In proceedings of *STACS'97*,
LNCS, Springer-Verlag, February-March 1997.

Temporal Logics for the Specification of Performance and Reliability*

Luca de Alfaro

Department of Computer Science
Stanford University

Abstract. In this paper we present a methodology for the verification of performance and reliability properties of discrete real-time systems. The methodology relies on a temporal logic that can express bounds on the probability of events and on the average time between them. The semantics of the logics is defined with respect to timed systems that exhibit both probabilistic and nondeterministic behavior. We present model-checking algorithms for the algorithmic verification of the specifications, and we discuss their complexity.

1 Introduction

Probabilistic temporal logics have been used for the formal study of correctness and reliability properties of both untimed systems and real-time systems. In this paper, we extend the range of its applications to include performance properties of real-time systems. By introducing an operator that expresses bounds on the average time between events, we propose a unified methodology for the specification and verification of performance, reliability and correctness properties of discrete-time probabilistic systems. