Step :

$\underline{F = F_1 \wedge F_2}$: Assume $\langle n, \gamma\mu \rangle^{+} \models F_1 \wedge F_2$. Then by definition of $\models$ we have that $\langle n, \gamma\mu \rangle^{+} \models F_1$ and $\langle n, \gamma\mu \rangle^{+} \models F_2$. Then according to IH we get $\vdash \langle\!\langle n, [\gamma\mu] \rangle\!\rangle^{+} : F_1$ and $\vdash \langle\!\langle n, [\gamma\mu] \rangle\!\rangle^{+} : F_2$. By the proof rule for $\wedge$ we now have that $\vdash \langle\!\langle n, [\gamma\mu] \rangle\!\rangle : F_1 \wedge F_2$

$\underline{F = F_1 \vee F_2}$: Assume $\langle n, \gamma\mu \rangle^{+} \models F_1 \vee F_2$. According to the definition of $\models$ we have that $\langle n, \gamma\mu \rangle^{+} \models F_1$ or $\langle n, \gamma\mu \rangle^{+} \models F_2$. Assume that $\langle n, \gamma\mu \rangle^{+} \models F_1$.

Now by IH we know that $\vdash \langle\!\langle n, [\gamma\mu]\rangle\!\rangle^{+} : F_1$ and by the proof rules for $\vee$ we have that $\vdash \langle\!\langle n, [\gamma\mu]\rangle\!\rangle : F_1 \vee F_2$. The case where $\langle n, \gamma\mu\rangle^{+} \models F_2$ follows as above.

$\underline{F' = \langle a\rangle_{\geq p}F}$: Assume $\langle n, \gamma\mu\rangle^{+} \models \langle a\rangle_{\geq p}F$, then by definition of $\models$ we have

$$\sum_{\{\langle n', \gamma'\mu\rangle^{+} \mid \langle n', \gamma'\mu\rangle^{+} \models F\}} \pi(\langle n, \gamma\mu\rangle^{+}, a, \langle n', \gamma'\mu\rangle^{+}) \geq p$$

For all $\langle n', \gamma'\mu\rangle^{+}$ belonging to the above mentioned index set we know, by IH, that the following holds: $\vdash \langle\!\langle n', [\gamma'\mu]\rangle\!\rangle^{+} : F$. By the Correspondance Lemma 1 (2) we have that for a TPG G there is a transition in $SR[G]$ for each transition in $\mathcal{C}[\![G]\!]$. Using this fact and the proof rule for $\langle a\rangle_{\geq p}$ we have that $\vdash \langle\!\langle n, [\gamma\mu]\rangle\!\rangle^{+} : F$

$\underline{F' = \langle a\rangle_{>p}F}$: As above.

$\underline{F' = [a]F}$: Assume $\langle n, \gamma\mu\rangle^{+} \models [a]F$. Then by the definition of $\models$ we have the following

$$\forall \langle n', \gamma'\mu\rangle^{+}.\ \langle n, \gamma\mu\rangle^{+} \xrightarrow{a,p} \langle n', \gamma'\mu\rangle^{+} \text{ and } p > 0 \Rightarrow \langle n', \gamma'\mu\rangle \models F$$

Then by IH we have that: $\langle n', \gamma'\mu\rangle^{+} \models F \Rightarrow \vdash \langle\!\langle n', [\gamma'\mu]\rangle\!\rangle^{+} : F$. By the Correspondence Lemma 1 (2) we have that each transition $\langle n, \gamma\rangle \xrightarrow{a,p} \langle n', \gamma'\rangle$ in $\mathcal{C}[\![G]\!]$ is matched by a transition $\langle\!\langle n, [\gamma]\rangle\!\rangle \overset{a,p}{\twoheadrightarrow} \langle\!\langle n', [\gamma]'\rangle\!\rangle$ in $SR[G]$. This together with the proof rule for $[a]F$ gives that $\vdash \langle\!\langle n, [\gamma\mu]\rangle\!\rangle^{+} : [a]F$

$\underline{F' = \forall\!\!\!\forall F}$: Assume $\langle n, \gamma\mu\rangle^{+} \models F'$. Then by definition of $\models$ we get $\forall d \in \mathbf{R}_{\geq 0}.\ \langle n, \gamma\mu\rangle^{+} \xrightarrow{\epsilon(d)} \langle n, \gamma + d\mu + d\rangle^{+}$ and $\langle n, \gamma + d\mu + d\rangle^{+} \models F$. By IH we get $\vdash \langle\!\langle n, [\gamma + d\mu + d]\rangle\!\rangle^{+} : F$. Because every transition $\langle n, \gamma\rangle \xrightarrow{\epsilon(d)} \langle n, \gamma + d\rangle$ is matched by transitions $\langle n, \gamma\mu\rangle^{+} \xrightarrow{\epsilon(d)} \langle n, \gamma + d\mu + d\rangle^{+}$, we can use the Correspondence Lemma (4) to conclude $\langle\!\langle n, [\gamma\mu]\rangle\!\rangle^{+} \overset{i}{\twoheadrightarrow}{}^{+}_{\omega} \langle\!\langle n, [\gamma\mu]^{i}\rangle\!\rangle^{+}$, where $0 \leq i \leq l_{[\gamma\mu]}$ and $\vdash \langle\!\langle n, [\gamma\mu]^{i}\rangle\!\rangle^{+} : F$. Hence we have that $\vdash \langle\!\langle n, [\gamma\mu]^{i}\rangle\!\rangle^{+} : F'$

$\underline{F' = c_1 \text{ in } F}$: Assume $\langle n, \gamma\mu\rangle^{+} \models F'$. Then by the definition of $\models$ we have that $\langle n, \gamma\mu[c_1 \leftarrow 0]\rangle^{+} \models F$. According to IH we then have $\vdash \langle\!\langle n, [\gamma\mu[c_1 \leftarrow 0]]\rangle\!\rangle^{+} : F$, and by the proof rule for c_1 in F we can conclude that $\vdash \langle\!\langle n, [\gamma\mu]\rangle\!\rangle^{+} : F'$ □

Corrollary 4.1 *Let G be a TPG and $\mathcal{F}$ a set of RTPL^{nf} formulas with formula clock set K. Then for two states $\langle n, \nu\rangle^{+}, \langle n, \nu'\rangle^{+} \in EC[\![G, K]\!]$ s.t. $\nu \overset{\bullet}{=} \nu'$ we have:*

$$\forall F \in \mathcal{F}.\ \langle n, \nu\rangle^{+} \models F \Leftrightarrow \langle n, \nu'\rangle^{+} \models F$$

PROOF: Assume $\nu \overset{\bullet}{=} \nu'$ and $\langle n, \nu\rangle^{+} \models F$. Now, the corollary follows from Theorem 4.1 and Theorem 4.2. □

Theorem 4.3 *Model checking TPG's with respect to formulas of RTPL$^-$ is decidable.*

PROOF: All tableaux in the proof system for $\langle\!\langle n, \xi \rangle\!\rangle^+ : F$ are finite due to the fact that the extended state-region graph has finite branching and all formulas of RTPL$^-$ are structurally finite. Also, there are only finitely many tableaux for $\langle\!\langle n, \xi \rangle\!\rangle^+ : F$ due to the finite branching of the extended state-region graph. □

4.3 The Proof System $\vdash_{weak}$

In this section we present the proof system $\vdash_{weak}$ forming the basis of the algorithm for model checking TPG's with respect to formulas of RTPL$_{weak}$.

We want to use the same strategy for the construction of the proof system $\vdash_{weak}$ as presented in section 4.2. That is, given a TPG G and a formula $F \in$ RTPL$_{weak}$ with formula clock set K, we want to construct our proof system over states of the extended state-region graph $ESR[G, K, k]$ where k is the maximal constant over F and G.

To construct proof rules for the weak modalities of RTPL$_{weak}$, we need the notions of a τ-abstracted *succ*-transition and a τ-abstracted action transition on the extended state-region graph. Where τ, as usual represents urgent actions due to internal communication.

But a problem occurs when considering the definition of the τ-abstracted *succ* transition. Because τ-actions can be resolved in several different ways a $\xrightarrow{\epsilon(d)}$ transition of timed probabilistic process may require different numbers of *succ*-transitions. This is due to the fact, that for a region ξ, the number of fractional part classes $\|\,\xi\,\|$ will determine the number of *succ*-transitions needed to delay a certain amount of time. This was discussed in section 4.1.1. The problem is illustrated on Figure 4.2.

Performing the left τ-transition in $P_0^{\xi_0}$ leads to the new start region ξ_1, in which 4 *succ*-transitions are required to delay exactly one time unit. Whereas performing the right τ-transition in $P_0^{\xi_0}$ leads to the new start region ξ_2, in which only 2 *succ*-transitions are required to delay the one time unit.

Using the representation of regions presented in section 4.1.1, we can see how the *succ*-transitions change the elements Z and F in the representations of ξ_1 and ξ_2 as these delay. The Z and F elements are the ones determining the number of fractional part classes. (See section 4.1.1).

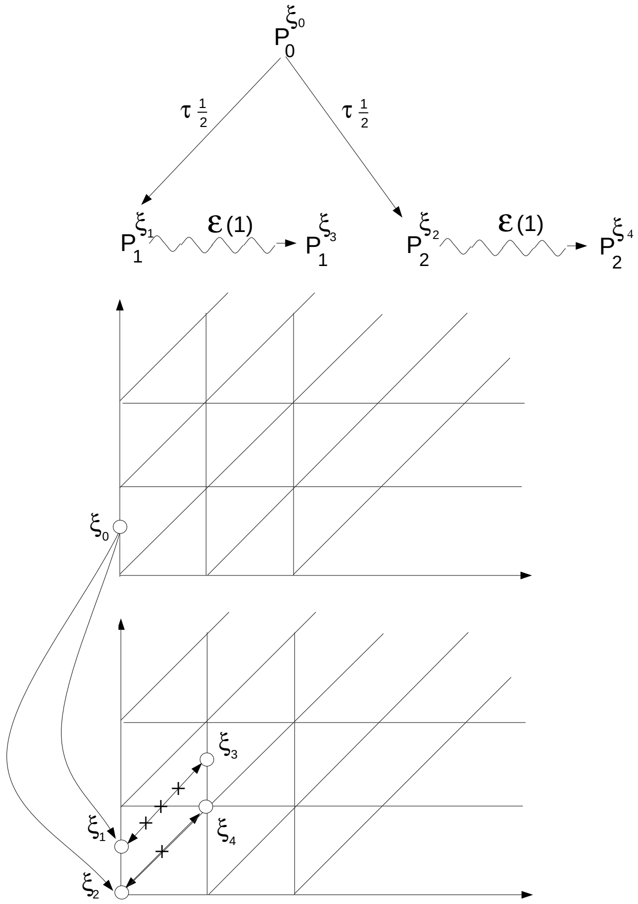


Figure 4.2: *Different number of successors*

In the region ξ_1 the number of fractional part classes $\parallel \xi_1 \parallel$ is 2 and in ξ_2 we have $\parallel \xi_2 \parallel = 1$. Now, the Z and F elements change as follows:

		Z	F			Z	F
ξ_1 :		$\{c_1\}$	$[\{c_2\}]$	ξ_2 :		$\{c_1, c_2\}$	
	$\hookrightarrow$	$\{\}$	$[\{c_1\}, \{c_2\}]$		$\hookrightarrow$	$\{\}$	$[\{c_1, c_2\}]$
	$\hookrightarrow$	$\{c_2\}$	$[\{c_1\}]$	ξ_4 :	$\hookrightarrow$	$\{c_1, c_2\}$	
	$\hookrightarrow$	$\{\}$	$[\{c_2\}, \{c_1\}]$				
ξ_3 :	$\hookrightarrow$	$\{c_1\}$	$[\{c_2\}]$				

To solve the problem illustrated above, we need to keep the number of fractional parts classes constant during *succ*-sequences and thereby ensuring that it always takes the same amount of *succ*-transitions to delay some time unit no matter how τ-transitions are resolved.

The change in the number of fractional part classes is caused by the resetting of some clocks in the clock set concurrently with a τ-transition. In the above example the τ-transition leading to region ξ_2 resets the clock c_2 and thereby decreases the number of fractional part classes. Obviously, not all τ-transitions will lead to a decrease of this number, e.g. the τ-transition from the example leading to ξ_1 does not reset any clocks at all and hence does not change the number of fractional part classes.

It is important to note that τ-actions are only enabled in boundary regions i.e. regions where at least one clock in the clock set has fractional part zero (the set Z is non-empty). Thus, the resetting of some clocks concurrently with a τ-transition will never increase the number of fractional part classes.

To keep the number of fractional part classes constant, we introduce the notion of *hybrid extended symbolic states* and *hybrid transitions*. The idea is that, when the resetting of some clock set concurrently with a τ-transition leads to a reduction in the number of fractional part classes, we simply introduce some hybrid clock set and then expand the region part of an extended symbolic state to be a region over the old clock set joined with the new hybrid clock set. Furthermore, the new hybrid clocks are introduced in a way that preserves the ordering of the existing clocks. This actually means that the fractional parts of the introduced hybrid clocks "fill the holes" in the fractional part list created by the resetting of some clocks.

If we look at the example from before, this leads to the following observations of the elements Z and F in the region part of the hybrid extended symbolic state χ_2 obtained from the resetting of c_2.

$$\begin{array}{llll}
 & & Z & F \\
\chi_2 : & & \{c_1, c_2\} & [\{c_3^h\}] \\
 & \hookrightarrow & \{\} & [\{c_1, c_2\}, \{c_3^h\}] \\
 & \hookrightarrow & \{c_3^h\} & [\{c_1, c_2\}] \\
 & \hookrightarrow & \{\} & [\{c_3^h\}, \{c_1, c_2\}] \\
\chi_4 : & \hookrightarrow & \{c_1, c_2\} & [\{c_3^h\}]
\end{array}$$

Now if we remove the hybrid clock from χ_4 we get ξ_4 and the delay of one time unit has taken exactly 4 *succ*-transitions.

When a hybrid clock is introduced, its integral part is set to 0 and as its fractional part is different from the fractional parts of any other clocks, the integral distance of the hybrid clock to each other clock is set to the integral part of these clocks, and the fractional distance is set to be non-zero.

Formally a hybrid extended symbolic state is defined as follows.

Definition 4.11 *Let n be some TPG state and let C be a clock set $C = \{c_1, \dots, c_n\}$. Furthermore, let $H \subseteq \{h_1, \dots, h_{n-1}\}$ be some set of hybrid clocks s.t. $C \cap H = \emptyset$. Then a hybrid extended symbolic state is a pair $\langle\langle n, \chi\rangle\rangle^+$ where $\chi \in \mathcal{R}^{C \cup H}$.*

On these hybrid states we can now define the notion of τ-abstracted *succ*-transitions. But first we need some preliminary definitions.

Definition 4.12 *Let $I_\chi = [n_1, \dots, n_k]$ be the integral part of a region $\chi \in \mathcal{R}^C$ and let $C' \subseteq \{c_j \mid c_j \in C\}$ be a subset of C. The reset integral part $I_\chi[C' \leftarrow 0]$ obtained from I_χ by resetting the clocks in C' is defined by:*

$$I_\chi[C' \leftarrow 0] \overset{\text{def}}{=} [m_1, \dots, m_k]$$

$$\textit{where } \forall i.\, 1 \leq i \leq k.\, m_i = \begin{cases} 0 & \textit{if } c_i \in C' \\ n_i & \textit{otherwise} \end{cases}$$

Definition 4.13 *Let $F_\chi = [C_1, \dots, C_l]$ be the fractional part of some region $\chi \in \mathcal{R}_k^{C^+}$ where $C^+ = C \cup H$ for some clock set C and hybrid clock set H s.t. $C \cap H = \emptyset$. Furthermore, let $R \subseteq C$ denote some reset set. The reset fractional part $\Delta(F_\chi, R, H)$ where the clocks in R have been removed and where the hybrid clocks in H are introduced to keep the number of classes in F_χ constant, is defined by:*

$$\Delta(F_\chi, R, H) \overset{\text{def}}{=} [D_1, \dots, D_l] \textit{ where } \forall i.\, 1 \leq i \leq l.\, D_i = \begin{cases} \{c_j\} & \textit{if } C_i \backslash R = \emptyset \\ & \textit{s.t. } c_j \in H \\ C_i & \textit{otherwise} \end{cases}$$

Each c_j is inserted in the order determined by j.

Definition 4.14 *Let D_χ be the distance part of some region $\chi \in \mathcal{R}^C$ where $C = \{c_1, \dots, c_n\}$. Furthermore, let $H = \{h_l, \dots, h_m\}$ denote a set of hybrid clocks. The reset distance part $\nabla(D_\chi, H)$ is defined by:*

$$\nabla(D_\chi, H) \overset{\text{def}}{=} D_\chi :: [L_{(1,l)}, \dots, L_{(n,m)}]$$

$$\textit{where } \forall j.\, 1 \leq j \leq n.\, \forall k.\, j \leq k \leq m.\, L_{(j,k)} \overset{\text{def}}{=} [c_j, h_k, \lfloor c_j \rfloor, 1]$$

The hybrid transitions can now be defined as follows.

Definition 4.15 *Let $G = \langle N, n_0, C, \longrightarrow \rangle$ be a TPG with maximal constant k_G and let K be a formula clock set and k_F a maximal constant. Furthermore, let $C^+ = C \cup K$ and $k = max(k_G, k_F)$. Finally, let E be the extended state-region graph $E = ESR[G, K, k]$ and let $\langle\langle n, \chi\rangle\rangle^+$ be a hybrid extended symbolic state with $\chi = (I_\chi, O_\chi, Z_\chi, F_\chi, D_\chi) \in \mathcal{R}_k^{C^+ \cup H}$ for some hybrid clock set H. By $\chi|_{C^+}$ we denote the region obtained by removing all hybrid clocks. Furthermore, we* *let* $\tau \in Act^{com}_{urg}$, $\alpha \in Act \backslash Act^{com}_{urg}$ *and* $R \subseteq C$. *We now define the following hybrid transitions:*

$$\langle\!\langle n, \chi \rangle\!\rangle^+ \xrightarrow{\tau,p}{}_h \langle\!\langle n', \rho(\chi) \rangle\!\rangle^+ \quad \mathit{iff} \quad \langle\!\langle n, \chi|_{C^+} \rangle\!\rangle^+ \overset{\tau,p}{\twoheadrightarrow} \langle\!\langle n', \rho(\chi)|_{C^+} [R \leftarrow 0] \rangle\!\rangle^+$$

$$\mathit{where} \begin{cases} \rho(\chi) & \overset{\text{def}}{=} \quad (\rho(I_\chi), \rho(O_\chi), \rho(Z_\chi), \rho(F_\chi), \rho(D_\chi)) \\ \rho(I_\chi) & \overset{\text{def}}{=} \quad I_\chi[R \leftarrow 0] :: L \\ \rho(O_\chi) & \overset{\text{def}}{=} \quad O_\chi \\ \rho(Z_\chi) & \overset{\text{def}}{=} \quad Z_\chi \cup R \\ \rho(F_\chi) & \overset{\text{def}}{=} \quad \Delta(F_\chi, R, H') \\ \rho(D_\xi) & \overset{\text{def}}{=} \quad \nabla(D_\chi, H') \end{cases}$$

$$\mathit{and} \begin{cases} L & = \begin{cases} \emptyset & \mathit{if}\ m = 0 \\ [n_{|C^+|+1}, \dots, n_{|C^+|+m}] & \mathit{if}\ m > 0 \end{cases} \\ H' & = \{c_{|C^+|+1}, \dots, c_{|C^+|+m}\} \\ m & = \parallel \chi|_{C^+} \parallel - \parallel \chi|_{C^+} [R \leftarrow 0] \parallel \end{cases}$$

$$\langle\!\langle n, \chi \rangle\!\rangle^+ \xrightarrow{\alpha,p}{}_h \langle\!\langle n', \chi'[R \leftarrow 0] \rangle\!\rangle^+ \ \mathit{iff}\ \langle\!\langle n, \chi|_{C^+} \rangle\!\rangle^+ \overset{\alpha,p}{\twoheadrightarrow} \langle\!\langle n', \chi|_{C^+} [R \leftarrow 0] \rangle\!\rangle^+$$

$$\langle\!\langle n, \chi \rangle\!\rangle^+ \longrightarrow_{h\omega} \langle\!\langle n, succ(\chi) \rangle\!\rangle^+ \ \mathit{iff}\ \langle\!\langle n, \chi|_{C^+} \rangle\!\rangle^+ \twoheadrightarrow^+_\omega$$

Given the above notion of hybrid transitions, we can now define the notion of τ abstracted hybrid transitions. In the definitions below we let $\mathcal{H}, \mathcal{H}'$ denote hybrid extended symbolic states.

Definition 4.16

1. $\mathcal{H} \xrightarrow{\tau,p}{}^*_h \mathcal{H}' \quad \mathit{if} \quad \sum \{\!| \ p' \mid \exists n. \mathcal{H} \xrightarrow{\epsilon,1}{}_h \xrightarrow{\tau,p_1}{}_h \dots \xrightarrow{\tau,p_n} \mathcal{H}' \ \mathit{and}\ p' = \prod_{i \leq n} p_i \ |\!\} = p$

2. $\mathcal{H} \xrightarrow{k}{}_{h\omega} \mathcal{H}' \quad \mathit{if} \quad \exists \mathcal{H}_1, \dots, \mathcal{H}_{k-1}. \mathcal{H} \longrightarrow_{h\omega} \mathcal{H}_1 \longrightarrow_{h\omega} \dots \longrightarrow_{h\omega} \mathcal{H}_{k-1} \longrightarrow_{h\omega} \mathcal{H}'$

Definition 4.17

1. $\mathcal{H} \overset{\tau,p}{\Longrightarrow}{}_h \mathcal{H}' \quad \mathit{if} \quad \mathcal{H} \xrightarrow{\tau,p}{}^*_h \mathcal{H}' \ \mathit{where}\ \tau \in Act^{com}_{urg}$

2. $\mathcal{H} \overset{\alpha,p}{\Longrightarrow}{}_h \mathcal{H}' \quad \mathit{if} \quad \sum \{\!| \ p' \mid \mathcal{H} \xrightarrow{\tau,p_1}{}^*_h \xrightarrow{\alpha,p_2}{}_h \xrightarrow{\tau,p_3}{}^*_h \mathcal{H}' \ \mathit{and}\ p' = \prod_{i \leq 3} p_i \ |\!\} = p$

3. $\mathcal{H} \overset{k,p}{\Longrightarrow}{}_{h\omega} \mathcal{H}' \quad \mathit{if} \quad \sum \{\!| \ p' \mid \mathcal{H} \xrightarrow{\tau,p_1}{}^*_h \xrightarrow{k_1}{}_{h\omega} \xrightarrow{\tau,p_2}{}^*_h \dots \xrightarrow{\tau,p_n}{}^*_h \xrightarrow{k_n}{}_{h\omega} \xrightarrow{\tau,p_{n+1}}{}^*_h \mathcal{H}'$
   $\mathit{and}\ p' = \prod_{i \leq n+1} p_i \ \mathit{and}\ k = \sum_{j \leq n} k_j \ |\!\} = p$

We are now able to present the proof system $\vdash_{weak}$ over the hybrid extended symbolic states. Before we present the final proof system, we will exemplify the construction of the rules by taking a closer look at how the rule for formulas of the type $\boxplus_{\exists p} F$ is constructed. We choose the above rule for examplification as it is one of the least obvious rules.

The conclusion of the above rule is obviously $\vdash_{weak} \langle\!\langle n, \chi \rangle\!\rangle^{+} : \exists\!\boxplus_{\sqsupseteq p} F$ where $\langle\!\langle n, \chi \rangle\!\rangle^{+}$ is an hybrid extended symbolic state. The first time the rule is used we have clearly not made any hybrid extensions yet.

To fulfill the conclusion there must exist some set of states satisfying F and each reachable from $\langle\!\langle n, \chi \rangle\!\rangle^{+}$ by a τ-abstracted hybrid transition such that each transition includes the same number of hybrid successor transitions and such that the sum of the probabilities for the transitions is $\geq p$. This leads to the following premise (or actually set of premises):

$$\{\vdash_{weak} \langle\!\langle n', \chi' \rangle\!\rangle^{+} : F \mid \langle\!\langle n', \chi' \rangle\!\rangle^{+} \in S\}$$

where S is a set s.t. the following side condition is satisfied:

$$\langle\!\langle n, \chi \rangle\!\rangle^{+} \overset{i,p'}{\Longrightarrow}_{h\omega} S \text{ and } p' \sqsupseteq p$$

Now, all that is needed to obtain a complete rule is to add some limits to the natural number i denoting the number of hybrid successor transitions in the τ-abstracted hybrid successor transitions. Obviously, due to the hybrid transitions introduced it will always be the case, that no matter how τ-actions are resolved, delaying a certain time unit will require the same number of hybrid successor transitions. All we need are some lower and upper limits on the number of hybrid successors that we need to examine.

The above number is determined by the TPG we are model checking with respect to. The maximal delay in any τ-abstracted delay sequence is determined by the enabling conditions and reset sets labelling the edges in τ-sequences in the TPG. Obviously, as τ-transitions are taken exactly at their moment of enabling, we can look at the enabling condition and determine at exactly what time the action is taken. Now we can compute the actual delay performed before enabling the τ-action as the difference between the value compared to some clock in the enabling condition and the actual value of this clock when entering the state having the outgoing τ-action. This last value is of course determined by how many times the clock has been reset during the τ-sequence performed until the current state. Adding all the above delays and the possible delay in the last state of the τ-sequence before a not τ-action is enabled, gives the maximal delay in some τ-abstracted sequence. Now, the maximal delay in *any* τ-abstracted delay sequence is the maximum of the above number over all sequences. For some TPG state n the above number is denoted by d_n.

The maximal number of hybrid successors in any τ-abstracted hybrid successor transition from some hybrid extended symbolic state $\langle\!\langle n, \chi \rangle\!\rangle^{+}$ can now be computed as $d_n \cdot \parallel \chi \parallel \cdot 2$. We denote this number by $t_{n,\chi}$.

The proof rule can now be completed as follows:

$$\exists\!\boxplus_{\sqsupseteq p} F \quad \frac{\{\vdash_{weak} \langle\!\langle n', \chi' \rangle\!\rangle^{+} : F \mid \langle\!\langle n', \chi' \rangle\!\rangle^{+} \in S\}}{\vdash_{weak} \langle\!\langle n, \chi \rangle\!\rangle^{+} : \exists\!\boxplus_{\sqsupseteq p} F} \quad \begin{array}{l} \langle\!\langle n, \chi \rangle\!\rangle^{+} \overset{i,p'}{\Longrightarrow}_{h\omega} S \text{ and} \\ 0 \leq i \leq t_{n,\chi},\ p' \sqsupseteq p \end{array}$$

Proving the system sound and complete relies on a correspondence between weak transitions of hybrid extended symbolic states and weak transitions of extended configuration graphs. This correspondence can be stated as the following lemma, which is very similar to the Lemma 4.1 stated in section 4.1.

Lemma 4.2 *(Correspondence Lemma 2)*

Let $G = \langle N, n_0, C, \longrightarrow \rangle$ be a TPG and let $ESR[G, K, k]$ be the extended state-region graph for G with respect to some formula clock set K and maximal constant k. Let $\chi \in \mathcal{R}^{C^+ \cup H}$ where $C^+ = C \cup K$ is a clock set over the TPG clocks C and the formula clocks K. Furthermore, H is some hybrid clock set. Now, for $\alpha \in Act \backslash Act_{urg}^{com}$ and $\tau \in Act_{urg}^{com}$ the following holds.

1. *If* $\langle\langle n, \chi \rangle\rangle^+ \overset{a,p}{\Longrightarrow}_h \langle\langle n', \chi' \rangle\rangle^+$ *then*
   $\forall \gamma\mu \in \chi|_{C^+} \,.\, \pi_w^+(\langle n, \gamma\mu\rangle^+, a, \langle n', \gamma'\mu\rangle^+) = p$ *where* $\gamma'\mu \in \chi'|_{C^+}$

2. *If* $\pi_w^+(\langle n, \gamma\mu\rangle^+, a, \langle n', \gamma'\mu\rangle^+) = p$ *then*
   $\langle\langle n, \chi \rangle\rangle^+ \overset{a,p}{\Longrightarrow}_h \langle\langle n', \chi' \rangle\rangle^+$ *where* $\exists \eta, \eta' \in \mathbf{R}^H .\, \chi = [\gamma\mu\eta],\ \chi' = [\gamma'\mu\eta']$

3. *If* $\langle\langle n, \chi \rangle\rangle^+ \overset{k,p}{\Longrightarrow}_h \langle\langle n', \chi' \rangle\rangle^+$ *then*
   $\forall \gamma\mu \in \chi|_{C^+} \,.\, \exists d \in \mathbf{R}_{\geq 0} .\, \langle n, \gamma\mu\rangle^+ \overset{\epsilon(d),p}{\Longrightarrow} \langle n', \gamma'\mu'\rangle^+$ *where* $\gamma'\mu' \in \chi|_{C^+}$

4. *If* $\langle n, \gamma\mu\rangle^+ \overset{\epsilon(d),p}{\Longrightarrow}{}^+ \langle n', \gamma'\mu'\rangle$ *then*
   $\langle\langle n, \chi \rangle\rangle^+ \overset{k,p}{\Longrightarrow}_h \langle\langle n', \chi' \rangle\rangle^+$ *where* $\exists \eta, \eta' \in \mathbf{R}^H .\, \chi = [\gamma\mu\eta],\ \chi' = [\gamma'\mu\eta']$

Proof:

1. Assume $\langle\langle n, \chi \rangle\rangle^+ \overset{a,p}{\Longrightarrow}_h \langle\langle n', \chi' \rangle\rangle^+$. Now we show that in general:

   (a) $\langle\langle n, \chi \rangle\rangle^+ \xrightarrow{\tau,p}{}^*_h \langle\langle n', \chi' \rangle\rangle^+ \Rightarrow \forall \gamma\mu \in \chi|_{C^+} \,.\, \langle n, \gamma\mu\rangle^+ \overset{\tau,p}{\Longrightarrow}{}^+ \langle n', \gamma'\mu\rangle^+$ where $\gamma'\mu \in \chi'|_{C^+}$.

   (b) $\langle\langle n, \chi \rangle\rangle^+ \xrightarrow{a,p}_h \langle\langle n', \chi' \rangle\rangle^+ \Rightarrow \forall \gamma\mu \in \chi|_{C^+} \,.\, \langle n, \gamma\mu\rangle^+ \overset{a,p}{\longrightarrow}{}^+ \langle n', \gamma'\mu\rangle^+$ where $\gamma'\mu \in \chi'|_{C^+}$.

   Having shown (a) and (b) now gives the result.

   (a) : The proof is by induction in the length of $\xrightarrow{\tau,p}{}^*_h$.

   k=0 : Trivial, since all states satisfies $\langle n, \gamma\mu\rangle^+ \overset{\epsilon,1}{\longrightarrow}{}^+ \langle n, \gamma\mu\rangle^+$.

   k+1 : We assume the following IH:

   $$\langle\langle n, \chi \rangle\rangle^+ \xrightarrow{\tau,p_k}{}^k_h \langle\langle n_k, \chi_k \rangle\rangle^+ \Rightarrow \forall \gamma\mu \in \chi|_{C^+} \,.\, \langle n, \gamma\mu\rangle^+ \overset{\tau,p_k}{\Longrightarrow}{}^+ \langle n_k, \gamma_k\mu\rangle^+$$

   where $\gamma_k\mu \in \chi_k|_{C^+}$.

   Furthermore, we assume that in the k+1 step $\langle\langle n_k, \chi_k \rangle\rangle^+ \xrightarrow{\tau,p_{k+1}}_h \langle\langle n_{k+1}, \chi_{k+1} \rangle\rangle^+$. From Definition 4.15 we know that for some reset set $R \subseteq C$ ($C^+ = C \cup K$)

$\langle\!\langle n_k, \chi_k \,|_{C^+} \rangle\!\rangle^+ \overset{\tau,p_{k+1}}{\twoheadrightarrow} \langle\!\langle n_{k+1}, \chi_k \,|_{C^+} [R \leftarrow 0] \rangle\!\rangle^+$. Due to the Correspondence Lemma 1 and the definition of transitions between extended states we know that for all $\gamma_k\mu \in \chi_k \,|_{C^+}$ there exists a unique $\gamma'_k\mu \in \chi_k \,|_{C^+}$ $[R \leftarrow 0]$ s.t. $\langle n_k, \gamma_k\mu\rangle^+ \overset{\tau,p_k+1}{\longrightarrow^+} \langle n_{k+1}, \gamma'_k\mu\rangle^+$. Furthermore, we know that χ_{k+1} is the region obtained by extending the region $\chi_k \,|_{C^+} [R \leftarrow 0]$ with a set of hybrid clocks. But because $\gamma'_k\mu$ is an assignment over C^+ then obviously $\gamma'_k\mu \in \chi_{k+1}$. Now, the result follows from IH.

(b) : Follows from Definition 4.15 and Correspondence Lemma 1 (1).

The rest of the cases are omitted as they are quite like the above. □

Now, the proof system $\vdash_{weak}$ is the following set of rules.

tt
$$\frac{}{\vdash_{weak} \langle\!\langle n, \chi \rangle\!\rangle^+ : tt}$$

$c_1 + x \sim c_2 + y$
$$\frac{}{\vdash_{weak} \langle\!\langle n, \chi \rangle\!\rangle^+ : c_1 + x \sim c_2 + y} \quad (c_1 + x \sim c_2 + y)(\chi)$$

c_1 in F
$$\frac{\vdash_{weak} \langle\!\langle n, \chi[c_1 \leftarrow 0] \rangle\!\rangle^+ : F}{\vdash_{weak} \langle\!\langle n, \chi \rangle\!\rangle^+ : c_1 \text{ in } F}$$

$F_1 \wedge F_2$
$$\frac{\vdash_{weak} \langle\!\langle n, \chi \rangle\!\rangle^+ : F_1 \quad \vdash_{weak} \langle\!\langle n, \chi \rangle\!\rangle^+ : F_2}{\vdash_{weak} \langle\!\langle n, \chi \rangle\!\rangle^+ : F_1 \wedge F_2}$$

$F_1 \vee F_2$
$$\frac{\vdash_{weak} \langle\!\langle n, \chi \rangle\!\rangle^+ : F_1}{\vdash_{weak} \langle\!\langle n, \chi \rangle\!\rangle^+ : F_1 \vee F_2} \qquad \frac{\vdash_{weak} \langle\!\langle n, \chi \rangle\!\rangle^+ : F_2}{\vdash_{weak} \langle\!\langle n, \chi \rangle\!\rangle^+ : F_1 \vee F_2}$$

$\langle\!\langle a \rangle\!\rangle_{\sqsupseteq p} F$
$$\frac{\{\vdash_{weak} \langle\!\langle n', \chi' \rangle\!\rangle^+ : F \mid \langle\!\langle n', \chi' \rangle\!\rangle^+ \in S\}}{\vdash_{weak} \langle\!\langle n, \chi \rangle\!\rangle^+ : \langle\!\langle a \rangle\!\rangle_{\geq p} F} \quad \langle\!\langle n, \chi \rangle\!\rangle^+ \overset{a,p'}{\Longrightarrow}{}_h S,\ p' \sqsupseteq p$$

$[\![a]\!] F$
$$\frac{\{\vdash_{weak} \langle\!\langle n', \chi' \rangle\!\rangle^+ : F \mid \langle\!\langle n', \chi' \rangle\!\rangle^+ \in S\}}{\vdash_{weak} \langle\!\langle n, \chi \rangle\!\rangle^+ : [\![a]\!] F} \quad S = \{\langle\!\langle n', \chi' \rangle\!\rangle^+ \mid \langle\!\langle n, \chi \rangle\!\rangle \overset{a,p}{\Longrightarrow}{}_h \langle\!\langle n', \chi' \rangle\!\rangle^+\}$$

$\mathbb{W}_{\sqsupseteq p} F$
$$\frac{\{\vdash_{weak} \langle\!\langle n', \chi' \rangle\!\rangle^+ : F \mid \langle\!\langle n', \chi' \rangle\!\rangle^+ \in S\}}{\vdash_{weak} \langle\!\langle n, \chi \rangle\!\rangle^+ : \mathbb{W}_{\sqsupseteq p} F} \quad S = \bigcup_{0 \leq i \leq t_{n,\chi}} \{S_i \mid \langle\!\langle n, \chi \rangle\!\rangle^+ \overset{i,p'}{\Longrightarrow}{}_h S_i,\ p' \sqsupseteq p\}$$

$\mathbb{E}_{\sqsupseteq p} F$
$$\frac{\{\vdash_{weak} \langle\!\langle n', \chi' \rangle\!\rangle^+ : F \mid \langle\!\langle n', \chi' \rangle\!\rangle^+ \in S\}}{\vdash_{weak} \langle\!\langle n, \chi \rangle\!\rangle^+ : \mathbb{E}_{\sqsupseteq p} F} \quad \begin{array}{l} \langle\!\langle n, \chi \rangle\!\rangle^+ \overset{i,p'}{\Longrightarrow}{}_h S \text{ and} \\ 0 \leq i \leq t_{n,\chi},\ p' \sqsupseteq p \end{array}$$

We omit proving soundness and completeness of this inference system as the proofs are exactly similar to those of Theorem 4.1 and Theorem 4.2, respectively.

4.4 Ideas on model checking RTPL$_{until}$

In this section we present ideas for model checking the logic RTPL$^{nf}_{until}$ without negation. The presentation will be informal as it still needs some further attention to become a complete formal theory.

When model checking formulas of RTPL$^{nf}_{until}$ we would like to use the same approach as used for both RTPL^{nf} and RTPL$_{weak}$. That is, we want to use a proof systematic approach, where conclusions for the until operators are of the form $\vdash_u \langle\!\langle n, \xi\rangle\!\rangle : F_1 EU^{\leq t}_{\geq p} F_2$ or $\vdash_u \langle\!\langle n, \xi\rangle\!\rangle : F_1 AU^{\leq t}_{\geq p} F_2$ where $\langle\!\langle n, \xi\rangle\!\rangle^+$ denotes a state of an extended state region graph. In the sequel we will present the model checking ideas for the negation free RTPL$_{until}$ by examining the construction of proof rules for the until formula $F_1 EU^{\leq t}_{\geq p} F_2$. A complete proof system for all negation free RTPL$_{until}$ formulas will be a simple extension.

Assume an extended state $\langle n, \gamma\mu\rangle^+$ of a TPG with respect to some formula clock set satisfies a formula of the type $F_1 EU^{\leq t}_{\geq p} F_2$. Then, by the semantics of $\models_u$ it is required that there exists a resolved probabilistic transition system originating in the node $\langle n, \gamma\mu\rangle^+$ such that the sum of probabilities of all paths in the resolved transition system leading to states satisfying F_2 within time limit t such that all intermediate states satisfies F_1, is greater than or equal to p.

Now, to construct proof rules for the above formula we must find some way of interpreting a resolved probabilistic transition system in the context of extended state region graphs.

The interpretation is quite simple. What we do, when considering a resolved probabilistic transition system, is simply to resolve all delay transitions in the original timed probabilistic transition system. Therefore, all we need to do to obtain what we can call a *resolved extended state-region graph* is to resolve all successor transitions in the underlying extended state-region graph. That is, for any state of the resolved extended state-region graph there is a unique number of consecutive successor transitions originating from that state.

Now, to fulfill the conclusion $\vdash_u \langle\!\langle n, \xi\rangle\!\rangle^+ : F_1 EU^{\leq t}_{\geq p} F_2$ where $\langle\!\langle n, \xi\rangle\!\rangle^+$ is a state in an ordinary unresolved extended state-region graph, we must search for the existence of a resolved extended graph such that the semantics of the until-formula is satisfied. Obviously when considering a particular symbolic state of the extended state-region graph this search amounts to either traversing the set of action transitions originating from the state or traversing some unique number of consecutive successor transitions. Performing either of the above two steps will lead to a new set of states that are included in the premises of the proof rule having the conclusion mentioned first in this paragraph.

As usual the new formulas occuring in the premises of rules are subformulas of the formula in the conclusion. The same holds for formulas of the until type. In the sequel we discuss the two types of subformulas obtained from either choosing a set of successor transitions in the original node or choosing the set of action transitions.

4.4.1 Successor transitions

Obviously, choosing some number of consecutive successor transitions will lead to a state where a subformula of the original $F_1 EU^{\leq t}_{\geq p} F_2$ formula must hold. Having performed the successor transitions means that time has passed and therefore in the new state a formula $F_1 EU^{\leq t'}_{\geq p} F_2$ must hold in which the upper time limit t' is the value of t decreased by the amount of time corresponding to the number of successor transitions traversed. Furthermore, in all intermediate states encountered during the successor sequence it must be the case that F_1 holds.

Now, the above strategy is quite obvious but there exists a problem. Given the number of successor transitions traversed we can not (exactly) determine the actual delay performed. We can only observe that this delay lies in between certain limits. Furthermore, trying to use this strategy could naturally lead to the specification of a non-integer time limit. As a solution to this problem we would like to construct a *symbolic until formula* where the upper limit is no longer an actual time value but a number of successor transition to traverse. We conjecture that it is possible to construct such a formula and show its equivalence to an ordinary until-formula. If t is the upper limit of some until formula we must be able to compute some number $F(t)$ denoting the exact maximal number of successor transitions to be traversed when searching for the resolved extended state-region graph. The symbolic formula will in the sequel be denoted $F_1\ sEU^{\leq F(t)}_{\geq p} F_2$.

Just as was the case when model checking the weak modalities, we encounter the problem of actions resetting clocks, thereby changing the number of fractional part classes and hence changing the number of successor transitions required to delay some certain time unit. Therefore, we must introduce the same notion of hybrid clocks as presented in the model checking of RTPL$_{weak}$. But this is not quite enough. In the case of weak model checking, all we need to ensure is that no matter how τ-actions are resolved then afterwards it always takes the same number of successor transitions to delay some time unit. We do not need to know that it also takes the same number of successor transitions as *before* the τ-actions were resolved. This is due to the fact that all we need to check is whether there exists some delay after which a certain property is satisfied. We do not beforehand have an actual limit on this delay. Therefore, when model checking, we simply take one successor step at a time and then examine whether the property is satisfied.

When model checking the until formula, we are confronted with a certain upper limit t and therefore we must be sure that when transforming t into a number of successor transitions then this number actually determines exactly a delay of t time units no matter how actions are taken, and hence no matter how clocks are reset.

What this discussion leads to is a need for some way of keeping a uniform region partitioning no matter whether some clocks have exceeded the maximal constant. Actually, it is in these situations that the number of fractional parts decrease and hence reduces the number of successor transitions required to delay the same time unit as before the maximal constant was exceeded. Still, we also have the need for the hybrid clocks.

To obtain the uniform region partitioning even when some clocks have exceeded the maximal constant we can introduce some notion of *virtual* regions. The representation of these regions are simply obtained by never removing clocks from the fractional part F in the representation even if clocks exceed the maximal constant. Obviously it is easy to obtain the real region representation from the virtual simply by removing all clocks in the fractional part having exceeded the maximal constant. We can now, given the upper time limit t compute the upper limit $F(t)$ denoting the maximal number of successors to traverse, and furthermore we can be sure that it always takes the same number of successors to delay the same time unit.

To recapitulate, we will perform the model checking of until formulas with respect to some notion of *virtual hybrid extended symbolic states*.

4.4.2 Probabilistic action transitions

Assume, given a virtual hybrid extended symbolic state N that, we choose to traverse the set of action transitions from N. Then it must be the case that each probabilistic derivative N_i of N must satisfy a subformula $\vdash_u N_i : F_1 \, sEU^{\leq k}_{\geq p_i} F_2$ with some probability p_i. Furthermore, assuming the model checking problem in N to be $\vdash_u N : F_1 \, sEU^{\leq k}_{\geq p} F_2$ and assuming the probabilty of taken the action transition to N_i is $\pi(N, N_i) = r_i$ then it must be the case, that $\sum_i r_i \cdot p_i = p$.

Now, of course we can not beforehand know the actual value of the p_i's because they depend on each other and on future derivatives. That is, the p_i's are actually *variables* which can only be assigned values (0 or 1) when the continued application of rules reaches a state where either F_2, $\neg F_1 \wedge \neg F_2$ or $\neg F_2$ holds and there exists no new successor state.

When walking through the proof system this way, we have to carry along the variables p_i. This is done by adding them to a list of hypotheses X. So when we ask $\vdash N : F_1 \, sEU^{\leq k}_{\geq p} F_2$. we build up an inequation system consisting of the introduced variables and when the rule application terminates the inequation system is solved and we know whether or not $\vdash_u N : F_1 \, sEU^{\leq k}_{\geq p} F_2$ holds.

Now as will be seen in the following attempt of a proof system $\vdash_u$ we have rules for both the cases where we either start out with the empty hypotheses list $\emptyset$ and the nonempty list X, p_i (the list X concatenated with the probability p_i).

When we start out with the empty hypotheses list in the rules for $F_1\, sEU^{\leq k}_{\geq p} F_2$, p is the actual *value* and hence in the rule (1) we need not assign any value (0 or 1) to p. Unlike the case where we have a nonempty hypotheses list in the rules for $F_1\ sEU^{\leq t}_{\geq p_i} F_2$. Then p_i is a variable and needs to be assigned either 0 or 1 in rules (6),(7) or (8).

We now present a first attempt to a proof system for the RTPL$_{until}$ logic following the principles discussed above. We only present the rules for the *sEU* modality.

$$(1)\quad \frac{\emptyset \vdash_u N : F_2}{\emptyset \vdash_u N : F_1\ sEU^{\leq k}_{\geq p} F_2}$$

$$(2)\quad \frac{\emptyset \vdash_u N : F_1\ \{\emptyset, p_i \vdash_u N_i : F_1\ sEU^{\leq k}_{\geq p_i} F_2 \mid \pi(N, N_i) = r_i\}}{\emptyset \vdash_u N : F_1\ sEU^{\leq k}_{\geq p} F_2}\ \sum_i r_i p_i \geq p$$

$$(3)\quad \frac{\emptyset \vdash_u N_j : F_1\ sEU^{\leq k'}_{\geq p} F_2\ \ \{\emptyset \vdash_u N_i : F_1 \mid N \xrightarrow{i}{}_{vh\omega} N_i,\ i \leq j\}}{\emptyset \vdash_u N : F_1\ sEU^{\leq k}_{\geq p} F_2}\ \begin{array}{l} N \xrightarrow{j}{}_{vh\omega} N_j \text{ and} \\ k' = k - j \end{array}$$

$$(4)\quad \frac{X, p_i\ \vdash_u N_i : F_1\ \{X, p_i, p_{ij}\ \vdash_u N_{ij} : F_1\ sEU^{\leq k}_{\geq p_{ij}} F_2 \mid \pi(N_i, N_{ij}) = r_{ij}\}}{X, p_i\ \vdash_u N_i : F_1\ sEU^{\leq k}_{\geq p_i} F_2}\ \sum_j r_{ij} p_{ij} \geq p_i$$

$$(5)\quad \frac{X, p_i\ \vdash_u N_j : F_1\ sEU^{\leq k'}_{\geq p_i} F_2\ \ \{X, p_i\ \vdash_u N_i : F_1 \mid N \xrightarrow{i}{}_{vh\omega} N_i,\ i \leq j\}}{X, p_i\ \vdash_u N : F_1\ sEU^{\leq k}_{\geq p_i} F_2}\ \begin{array}{l} N \xrightarrow{j}{}_{vh\omega} N_j \text{ and} \\ k' = k - j \end{array}$$

$$(6)\quad \frac{X, p_i \mapsto 1\ \vdash_u N_i : F_2}{X, p_i\ \vdash_u N_i : F_1\ sEU^{\leq k}_{\geq p_i} F_2}$$

$$(7)\quad \frac{X, p_i \mapsto 0\ \vdash_u N_i : \neg F_2\ \ \vdash_u N_i : \neg F_1}{X, p_i\ \vdash_u N_i : F_1\ sEU^{\leq k}_{\geq p_i} F_2}$$

$$(8)\quad \frac{X, p_i \mapsto 0\ \vdash_u N_i : \neg F_2}{X, p_i\ \vdash_u N_i : F_1\ sEU^{\leq 0}_{\geq p_i} F_2}\ N_i \longrightarrow{}_{vh\omega} N_i$$

Transitions $\longrightarrow_{vh\omega}$ denote successor transitions between virtual hybrid extended symbolic states.

Chapter 5

Model Construction

In this chapter we address the *satisfiability* problem for the logic RTPL. That is: *Given a logical property is it possible to automatically synthesize a satisfying finite TPG (provided any such exists) ?.* This problem is also known as the problem of *model construction.* The satisfiability problems for CTL and the modal μ-calculus have been proven decidable [EC82, EH85, Wol85, KP83] whereas for TCTL and T_μ the same problems are undecidable [ACD90].

In [LLW95] a bounded satisfiability problem is proven decidable for the logic L_ν, and as our logic RTPL has adopted the same concept of formula clocks and delay modalities as presented in L_ν, we can also only prove the same kind of bounded satisfiability for RTPL. That is, we present a model construction algorithm which, given a formula of RTPL and bounds k and M, will synthesize a timed probabilistic graph with no more than k clocks and no clock being compared with constants greater than M. Furthermore, to obtain decidability we require that, in formulas $\langle\alpha\rangle_{\sqsupseteq p}F$, p is a multiple of a given least $q \in \mathbf{Q}_0^+$, s.t. $p \in [0,1]$.

The technique applied in the model construction is based on classical tableau methods applied in modal logic [HC68] and fundamentally it is a *reduction-construction* technique, where initially the given problem is reduced to a set of simple problems on which a construction step is applied. Then, recursively, the reduction-construction step is again applied to a new set of smaller problems with termination guaranteed.

To get hold of the basic ideas in the approach mentioned above, we consider a very simple example of applying the approach in the context of standard HML [Mil89] and traditional labelled transition systems.

Example 5.1 Consider the simple formula $F \equiv \langle a\rangle tt \wedge [a]\langle b\rangle tt$. We want to construct a transition system containing a state P s.t. $P \models F$. First, since P must satisfy $\langle a\rangle tt$, we add a new state Q, and a transition $P \xrightarrow{a} Q$, and the only requirement yet for Q is to satisfy tt, which holds immediately. The other conjunct in the specification of P is $[a]\langle b\rangle tt$. To satisfy this, we must follow all a-transitions from P, and require of each destination state that it satisfies

$\langle b\rangle tt$. There is only one such state, namely Q, and for Q to satisfy $\langle b\rangle tt$, we add a state R and a transition $Q \xrightarrow{b} R$. All requirements are now fulfilled and the construction has finished. $\square$

Fundamentally our algorithm works in the same way as illustrated in the example, allthough it is far more complicated due to the greater expressiveness of RTPL. Also, we obviously do not want to (and cannot) construct the infinite timed probabilistic transition system validating a formula, but instead we want to construct (if possible) a TPG whose initial state satisfies the given formula.

This chapter is organized as follows. In section 5.1 we first introduce the notion of *problems* and satifiability of these. Next we define the reduction of problems into irreducible problems and we show that satisfiability of a problem is the same as satisfiability of one of the irreducible problems. In section 5.2 we present the construction theorem and show decidability of bounded satisfiability.

5.1 Reduction of problems

The general problem of satisfiability that we are facing is: Given a logical property F of RTPL is it possible to automatically synthesize a finite TPG whose initial state satisfies F ?. Now, let us try to define exactly and in a formal sense what this means.

When presented with a formula $F \in$ RTPL that we want to satisfy, all we initially know is that F contains a certain set of formula clocks K and a maximal constant k_F. Now, the obvious first attempt to define the satisfiability problem will be the unbounded problem defined as follows: *Given the formula F with clock set K and maximal constant k_F does there exist a TPG G, some clock set C and some maximal constant M s.t. G satisfies F ?*. In the context of timed automata and the logic L_ν, from which we have adopted the same semantics of formula clocks and delay-modalities, it has been shown [LLW95] that the above unbounded satisfiability problem is undecidable. In general it is known that it is not possible to deduce the number of clocks in the resulting automata from the number of formula clocks in the formula F that we wish to satisfy. Therefore, when asking whether a formula F is satisfiable one also has to specify the type of the possible resulting automaton. That is the clock set and the largest constant being compared to any clock. The satisfiability problem that we are addressing for RTPL is this *bounded satisfiability* problem.

To recapitulate, the bounded satisfiablity problem is as follows: *Given a formula F of RTPL with k_F as maximal constant and K the set of formula clocks occurring in F, and given a set C of TPG clocks and M an integer, does there exist a (C, M)-TPG satisfying F ?*. Here a (C, M)-TPG is a TPG with clock set C and no enabling condition containing constants greater than M. We formally define the notion of a *problem* in the following way.

Definition 5.1 *Let F be an RTPL formula with maximal constant k_F and let K be the formula clocks in F. Also, let C be a set of clocks and M an integer. Now a problem Π is defined as*

$$\Pi \subseteq \mathcal{R}_k^{C \cup K} \times L^F$$

where L^F is the set of all subformulas of F (including F) and $k = max(M, k_F)$.

The formal notion of satisfiability of problems is now defined as follows.

Definition 5.2 *A problem Π is said to be (C,M)-satisfiable if there exists a (C,M)-TPG, G, and a node n of G s.t.*

$$\forall(\xi, F) \in \Pi . \langle\!\langle n, \xi \rangle\!\rangle^+ \vdash F$$

We call G a model of (solution to) Π. Whenever G is a model of an initial problem $([\mathbf{0}], F)$ we write $G \models F$

As illustrated in example 5.1, for standard HML it is the $\langle\,\rangle F$-formulas that initiate the construction of transitions to new states in the resulting model, and the $[\,]F$-formulas initiate a collection of formulas that are required to hold in the newly created states. The same fundamental strategy applies to formulas of RTPL as well, though the formulas initiating respectively the construction and the collection are probabilistic and generalized versions of the corresponding HML-formulas.

In RTPL the formulas that cause edges to be constructed in the (possible) resulting TPG are of the kind $\langle\alpha\rangle_{\sqsupseteq p}F$ where $\sqsupseteq \in \{>, \geq\}$. Though, as we will see later, formulas of the above type do not correspond one-to-one with the construction of edges in the resulting TPG. By this we mean, that each formula $\langle\alpha\rangle_{\sqsupseteq p}F$ does not necessarily give rise to exactly one transition in the resulting TPG. In HML the $[\alpha]F$-formulas initiate the collection of formulas required to hold in newly created states, and in RTPL formulas of the type $\neg\langle\alpha\rangle_{\sqsupseteq p}F$ serve the same purpose. As a special case the RTPL formula $\neg\langle\alpha\rangle_{>0}\neg F$ is equivalent to the HML formula $[\alpha]F$.

Formulas of the above type together with the atomic formula $c_1 + x \sim c_2 + y$ will in the sequel be denoted *simple* problems. As we will see, any problem can be reduced to a set of simple problems with bounded satisfiability maintained.

Definition 5.3 *A problem Π is said to be a simple problem if whenever $(\xi, F) \in \Pi$ then F is of the form*

$$F ::= \langle\alpha\rangle_{\sqsupseteq p}F' \mid \neg\langle\alpha\rangle_{\sqsupseteq p}F' \mid c_1 + x \sim c_2 + y$$

where $F' \in$ RTPL, $\alpha \in Act$, $\sqsupseteq \in \{>, \geq\}$ and $\sim \in \{=, \leq, \geq, <, >\}$.

Thus, in a simple problem we have resolved all conjunctions, disjunctions and delay modalities. In the sequel we provide a reduction relation for transforming problems into simple ones.

Definition 5.4 *Let $\Pi \in \mathcal{R}_k^{C \cup K} \times L^F$ be a problem and let $\xi \in \mathcal{R}_k^{C \cup K}$. The reduction relation $\rightarrow_{red} \subseteq (\mathcal{R}_k^{C \cup K} \times L^F) \times (\mathcal{R}_k^{C \cup K} \times L^F)$ between problems is the least relation satisfying :*

1. $\Pi \uplus \{(\xi, tt)\} \ \rightarrow_{red} \ \Pi$
2. $\Pi \uplus \{(\xi, F_1 \wedge F_2)\} \ \rightarrow_{red} \ \Pi \cup \{(\xi, F_1)\} \cup \{(\xi, F_2)\}$
3. $\Pi \uplus \{(\xi, F_1 \vee F_2)\} \ \rightarrow_{red} \ \Pi \cup \{(\xi, F_1)\}$
4. $\Pi \uplus \{(\xi, F_1 \vee F_2)\} \ \rightarrow_{red} \ \Pi \cup \{(\xi, F_2)\}$
5. $\Pi \uplus \{(\xi, \exists F)\} \ \rightarrow_{red} \ \Pi \cup \{(succ^l(\xi), F)\}$ *where* $0 \leq l \leq l_\xi$
6. $\Pi \uplus \{(\xi, \forall\!\!\!\forall F)\} \ \rightarrow_{red} \ \Pi \cup \{(succ^l(\xi), F) \mid 0 \leq l \leq l_\xi\}$
7. $\Pi \uplus \{(\xi, c \textbf{ in } F)\} \ \rightarrow_{red} \ \Pi \cup \{(\xi[c \leftarrow 0], F)\}$

where $\uplus$ denotes disjoint union of sets. Furthermore, we let $\rightarrow_{red}^*$ denote the reflexive and transitive closure of $\rightarrow_{red}$.

As the use of $\rightarrow_{red}$ always strictly decreases the total size of the formulae in Π, it is clear that any reduction sequence from Π must be finite. In fact any problem determines a finite reduction tree with the leaves being the irreducible reductions of Π. Formally the above is defined as follows:

Definition 5.5 *Let Π be a problem. We say that Π' is an irreducible reduction of Π if:*

$$\Pi \rightarrow_{red}^* \Pi' \quad \textit{and} \quad \Pi' \not\rightarrow_{red}$$

Lemma 5.1 *For all problems Π the set of irreducible reductions for Π is finite.*

Proof: Follows from the fact that each formula F is finite and there are only finitely many regions. Therefore, the axioms 3, 4 and 5 can only be applied finitely many times due to backtracking. The rest of the rules only give rise to exactly one reduction. □

We now know that the reduction of a given problem indeed terminates. Finally, we get that there is a nice connection between the satisfiability of a problem and its irreducible problems:

Lemma 5.2 *A problem Π is satisfiable if and only if there exists an irreducible problem Π' of Π being satisfiable.*

Proof: From the rules of the proof system $\vdash$ it follows immediately that the reduction rules preserves satisfiability in both directions. □

It is clear, from the definition of $\rightarrow_{red}$ that any irreducible problem is either simple or contains a pair of the form (ξ, ff) in which case it is obviously not satisfiable. Thus, we are left with the problem of deciding satisfiability of simple problems.

5.2 Construction

In this section we show that the bounded satisfiability problem for simple problems is indeed decidable. First, we introduce the fundamental ideas in the construction technique by examining the satisfiability for a fragment of RTPL with a very restricted kind of probabilistic quantification. We denote this fragment RTPL^0, and the syntax is:

$$\begin{aligned} F \quad ::= \quad & tt \mid \neg F \mid F_1 \vee F_2 \mid F_1 \wedge F_2 \mid \langle\alpha\rangle_{\ 0}F \mid \exists F \mid \forall F \\ & \mid c_1 \textbf{ in } F \mid c_1 + x \sim c_2 + y \end{aligned}$$

where $\alpha \in Act$; $c_1, c_2 \in K$; $x, y \in \{0, 1, \ldots, k\}$; $\sim \in \{=, \leq, \geq, <, >\}$.

The $\langle\alpha\rangle_{>0}F$ formula is equivalent to the HML formula $\langle\alpha\rangle F$ and in the sequel we will use this notation. Similarly, the formulas of type $\neg\langle\alpha\rangle_{>0}$ will be denoted as their HML equivalent, $[\alpha]F$.

In the above simple setting the construction, if possible, of a satisfying TPG will be relatively easy. In the resulting TPG the actual weights, greater than 0, on the edges will be of no importance with respect to the satisfiability of formulas. Therefore, we simply assume that all weights on edges are equal to 1. Now, given a clock set C, a maximal constant M, a formula clock set K and a maximal formula constant k_F, the simple problems are of the form:

$$(\xi, \langle\alpha\rangle F) \quad \text{and} \quad (\xi, [\alpha]F)$$

where $\xi \in \mathcal{R}_k^{C \cup K}$, $\alpha \in Act$, $F \in \text{RTPL}^0$ and $k = max(M, k_F)$.

Now, whenever $(\xi, \langle\alpha\rangle F)$ is a subproblem of some satisfiable simple problem then this will give rise to the construction of a transition in the resulting TPG. Due to the finitely many clocks C in the TPG, and consequently finite set of enabling conditions, it is possible, for each subproblem, to choose an enabling condition b and reset set $s \subseteq C$ s.t. $b(\xi)$ holds and s.t. $(s(\xi), F)$ is satisfiable. Where $s(\xi)$ denotes the region obtained by resetting the clocks in s. Furthermore, according to how the enabling condition b is chosen, we have that for all regions ξ', being part of subproblems $(\xi', [\alpha]F')$ where $F' \in \text{RTPL}^0$, s.t. $b(\xi')$ holds, it must be the case that $(s(\xi'), F')$ is satisfiable.

The above ideas are illustrated in example 5.2.

Example 5.2 Let Π be the simple problem:

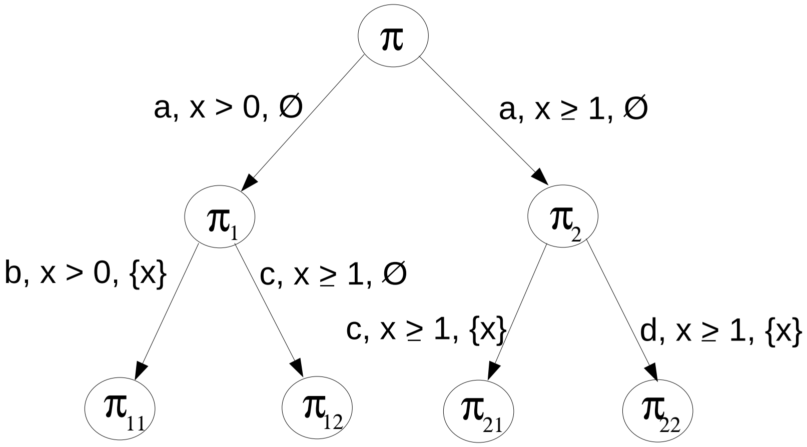
$$\Pi = \{\ (]0;1[, \langle a\rangle tt)\ ,\ (]0;1[, [a]\langle b\rangle tt)\ ,\ ([1;1], [a]\langle c\rangle tt)\ ,\ ([1;1], \langle a\rangle\langle d\rangle tt)\ \}$$

where $]0;1[$ and $[1;1]$ denote regions over *one* TPG clock x.

Now the two subproblems $(]0;1[, \langle a\rangle tt)$ and $([1;1], \langle a\rangle\langle d\rangle tt)$ give rise to the selection of reset sets s_1, s_2 and enabling conditions b_1, b_2, s.t. we obtain two new simple problems Π_1, Π_2 being satisfiable. If we choose $s_1 = s_2 = \emptyset$ and $b_1 = (x > 0)$, $b_2 = (x \geq 1)$ we get

$$\begin{aligned}\Pi_1 &= \{(]0;1[, \langle b\rangle tt)\ ,\ ([1;1], \langle c\rangle tt\} \\ \Pi_2 &= \{([1;1], \langle c\rangle tt)\ ,\ ([1;1], \langle d\rangle tt\}\end{aligned}$$

Now each of Π_1, Π_2 is applied the same strategy as above. Using *one* guessing strategy this yields: $s_{11} = s_{21} = s_{22} = \{x\}$, $s_{12} = \emptyset$, $b_{11} = (x > 0)$ and $b_{12} = b_{21} = b_{22} = (x \geq 1)$, $\Pi_{11} = \Pi_{21} = \Pi_{22} = \{([0;0], tt)\}$ and $\Pi_{12} = \{([1;1], tt)\}$. The resulting TPG is depicted below omitting the weights on edges.



□

The above technique is not directly applicable when considering the general probabilistic formulas of RTPL. The reason that simple problems of the form $(\xi, \langle\alpha\rangle_{\sqsupset p} F)$ do not immediately give rise to the construction of a α-transition with probability p is, that simple problems are, in some sense, not *disjoint*.

Consider e.g. the simple problem

$$\Pi = \{(\xi, \langle a\rangle_{\geq\frac{1}{2}} F_1), (\xi, \langle a\rangle_{\geq\frac{1}{2}} F_2), (\xi, \neg\langle a\rangle_{>\frac{1}{2}} F_1), (\xi, \neg\langle a\rangle_{>\frac{1}{2}} F_2)\}$$

The naive idea in this case would be to construct two transitions, each with weight 1, s.t. the normalized probabilities are both $\frac{1}{2}$, to two new states (problems), where respectively the formulas F_1 and F_2 holds. But this is not quite enough. Obviously, we have to require that the formulas satisfied in the two new states are disjoint; i.e. in the state where F_1 is supposed to hold $\neg F_2$ must also hold, and in the state where F_2 holds $\neg F_1$ must also hold. If these requirements are not satisfied we cannot guarantee the satisfaction of the last two subproblems in Π.

Another way of satisfying the problem would be to construct two transitions each having weights 1 (probabilty $\frac{1}{2}$) to two new states where respectively $F_1 \wedge F_2$ and $\neg F_1 \wedge \neg F_2$ holds.

What we can see from this example is that to construct transitions in a satisfying TPG from a subproblem $(\xi, \langle\alpha\rangle_{\sqsupseteq p}F)$ of a simple problem Π, we have to consider all the disjoint ways of satisfying formulas succeeding $\langle\alpha\rangle_{\sqsupseteq p'}$- and $\neg\langle\alpha\rangle_{\sqsupseteq p'}$-modalities. In the example we have two subproblems each giving rise to the construction of edges in a satisfying TPG. These two problems are $(\xi, \langle a\rangle_{\geq\frac{1}{2}}F_1)$ and $(\xi, \langle a\rangle_{\geq\frac{1}{2}}F_2)$.

Consider the first problem $(\xi, \langle a\rangle_{\geq\frac{1}{2}}F_1)$. When trying to construct an a-edge we have to consider all the disjoint ways of satisfying conjunctions of F_1 and formulas succeding a $\neg\langle a\rangle_{\sqsupseteq p}$-modality. That is we consider all possible and disjoint ways of conjuncting the above formulas and their negations. In this case we have two subproblems containing a $\neg\langle\alpha\rangle_{\sqsupseteq p}$-modality. That is, we must consider all possible ways of conjuncting the formulas in the multiset $\{\!|\ F_1, F_1, F_2\ |\!\}$ and their negations. There are 8 ways of combining these formulas in the way described above, but only 4 of them are consistent, or at least not obviously inconsistent. In the other 4 combinations the inconsistent conjunction $F_1 \wedge \neg F_1$ is part of the combination. The combinations not obviously inconsistent are: $(F_1 \wedge F_1 \wedge F_2)$, $(F_1 \wedge F_1 \wedge \neg F_2)$, $(\neg F_1 \wedge \neg F_1 \wedge F_2)$ and $(\neg F_1 \wedge \neg F_1 \wedge \neg F_2)$. Removing the redundant F_1 or $\neg F_1$ gives exactly the 4 ways of combination as described first in the example.

Obviously, the same 8 combinations as above will arise when we consider the construction of edges initiating from the subproblem $(\xi, \langle a\rangle_{\geq\frac{1}{2}}F_2)$ and thus we would have been satisfied with just making the 8 combinations originating from the first subproblem considered. Though, this is a special case due to the fact that both subproblems $(\xi, \langle a\rangle_{\geq\frac{1}{2}}F_1)$ and $(\xi, \langle a\rangle_{\geq\frac{1}{2}}F_2)$ have the same action modality and the same region part. In general we need to consider the combinations of all formulas succeding the same $\langle\alpha\rangle_{\sqsupseteq p}$-modalities for subproblems in the *same* region, and formulas succeding the same $\neg\langle\alpha\rangle_{\sqsupseteq p}$-modality in *any* region.

Having introduced the idea of disjoint formulas we can now in general describe the construction strategy as follows. Whenever $(\xi, \langle\alpha\rangle_{\sqsupseteq p}F)$ is a subproblem of some simple problem Π we consider the set of disjoint formulas as described above. To obtain decidability there must exist transitions to a set of decidable subproblems spanning a subset of the disjoint formulas. That is there must exist reset sets, enabling conditions and weights s.t. the above transitions exist. Furthermore, the normalized sum of weights for transitions leading to subproblems including F must be $\sqsupseteq p$. Likewise, for each subproblem $(\xi', \neg\langle\alpha\rangle_{\sqsupseteq p''}F') \in \Pi$ it must be the case that the normalized sum of weights for transitions leading to subproblems including F' must be $\sqsubseteq p''$ where $\sqsubseteq$ denotes the dual of $\sqsupseteq$.

Formally the construction approach is described as follows. First we define the set of formulas taking part in the disjoint joining.

Definition 5.6 *Let Π be a simple problem. We define the modal projection of Π by:*

$$\begin{aligned} \mathcal{G}^{\xi,\alpha}_{\Pi} &= \{\!|\ F \mid (\xi, \langle\alpha\rangle_{\sqsupseteq p} F) \in \Pi\ |\!\} \cup \\ & \quad \textstyle\bigcup_{\xi'} \{\!|\ F \mid (\xi', \neg\langle\alpha\rangle_{\sqsupseteq p} F) \in \Pi\ |\!\} \end{aligned}$$

Now, given a set $\mathcal{G}^{\xi,\alpha}_{\Pi}$ the set of disjoint formulas are defined as follows.

Definition 5.7 *Let $\mathcal{G}^{\xi,\alpha}_{\Pi} = \{\!|\ F_1, \ldots, F_n\ |\!\}$ and let $A \subseteq \{1, \ldots, n\}$. The disjoint formulae of $\mathcal{G}^{\xi,\alpha}_{\Pi}$ are given by:*

$$\mathcal{P}^{\xi,\alpha}_{A,\Pi} = \textstyle\bigwedge_{i \in A} F_i \wedge \bigwedge_{j \notin A} \neg F_j$$

Finally, we can now formally define the model construction theorem implementing the construction strategy descibed informally above.

Theorem 5.1 *Let Π be a simple problem. Furthermore, let C be some TPG clock set, M a maximal constant, $\mathcal{B}_M(C)$ the set of enabling conditions over C and M and finally let m denote a maximal weight on any edge of a TPG. Now, Π is satisfiable if and only if:*

1. *Whenever $(\xi, \langle\alpha\rangle_{\sqsupseteq p} F_k) \in \Pi$ and $F_k \in \mathcal{G}^{\xi,\alpha}_{\Pi} = \{\!|\ F_1, \ldots, F_n\ |\!\}$, there exists a subset $\mathcal{A}^{\xi,\alpha}_{\Pi} \subseteq \{A \mid A \subseteq \{1, \ldots, n\}\}$ s.t. for each $A \in \mathcal{A}^{\xi,\alpha}_{\Pi}$ and $\mathcal{P}^{\xi,\alpha}_{A,\Pi} = \bigwedge_{i \in A} F^{\xi,\alpha}_i \wedge \bigwedge_{j \notin A} \neg F^{\xi,\alpha}_j$ there exists $s^{\xi,\alpha}_{A,\Pi} \in C$, $b^{\xi,\alpha}_{A,\Pi} \in \mathcal{B}_M(C)$ with $b^{\xi,\alpha}_{A,\Pi}(\xi) = tt$ and $w^{\xi,\alpha}_{A,\Pi} \in [0, \ldots, m]$ s.t. $(s^{\xi,\alpha}_{A,\Pi}(\xi), \mathcal{P}^{\xi,\alpha}_{A,\Pi})$ is satisfiable. Furthermore,*

2. *Whenever $(\xi, \langle\alpha\rangle_{\sqsupseteq p} F_k) \in \Pi$*

   (a)
   $$\frac{\sum\{w^{\xi,\alpha}_{A,\Pi} \mid k \in A \text{ and } A \in \mathcal{A}^{\xi,\alpha}_{\Pi}\}}{\sum_{\xi',\alpha'}\{w^{\xi',\alpha'}_{A,\Pi} \mid b^{\xi',\alpha'}_{A,\Pi}(\xi) = tt \text{ and } A \in \mathcal{A}^{\xi',\alpha'}_{\Pi}\}} \sqsupseteq p$$

   (b) Whenever $(\xi'', \neg\langle\alpha\rangle_{\sqsupseteq p''} F_l) \in \Pi$
   $$\frac{\sum\{w^{\xi,\alpha}_{A,\Pi} \mid l \in A,\ A \in \mathcal{A}^{\xi,\alpha}_{\Pi} \text{ and } b^{\xi,\alpha}_{A,\Pi}(\xi'') = tt\}}{\sum_{\xi',\alpha'}\{w^{\xi',\alpha'}_{A,\Pi} \mid b^{\xi',\alpha'}_{A,\Pi}(\xi'') = tt \text{ and } A \in \mathcal{A}^{\xi',\alpha'}_{\Pi}\}} \sqsubseteq p''$$

   where $\sqsubseteq$ denotes the dual of $\sqsupseteq$ and $\mathcal{A}^{\xi',\alpha'}_{\Pi}$ are found as in 1.

3. *Whenever $(\xi, c_1 + x \sim c_2 + y) \in \Pi$ then $\xi(c_1) + x \sim \xi(c_2) + y$*

PROOF:

Only if-direction. Let Π be a simple problem and assume that Π is satisfiable. Then there exists a $(C, M){-}TPG$, G, with a node n s.t. whenever $(\xi, \langle\alpha\rangle_{\sqsupseteq p} F) \in \Pi$ then $\vdash \langle\!\langle n, \xi\rangle\!\rangle^+ : \langle\alpha\rangle_{\sqsupseteq p} F$. From the definition of $\vdash$ we now know that there exists an S s.t. $\{\vdash \langle\!\langle n', \xi' \rangle\!\rangle^+ : F \mid \langle\!\langle n', \xi' \rangle\!\rangle^+ \in S\}$ and $\pi^+(\langle\!\langle n, \xi \rangle\!\rangle^+, \alpha, S) \sqsupseteq p$. Therefore, there exists a multiset of satisfiable problems $P = \{\!\mid (\xi', F) \mid (\vdash \langle\!\langle n', \xi' \rangle\!\rangle^+ : F) \in \{\langle\!\langle n', \xi' \rangle\!\rangle^+ : F \mid \langle\!\langle n', \xi' \rangle\!\rangle^+ \in S\} \mid\!\}$. Now, if $\mathcal{G}_{\Pi}^{\xi,\alpha} = \{F_1, \dots, F_n\}$ then $\bigcup_{A \subseteq \{1,\dots,n\}} \mathcal{P}_{A,\Pi}^{\xi,\alpha}$ spans all disjoint ways of satisfying formulas following an $\langle\ \rangle$-modality, and indeed it spans all disjoint ways of satisfying F. Now, it is obvious that each of the problems in P exactly matches one of the disjoint ways of satisfying F, and therefore it is possible to choose a subset $\mathcal{A}_{\Pi}^{\xi,\alpha} \subseteq \{A \mid A \subseteq \{1, \dots, n\}\}$ s.t. for each $A \in \mathcal{A}_{\Pi}^{\xi,\alpha}$ there exist $s_{A,\Pi}^{\xi,\alpha} \in C$, $b_{A,\Pi}^{\xi,\alpha} \in \mathcal{B}_M(C)$ with $b_{A,\Pi}^{\xi,\alpha}(\xi) = tt$ and $w_{A,\Pi}^{\xi,\alpha} \in [0, \dots, m]$ s.t. $(s_{A,\Pi}^{\xi,\alpha}(\xi), \mathcal{P}_{A,\Pi}^{\xi,\alpha})$ is satisfiable and s.t. $2(a)$ holds. $2(b)$ holds using s similar argument as above, and 3 holds obviously.

If-direction. For $(\xi, \langle\alpha\rangle_{\sqsupseteq p} F) \in \Pi$, $\mathcal{G}_{\Pi}^{\xi,\alpha} = \{\!\mid F_1, \dots, F_n \mid\!\}$ and a subset $\mathcal{A}_{\Pi}^{\xi,\alpha} \subseteq \{A \mid A \subseteq \{1, \dots, n\}\}$, let for each $A \in \mathcal{A}_{\Pi}^{\xi,\alpha}$ and $s_{A,\Pi}^{\xi,\alpha} \in C$, $b_{A,\Pi}^{\xi,\alpha}(\xi) = tt$ and $w_{A,\Pi}^{\xi,\alpha}$, $G_{A,\Pi}^{\xi,\alpha}$ denote the TPG with start node $n_{A,\Pi}^{\xi,\alpha}$ s.t. it satisfies $(s_{A,\Pi}^{\xi,\alpha}, \mathcal{P}_{A,\Pi}^{\xi,\alpha})$ and 2 and 3. Then the TPG, G with startnode n_0 and with edgeset from n_0 being:

$$\bigcup_{\xi,\alpha}\{\bigcup\{n_0 \overset{\alpha, w_{A,\Pi}^{\xi,\alpha}, b_{A,\Pi}^{\xi,\alpha}, s_{A,\Pi}^{\xi,\alpha}}{\longrightarrow} n_{A,\Pi}^{\xi,\alpha}\}\}$$

satisfies Π. □

Theorem 5.2 *Bounded satisfiability of problems is decidable.*

Proof: From the properties of $\rightarrow_{red}$ and from the fact that the possible choices for reset sets s over C, $\mathcal{B}(C)$ and weights $w \in [1, \dots, m]$ are all finite, it follows that the bounded satisfiability is decidable. □

Example 5.3 We want to decide whether there exists a $(1,1)$-TPG solving the problem:

$$\Pi = \{(\xi_0, c \text{ in } (\exists(c = 0 \land \langle a \rangle_{\geq \frac{1}{2}} F_1)) \land c \text{ in } (\exists(0 < c < 1 \land \neg\langle a \rangle_{>0} F_2)))\}$$

where we have the TPG-clock is denoted x and where we have the following regions $\xi_0 = \{x = 0, c = 0\}$ and $\xi_1 = \{0 < x < 1, 0 < c < 1, x = c\}$.

Using the reduction $\rightarrow_{red}$ we obtain the following:

$$\begin{array}{lll} \Pi & \rightarrow_{red} & \{(\xi_0, \exists(c = 0 \land \langle a \rangle_{\geq \frac{1}{2}} F_1)), (\xi_0, \exists(0 < c < 1 \land \neg\langle a \rangle_{>0} F_2))\} \\ & \rightarrow_{red}^{*} & \{(\xi_0, c = 0), (\xi_0, \langle a \rangle_{\geq \frac{1}{2}} F_1), (\xi_1, 0 < c < 1), (\xi_1, \neg\langle a \rangle_{>0} F_2)\} \end{array}$$

Now, the two problems $(\xi_0, c = 0)$ and $(\xi_1, 0 < c < 1)$ are obviously satisfiable due to 3. According to 1 we now have to consider the problem $(\xi_0, \langle a \rangle_{\frac{1}{2}} F_1)$. Following 1 we get that $\mathcal{G}^{\xi_0,a} = \{F_1, F_2\}$. We now have to choose a subset of the set of disjoint formulas obtained from conjuncting F_1, F_2 and their negations. We choose $\mathcal{A}^{\xi_0,a} = \{\emptyset, \{1\}\}$ and we get $\mathcal{P}_{\emptyset}^{\xi_0,a} = \neg F_1 \land \neg F_2$ and $\mathcal{P}_{\{1\}}^{\xi_0,a} = F_1 \land \neg F_2$.

For the sake of simplicity we choose the same reset sets $s^{\xi_0,a}_{\emptyset} = s^{\xi_0,a}_{\{1\}} = \emptyset$ and the same enabling conditions $b^{\xi_0,a}_{\emptyset} = b^{\xi_0,a}_{\{1\}} = (x \geq 0)$. From $2(a)$ we get the following requirements on the weights:

$$\frac{w^{\xi_0,a}_{\{1\}}}{w^{\xi_0,a}_{\emptyset} + w^{\xi_0,a}_{\{1\}}} \geq \frac{1}{2}$$

We choose the following weights satisfying the above inequality: $w^{\xi_0,a}_{\emptyset} = w^{\xi_0,a}_{\{1\}} = 1$. □

Though the problem of bounded satisfiability has been shown decidable, the complexity of the algorithm for model construction defined by the iterative steps of reduction and construction is obviously very high. If we just take a look at the construction steps of the algorithm, defined by Theorem 5.1 we will see that each of the construction steps in the iterative algorithm is exponential in the size of the simple problem initiating the construction.

Given a simple problem Π and given some subproblem $(\xi, \langle\alpha\rangle_{\sqsupseteq p}F) \in \Pi$ then we get the following complexity for constructing the edges in a resulting TPG originating from this subproblem.

First, if we let $n = \mid \mathcal{G}^{\xi,\alpha}_{\Pi} \mid$ denote the number of formulas in subproblems of Π that succeeds a $\langle\alpha\rangle$-modality. Now in the worst case situation we will have to examine all possible ways of satisfying some subset of disjoint formulas. That is, worst case we have to examine all possible subsets $\mathcal{A}^{\xi,\alpha}_{\Pi} \subseteq \{A \mid A \subseteq \{1, \ldots, n\}\}$. Already here we get the exponential complexity, as this number of possible subsets is 2^n. As n is bounded by the size of Π ($\mid \Pi \mid$) i.e. the number of subproblems in Π, the complexity for the construction step is already $O(2^{|\Pi|})$.

For each of the above subsets, having $\mid \mathcal{A}^{\xi,\alpha}_{\Pi} \mid$ elements, we are now confronted with the problem of finding enabling conditions, reset sets and weights to label the edges of the resulting TPG.

Enabling conditions are of the form $c_1 + x \sim c_2 + y$ where $\sim \in \{<, \leq, =, \geq, >\}$. If we let K denote the maximal value of x, y and if $\mid C \mid$ denotes the number of clocks, then the number of possible enabling conditions will be $\mid C \mid^2 \cdot K^2 \cdot 5$ i.e. choosing enabling conditions is $O(\mid C \mid^2)$. Furthermore, we can choose among $2^{|C|}$ different reset sets, and the number of possible weights to label edges is given beforehand. We let W denote this number.

Now, the final complexity of one construction step is as follows:

$$O(2^{|\Pi|} \cdot 2^{|C|} \cdot \mid \mathcal{A}^{\xi,\alpha}_{\Pi} \mid \cdot \mid C \mid^2 \cdot W)$$

To conclude, we can see that the complexity of one construction step is exponential in the size of the simple problem at hand and in the number of clocks in the TPG.

Chapter 6

Characteristic Formulas

This chapter describes how to construct characteristic formulas for timed probabilistic processes described by TPG's.

Before we describe the construction of characteristic formulas for timed probabilistic processes, we present some notational conventions used in this chapter. Afterwards we construct the characteristic formulas for regular timed probabilistic processes, that is, processes described by acyclic TPG's.

In the rest of this chapter the following abbreviations will be used.

C in F means c_1 in (c_2 in $(\ldots(c_n$ in $F)\ldots))$ for a clock set C and a formula F.

Given a TPG, $\langle N, n_0, C, \longrightarrow \rangle$, let e range over $\longrightarrow$ and for $e = \langle n, a, w, b, C', n' \rangle$ we let n_e (resp. a_e, w_e, b_e, C'_e, n'_e) denote n (resp. a, w, b, C', n').

For a region ξ we define $E(n,\xi) = \{e \in \longrightarrow \mid n_e = n \wedge b_e(\xi)\}$ that is, the set of edges from n enablet in the region ξ and similarly we define $E(\eta, \xi, a) = \{e \in \longrightarrow \mid n_e = n \wedge a_e = a \wedge b_e(\xi)\}$ to be the set of edges from n labelled with an a-action and enabled in the region ξ. For a region ξ, $C(\xi)$ means $\xi[C \leftarrow 0]$. For any ξ there exists a RTPL formula describing the region. This formula is a conjunction of atomic formulas of the type $c_1 + x \sim c_2 + y$. Let $\beta(\xi)$ denote the RTPL formula describing ξ.

6.1 Construction of characteristic formulas

In this section we construct the characteristic formulas for timed probabilistic processes described by acyclic TPG's. Due to the lack of recursion in RTPL we require that any edge in the TPG has at least *one* clock in the reset set, since we otherwise could be constructing formulas with mutual recursion, which we want to avoid. For the same reason we require that the TPG is *acyclic.*

Given an acyclic TPG G, what is the characteristic formula F^G for G ? The formula we want to construct is the formula for the configuration $\langle n, \gamma \rangle$ given an arbitrary time assignment γ and an arbitrary node n of G. That is, a formula for each state in the configuration graph for G. Since the configuration graph has infinite branching we cannot efficiently construct the characteristic formula for timed probabilistic processes taking the configuration graph as our basis. But we know that given two time assignments $\gamma, \gamma' \in \mathbf{R}^C$ where $\gamma \doteq \gamma'$, we have for all $\mu \in \mathbf{R}^K$ for some formula clock set K, and for some node n in G, that $\langle n, \gamma\mu\rangle^+$ and $\langle n, \gamma'\mu\rangle^+$ will satisfy the same formulas. Therefore it should be enough to construct the characteristic formula for node n associated with the regions over the clocks of G that is, the symbolic states $\langle\!\langle n, [\gamma]\rangle\!\rangle$. This causes the characteristic formula for a TPG to be the characteristic formula for the initial node n_0 associated with the initial region ξ_0 and we denote this formula F_{n_0,ξ_0}. Now, since we only consider closed formulas, F^G should be $F^G = C^F \text{in } F_{n_0,\xi_0}$ where C^F is a set of formula clocks such that for each $c \in C$ there exists a $c^F \in C^F$. In the sequel we will use the clocks in the clock set of the TPG, C to describe the clocks in the clock set C^F.

Considering a TPG in some symbolic state $\langle\!\langle n, \xi\rangle\!\rangle$, then $E(n, \xi)$ denotes the set of enabled edges for n in the region ξ. Given this set of edges, we want to characterize for any action in the action set of the TPG, all the different $\langle\rangle$-formulas this action give rise to. That is, for any edge in the set $E(n, \xi, a)$ where we have that $E(n, \xi, a) \neq \emptyset$ we characterize the action by the formula

$$\bigwedge_{I \subseteq E(n,\xi,a)} \langle a\rangle_{\geq p_{I,\xi}} \left(\bigvee_{e \in I} C'_e \text{ in } F_{n'_e, C'_e(\xi)} \right)$$

where

$$p_{I,\xi} = \frac{\sum_{e \in I} w_e}{\sum_{e \in E(n,\xi)} w_e}$$

The reason why we require $E(n, \xi, a) \neq \emptyset$ is that

$$\langle a\rangle_{\geq p_{\emptyset,\xi}} \left(\bigvee_{\emptyset} C'_e \text{ in } F_{n'_e, C'_e(\xi)} \right) = tt$$

This implies that the characterization of the action behaviour also shall contain a conjunct $\bigwedge_{a.E(n,\xi,a)=\emptyset} [a]ff$. The characterisation of the action behaviour of the node $\langle\!\langle n, \xi\rangle\!\rangle$ must now be

$$F^1_{n,\xi} = \bigwedge_{a \in Act} \bigwedge_{I \subseteq E(n,\xi,a)} \langle a\rangle_{\geq p_{I,\xi}} \left(\bigvee_{e \in I} C'_e \text{ in } F_{n'_e, C'_e(\xi)} \right) \wedge \bigwedge_{a.E(n,\xi,a)=\emptyset} [a]ff$$

where $p_{I,\xi}$ is defined as above.

Having described the action behaviour, we now consider the description of the delay behaviour of a symbolic state $\langle\!\langle n, \xi\rangle\!\rangle$. The formula describing the delay behaviour should express that any delay performed by the symbolic state will lead to a new state where the characteristic formula for this new state is satisfied. By requiring that at least one clock is reset when an action is performed we ensure that the region we reach by any action transition is a boundary-region.

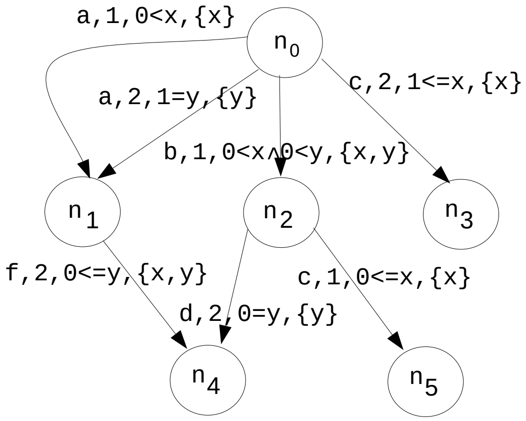
Also, *any* positive delay will this way lead to a new region. Since we have to characterize *any* delay possible the formula characterizing time transitions must be a $\forall\!\!\!\forall$-formula. The formula characterizing the delay behaviour is as follows:

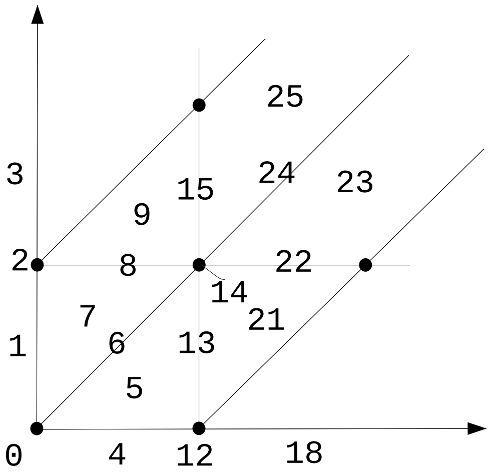
$$F^2_{n,\xi} = \forall\!\!\!\forall \left(\bigwedge_{i=1}^{l_\xi} \beta(\xi^i) \Rightarrow F_{n,\xi^i} \right)$$

If we combine the formulas describing the action behaviour and the delay behaviour for $\langle\!\langle n, \xi \rangle\!\rangle$ we get the following characteristic formula for this symbolic state.

$$\begin{aligned} F_{n,\xi} \overset{\text{def}}{=} \quad & \bigwedge_{a \in Act} \bigwedge_{I \subseteq E(n,\xi,a)} \langle a \rangle_{\geq p_{I,\xi}} \left(\bigvee_{e \in I} C'_e \text{ in } F_{n,C'_e(\xi)} \right) \wedge \\ & \bigwedge_{a.E(n,\xi,a)=\emptyset} [a] ff \wedge \forall\!\!\!\forall \left(\bigwedge_{i=1}^{l_\xi} \beta(\xi^i) \Rightarrow F_{n,\xi^i} \right) \end{aligned}$$

Example 6.1 Given a TPG G, as below to the left we construct the characteristic formula for this TPG. The actionset of G is $Act_{lazy} \cup Act_{urg}$ where $Act_{lazy} = \{a, b, c, f\}$ and $Act_{urg} = \{d\}$. We will in this example use *nil* as abbreviation for $\bigwedge_{a.E(n,\xi,a)=\emptyset} [a] ff$.





The characteristic formula for G will then be as follows:

$$
\begin{aligned}
F_{n_0,\xi_0} = {} & nil \wedge \forall\!\!\!\forall \left(\bigwedge_{i\in\{6,14,24\}} \beta(\xi_i) \Rightarrow F_{n_0,\xi_i} \right) \\
F_{n_0,\xi_6} = {} & \langle a\rangle_{\geq\frac{1}{2}}(x \text{ in } F_{n_1,\xi_1}) \wedge \langle b\rangle_{\geq\frac{1}{2}}(x \text{ in } (y \text{ in } F_{n_2,\xi_0})) \wedge [c]ff \wedge [d]ff \wedge [f]ff \wedge \\
& \forall\!\!\!\forall \left(\bigwedge_{i\in\{14,24\}} \beta(\xi_i) \Rightarrow F_{n,\xi_i} \right) \\
F_{n_0,\xi_{14}} = {} & \langle a\rangle_{\geq\frac{1}{6}}(x \text{ in } F_{n_1,\xi_2}) \wedge \langle a\rangle_{\geq\frac{1}{3}}(y \text{ in } F_{n_1,\xi_{12}}) \wedge \langle a\rangle_{\geq\frac{1}{2}}((x \text{ in } F_{n_1,\xi_2}) \vee (y \text{ in } F_{n_1,\xi_{12}})) \wedge \\
& \langle b\rangle_{\geq\frac{1}{6}}(x \text{ in } (y \text{ in } F_{n_2,\xi_0})) \wedge \langle c\rangle_{\geq\frac{1}{3}}(x \text{ in } F_{n_3,\xi_2}) \wedge [d]ff \wedge [f]ff \wedge \\
& \forall\!\!\!\forall((1 < x \wedge 1 < y \wedge x = y) \Rightarrow F_{n,(\xi_{24})}) \\
F_{n_0,\xi_{24}} = {} & \langle a\rangle_{\geq\frac{1}{4}}(x \text{ in } F_{n_1,\xi_3}) \wedge \langle b\rangle_{\geq\frac{1}{4}}(x \text{ in } (y \text{ in } F_{n_2,\xi_0})) \wedge \langle c\rangle_{\geq\frac{1}{2}}(x \text{ in } F_{n_3,\xi_3}) \wedge [d]ff \wedge [f]ff \\
F_{n_1,\xi_1} = {} & \langle f\rangle_{\geq 1}(x \text{ in } (y \text{ in } F_{n_4,\xi_0})) \wedge [a]ff \wedge [b]ff \wedge [c]ff \wedge [d]ff) \wedge \\
& \forall\!\!\!\forall \left(\bigwedge_{i\in\{7,8,9,15,25\}} \beta(\xi_i) \Rightarrow F_{n_1,\xi_i} \right) \\
F_{n_1,\xi_4} = {} & \langle f\rangle_{\geq 1}(x \text{ in } (y \text{ in } F_{n_4,\xi_0})) \wedge [a]ff \wedge [b]ff \wedge [c]ff \wedge [d]ff) \wedge \\
& \forall\!\!\!\forall \left(\bigwedge_{i\in\{4,5,13,21,22,23\}} \beta(\xi^i) \Rightarrow F_{n_1,\xi_i} \right) \\
F_{n_2,\xi_0} = {} & \langle d\rangle_{\geq\frac{2}{3}} F_{n_4,\xi_0} \wedge \langle c\rangle_{\geq\frac{1}{3}} F_{n_5,\xi_0}) \wedge [a]ff \wedge [b]ff \wedge [f]ff \\
F_{n_4,\xi_0} = {} & nil \wedge \forall\!\!\!\forall \left(\bigwedge_{i\in\{6,14,24\}} \beta(\xi^i) \Rightarrow F_{n_4,\xi_i} \right) \\
F_{n_5,\xi_0} = {} & nil \wedge \forall\!\!\!\forall \left(\bigwedge_{i\in\{6,14,24\}} \beta(\xi^i) \Rightarrow F_{n_5,\xi_i} \right)
\end{aligned}
$$

□

We now prove that $F_{n,\xi}$ is the characteristic formula for the node n in the region ξ.

Theorem 6.1 *Let $G = \langle N, n_0, C, \rightarrow\rangle$ and $H = \langle S, s_0, K, \rightarrow\rangle$ be two acyclic TPG's. Then for any $s \in S$, $n \in N$, $\gamma \in \mathbf{R}^C$ and $\eta \in \mathbf{R}^K$*

$$\langle s, \eta\rangle_H \sim_{tp} \langle n, \gamma\rangle_G \Leftrightarrow \langle s, \eta\gamma\rangle_H^+ \models F_{n,[\gamma]}$$

PROOF:
$\Rightarrow$: Assume $\langle s, \eta\rangle_H \sim_{tp} \langle n, \gamma\rangle_G$. We now have to show that this implies $\langle s, \eta\gamma\rangle_H^+ \models F_{n,[\gamma]}$. By Theorem 3.1 it will be enough to show that $\langle n, \gamma\gamma\rangle_G^+ \models$

$F_{n,[\gamma]}$. Proof by induction in the depth of $\langle\!\langle n,[\gamma]\rangle\!\rangle \in SR[G]$ (finite since $SR[G]$ is acyclic). We then have to show

$$\begin{aligned}\langle n,\gamma\gamma\rangle_G^+ \models\ & \bigwedge_{a\in Act}\ \bigwedge_{I\subseteq E(n,[\gamma],a)} \langle a\rangle_{\geq p_{I,[\gamma]}} \left(\bigvee_{e\in I} C'_e \text{ in } F_{n'_e,C'_e([\gamma])}\right) \wedge \\ & \bigwedge_{a.E(n,[\gamma],a)=\emptyset} [a]ff \wedge \forall\!\!\forall \left(\bigwedge_{i=1}^{l_{[\gamma]}} \beta([\gamma]^i) \Rightarrow F_{n,[\gamma]^i}\right)\end{aligned}$$

If $E(n,[\gamma]) = \emptyset$ it is obvious that $\langle n,\gamma\gamma\rangle_G^+ \models \bigwedge_{a\in Act}[a]ff$ and since no urgent actions are enabled and we have transition liveness there must be time transitions. Now let $S = \{\langle\!\langle n,[\gamma]^i\rangle\!\rangle_G \mid \exists i.\langle\!\langle n,[\gamma]\rangle\!\rangle_G \overset{i}{\twoheadrightarrow} \langle\!\langle n,[\gamma]^i\rangle\!\rangle_G\}$. By IH we then have for all $\langle n,\gamma'\gamma'\rangle_G^+$ where $\langle\!\langle n,[\gamma']\rangle\!\rangle_G \in S$ that
$\langle n,\gamma'\gamma'\rangle_G^+ \models F_{n,[\gamma']}$. This yields

$$\langle n,\gamma\gamma\rangle_G^+ \models \bigwedge_{a\in Act} [a]ff \wedge \forall\!\!\forall \left(\bigwedge_{i=1}^{l_{[\gamma]}} \beta([\gamma]^i) \Rightarrow F_{n,[\gamma]^i}\right)$$

If $E(n,[\gamma]) \neq \emptyset$ then for some action $a \in Act$ we must have $E(n,[\gamma],a) \neq \emptyset$, that is there exists a set of states $S = \{\langle\!\langle n'_e, C'_e([\gamma])\rangle\!\rangle \mid \langle\!\langle n,[\gamma]\rangle\!\rangle \overset{a,p}{\twoheadrightarrow} \langle\!\langle n'_e, C'_e([\gamma])\rangle\!\rangle\}$. From this set we can construct the set $S' = \{\langle n'_e, C'_e(\gamma)\rangle_G \mid \langle\!\langle n'_e, C'_e([\gamma])\rangle\!\rangle \in S\}$. By IH we know for all $\langle n'_e, C'_e(\gamma)\rangle_G \in S'.\langle n'_e, C'_e(\gamma)\gamma\rangle_G^+ \models \bigvee_{e\in I} C'_e$ in $F_{n'_e,C'_e([\gamma])}$ for all $I \subseteq E(n,[\gamma],a)$. Therefore we must have for all $a \in Act.E(n,[\gamma],a) \neq \emptyset$ and where $I \subseteq E(n,[\gamma],a)$ that

$$\begin{aligned}\pi^+(\langle n,\gamma\gamma\rangle^+, a, [\![\textstyle\bigvee_{e\in I} C'_e \text{ in } F_{n'_e,C'_e[\gamma]}]\!]) &\geq \pi^+(\langle n,\gamma\gamma\rangle^+, a, \{\langle n'_e C'_e(\gamma)\gamma\rangle^+ \mid e \in I\}) \\ &\geq \textstyle\sum\{\!|\ p' \mid \langle n,\gamma\gamma\rangle^+ \xrightarrow{a,p'}{}^{+} \langle n'_e, C'_e(\gamma)\gamma\rangle^+ \ \textit{and}\ e \in I\ |\!\}\end{aligned}$$

and for $a \in Act.E(n,[\gamma],a) = \emptyset$ that is $I = \emptyset$ we have

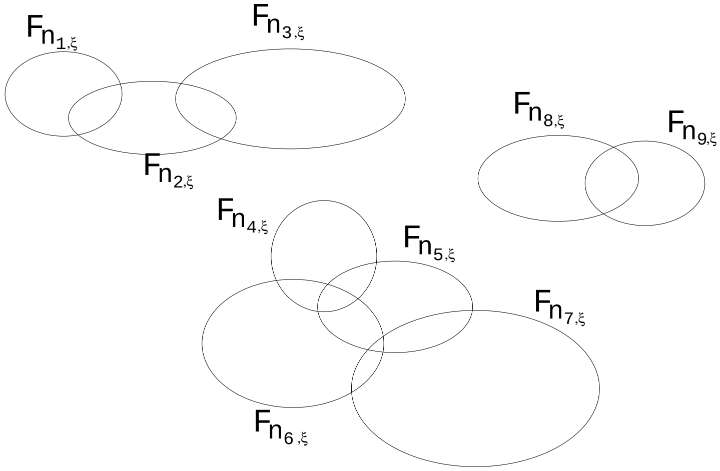
$$\pi^+(\langle n,\gamma\gamma\rangle^+, a, [\![\bigvee_{e\in\emptyset} C'_e \text{ in } F_{n'_e,C'_e[\gamma]}]\!]) = 0$$

If for all $b \in Act_{urg}$ we have that $\pi^+(\langle n,\gamma\gamma\rangle^+, b, \langle n',\gamma'\gamma\rangle^+) = 0$ then we know there must be time transitions since we have transition liveness. Otherwise the conjunction in the $\forall\!\!\forall$ formula will be over the empty set implying that that no time transition can be performed. Now the above give us

$$\begin{aligned}\langle n,\gamma\gamma\rangle_G^+ \models\ & \bigwedge_{a\in Act}\ \bigwedge_{I\subseteq E(n,[\gamma],a)} \langle a\rangle_{\geq p_{I,[\gamma]}} \left(\bigvee_{e\in I} C'_e \text{ in } F_{n,C'_e([\gamma])}\right) \wedge \\ & \bigwedge_{a.E(n,[\gamma],a)=\emptyset} [a]ff \wedge \forall\!\!\forall \left(\bigwedge_{i=1}^{l_{[\gamma]}} \beta([\gamma]^i) \Rightarrow F_{n,[\gamma]^i}\right)\end{aligned}$$

$\Leftarrow$: Let $B = \{(\langle s_1,\eta\rangle_H, \langle s_2,\eta\rangle_H) \mid \langle s_1,\eta\gamma\rangle_H^+, \langle s_2,\eta\gamma\rangle_H^+ \models F_{n,[\gamma]}\}$. B is reflexive and symmetric. Let $\mathcal{B} = \bigcup_{l=0}^{\infty} B^l$ be the transitive closure of B. Clearly $\mathcal{B}$ is an equivalence relation. We now have to show that $\mathcal{B}$ is a timed probabilistic bisimulation.

The classes in $\mathcal{B}$ are unions of sets $[\![C'_e \text{ in } F_{n'_e,C'_e([\gamma])}]\!]$. That is, timed probabilistic processes satisfying C'_e in $F_{n'_e,C'_e([\gamma])}$. This partition of $\mathcal{B}$ is illustated below



To prove that $\mathcal{B}$ is a strong timed probabilistic bisimulation, we have to show that whenever $\langle s_1,\eta\rangle_H\mathcal{B}\langle s_2,\eta\rangle_H$ and S is an equivalence class for $\mathcal{B}$ we have

$$(*)\quad\left\{\begin{array}{lcl}\pi(\langle s_1,\eta\rangle_H,a,S) & = & \pi(\langle s_2,\eta\rangle_H,a,S)\\ \forall t\in\mathbf{R}_{\geq 0}.\langle s_1,\eta\rangle_H \xrightarrow{\epsilon(t)} S & \Leftrightarrow & \langle s_2,\eta\rangle_H \xrightarrow{\epsilon(t)} S\end{array}\right.$$

So let $\langle s_1,\eta\rangle_H\ \mathcal{B}\ \langle s_2,\eta\rangle_H$. Clearly, for some l we have that $\langle s_1,\eta\rangle_H\ B^l\ \langle s_2,\eta\rangle_H$. We proceed by induction in the number of relations, l with the following I.H: $\langle s'_1,\eta'\rangle_H B^l\langle s'_2,\eta'\rangle_H \Rightarrow \langle s'_1,\eta'\rangle_H \sim_{tp} \langle s'_2,\eta'\rangle_H$

$l=0$: then we have to prove that $\langle s_1,\eta\rangle_H \sim_{tp} \langle s_1,\eta\rangle_H$ which should be fairly obvious.

$l=i+1$: Assume $\langle s_1,\eta\rangle_H B^i\langle s',\eta'\rangle_H B\langle s_2,\eta\rangle_H$. By IH, it must enough to show that whenever $\langle s',\eta\rangle_H B\langle s_{\prime\prime},\eta\rangle_H$ and S is an equivalence class for $\mathcal{B}$ then $(*)$ holds.

Therefore let $S_1=\cup_{e\in I_1}[\![C'_e \text{ in } F_{n'_e,C'_e([\gamma])}]\!]$ where $I_1\subseteq E(n,[\gamma],a)$ be an equivalence class of $\mathcal{B}$. Since we know that $\langle s',\eta\rangle_H B\langle s'',\eta\rangle_H$, then by definition of $\models$ it must be the case that

$$\pi^+(\langle s',\eta\gamma\rangle^+_H,a,S_1)\geq\sum_{e\in I_1}\{\!|\ p'\mid\langle n,\gamma\gamma\rangle^+_G \overset{a,p'}{\longrightarrow}{}^+ \langle n'_e,C'_e(\gamma)\gamma\rangle^+_G\ |\!\} \tag{6.1}$$

$$\pi^+(\langle s'',\eta\gamma\rangle^+_H,a,S_1)\geq\sum_{e\in I_1}\{\!|\ p'\mid\langle n,\gamma\gamma\rangle^+_G \overset{a,p'}{\longrightarrow}{}^+ \langle n'_e,C'_e(\gamma)\gamma\rangle^+_G\ |\!\} \tag{6.2}$$

In the same way for any equivalence class at the form $S_j=\cup_{e\in I_j}[\![C' \text{ in } F_{n'_e,C'_e([\gamma])}]\!]$ where $I_j\subseteq E(n,[\gamma],a)$ the above will hold.

Furthermore we know since $\langle s', \eta\gamma\rangle_H^+ \models F_{n,[\gamma]}$ that

$$1 = \sum_{a \in Act} \pi^+(\langle s', \eta\gamma\rangle_H^+, a, \cup_{e \in E(n,[\gamma],a)} [\![C'_e \text{ in } F_{n'_e, C'_e([\gamma])}]\!]) = \sum_{a \in Act} \sum_{j=1}^{k} \pi^+(\langle s', \eta\gamma\rangle_H^+, a, S_j) \tag{6.3}$$

since the S_j's gives us a disjoint partitioning of the equivalence classes. Likewise will (6.3) hold for $\langle s'', \eta\gamma\rangle_H^+$

We then have:

$$1 = \sum_{a \in Act} \sum_{j=1}^{k} (\sum_{e \in I_j} \{\!|\, p' \mid \langle n, \gamma\gamma\rangle_G^+ \xrightarrow{a,p'}{}^+ \langle n'_e, C'_e(\gamma)\gamma\rangle_G^+ \,|\!\}) \tag{6.4}$$

Therefore for all $j \in \{1..k\}$ and for all $a \in Act$ we must have

$$\pi^+(\langle s', \eta\gamma\rangle_H^+, a, S_j) = \sum_{e \in I_j} \{\!|\, p' \mid \langle n, \gamma\gamma\rangle_G^+ \xrightarrow{a,p'}{}^+ \langle n'_e, C'_e(\gamma)\gamma\rangle_G^+ \,|\!\} \tag{6.5}$$

because if we had for just one S_j that

$$\pi^+(\langle s', \eta\gamma\rangle_H^+, a, S_j) > \sum_{e \in I_j} \{\!|\, p' \mid \langle n, \gamma\gamma\rangle_G^+ \xrightarrow{a,p'}{}^+ \langle n'_e, C'_e(\gamma)\gamma\rangle_G^+ \,|\!\}$$

then (6.4) could not hold. (6.4) and (6.5) also hold for $\langle s_{\prime\prime}, \eta\rangle_H^+$ and thus we have:

$$\pi^+(\langle s', \eta\gamma\rangle_H^+, a, S_j) = \pi^+(\langle s_{\prime\prime}, \eta\gamma\rangle_H^+, a, S_j) \tag{6.6}$$

proving the first part of $(*)$.

The proof for the second part of $(*)$ is omitted since it follows the exactly above approach. Hence we now have proven that $\mathcal{B}$ is a timed probabilistic bisimulation.

Now let $\langle s, \eta\gamma\rangle_H \models F_{n,[\gamma]}$. Clearly we have that $(\langle s, \eta\rangle_H, \langle n, \gamma\rangle_G) \in B^l \subseteq \mathcal{B}$ since $\langle n, \gamma\gamma\rangle_G^+ \models F_{n,[\gamma]}$. But this give us that $\langle s, \eta\rangle_H \sim_{tp} \langle n, \gamma\rangle_G$ □

Since we now have proven that we efficiently can construct the characteristic formula for acyclic TPG's, we have an algorithm to determine whether two regular timed probabilistic processes are tp-bisimular. We construct the characteristic formula for the first process and the use the modelchecking algorithm to determine whether the other process satisfies this formula; if so, the two processes are strong timed probabilistic bisimular. This is expressed by the following theorem.

Theorem 6.2 *Strong timed probabilistic bisimulation between acyclic TPG's is decidable*

Proof: Since the TPG only has a finite number of nodes and the number of regions we associate with each node is finite, the construction of the characteristic formula will terminate. The characteristic formula of a acyclic TPG is written in the logic RTPL and the formula is finite. By Theorem 4.3 model checking RTPL formula is deciadable. From the above we get that bisimulation between acyclic TPG's is decidable □

Chapter 7

Conclusion

In this thesis we have presented a formal model of reliable real time systems. We have furthermore presented three different specification languages useful for specifying about the reliable real time systems. To be used in connection with the specification languages two major techniques have been developed. First model checking algorithms for verifying reliable real time systems against specifications of the three specification languages and then an algorithm for automated construction of reliable real time systems.

The formal model of reliable real time systems is in terms of timed probabilistic graphs (TPG's). The semantics of TPG's are timed probabilistic extensions of traditional labelled transition systems.

The specification languages developed are the real timed probabilistic logics RTPL, RTPL_{weak} and RTPL_{until}. In RTPL the notion of formula clocks and delay-modalities gives an elegant way of expressing about the real time domain, and furthermore the delay-modalities are natural generalizations of traditional time-interval modalities. Action-modalities are labelled with probabilities which make it possible to specify about reliable real time systems.

In RTPL_{weak} it is possible to abstract away from probabilistic internal actions when specifying about systems. This possibility is an obvious need when specifying any interesting systems.

Finally, in RTPL_{until} we have extended RTPL with quantification over global time and probabilities, thereby obtaining a timed probabilistic version of the traditional until-operators of CTL. Properties which can be expressed by the until operators are indispensable when specifying about any interesting reliable real time systems.

The model checking problems of the logics RTPL and RTPL_{weak} are proven decidable using the state-region technique to obtain a finite state space of reliable real time processes. We have developed proof systems $\vdash$ and $\vdash_{weak}$ for model checking with respect to the above two logics. Finally, we have proposed a similar proof system for the logic RTPL_{until}.

Also the model construction problem from specifications in RTPL to models in terms of TPG's is shown decidable using the finite state space obtained from the state-region technique.

Finally, we have described how to construct characteristic properties of TPG's in terms of formulae of RTPL.

7.1 Open problems

In this section we will briefly describe some natural extensions of the work already done.

In relation to the work on model checking presented in the thesis, a natural area of further work is the formalization of the model checking ideas for RTPL_{until}. Also, an implementation of the algorithms in an integrated tool would be of obvious use when developing reliable real time systems.

The model construction algorithm presented in the thesis is obviously not an efficient and ready-to-implement algorithm, so concerns about implementing the strategy are very interesting.

When considering characteristic formulas we made some restriction about the TPG's for which we were able to construct the above formulas. Obvious these restrictions would be nice to omit. E.g. it would be interesting to examine the possibility of characterizing cyclic TPG's using the developed until-formulas.

Considering more ambitious goals, the generalization of the theory of reliable real time systems to the context of *hybrid systems* is a very interesting field of study. The TPG's developed in this thesis can be seen as a special case of a model for a hybrid system. A general model for a hybrid system has two phases. One describing discrete transitions of the system, and one modelling a global state continuously changing with time in relation to the laws of physics. In a TPG the only global state variable is a clock, which ticks linear with time and a probability changing as a stepwise function. In hybrid models the global states are not beforehand instantiated to e.g. time or probability. Instead there is a definition of a set of global state variables changing with time with respect to some general time-function. According to the situation at hand it is then possible to instantiate the variables to model different kinds of global states. As an example a global state can in one situation be modelling the temperature of the cooling water in a nuclear plant and in another situation it can be modelling some clocks measuring time, just as in the case of TPG's.

Bibliography

[ACD90] R. Alur, C. Courcoubetis, and D. Dill. Model checking for real-time systems. In *Proceedings of Logic in Computer Science*, pages 117–126, 1990.

[AD90] Rajeev Alur and David Dill. Automata for modelling real-time systems. In *Automata, Languages an Programmming*, pages 332–335. LNCS 443, Springer-Verlag, 1990.

[CES83] E. M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications: A practical approach. In *Proc. 10'th ACM Symp. on Principles of Programming Languages*, pages 117–126, 1983.

[EC82] E. A. Emerson and E.M Clarke. Using branching time temporal logic to synthesize synchronization skeletons. *Sci. Comput. Programming*, pages 2:241–266, 1982.

[EH85] E. A. Emerson and J. Y. Halpern. Decision procedures and expressiveness in the temporal logic of branching time. *J. Comput. System Sci.*, pages 30(1):1–24, 1985.

[EMSS89] A. Emerson, A. Mok, A. Sistla, and J. Srinivasan. Quantitative temporal reasoning. In *Proc. Workshop on Automatic Verification Methods for Finite State Systems, volume 407 of Lecture Notes in Computer Science*. Springer-Verlag, 1989.

[God94] Jens Chr. Godskesen. *Timed Model Specifications - A Theory for Verification of Real-Time Concurrent Systems.* Aalborg University, Department of mathematics and computer science, Ph.D. thesis, R-94-2039, 1994.

[Han91] Hans A. Hansson. *Time and Probability in Formal Design of Distributed Systems.* Uppsala University, Department of Computer Systems, Pd.D. Thesis, 1991.

[HC68] G.E. Hughes and M.J. Cresswell. *An Introduction to Modal Logic.* Methuen and Co., 1968.

[HM85] M. Hennessy and Robin Milner. Algebraic laws for nondeterminism and concurrency. In *Journal of the Association for Computing Machinery, Vol. 32, pp. 137-161*, 1985.

[JG95] Henrik E. Jensen and Heidi Gregersen. *Design af Real Tids Probabilistisk Logik.* Aalborg University, Department of mathematics and computer science, 1995.

[KP83] D. Kozen and R. Parikh. A decision procedure for the propositional mu-calculus. *Lecture Notes in Computer Science*, 1983.

[Lar90] K.G. Larsen. Proof systems for satisfiability in hennessy-milner logic with recursion. *Theoretical Computer Science*, pages 265–288, 1990.

[LLW95] François Laroussinie, Kim G. Larsen, and Carsten Weise. From timed automata to logic - and back. *Brics Report Series, RS-95-2*, 1995.

[LS89] K. G. Larsen and A. Skou. Bisimulation through probabilistic testing: Preliminary report. In *Proceedings 16'th ACM POPL*, 1989.

[Mil89] Robin Milner. *Communication and Concurrency.* Prentice Hall, 1989.

[Par80] D.M.R. Park. Concurrency and automata on infinite sequences. *LNCS 104, Springer-Verlag*, 1980.

[vGSST90] R. van Glabbeek, S.A. Smolka, B. Steffen, and C. Tofts. Reactive, generative and stratified models of probabilistic processes. In *Int Symp. on Logic in Computer Science.* IEEE Computer Society Press, 1990.

[Wol85] P. Wolper. The tableau method for temporal logic: an overview. *Logique et Anal.*, pages 28:119–136, 1985.

[Yi91] Wang Yi. *A Calculus of Real Time System.* Chalmers Tekniska Höjskola. Department of computer science., 1991.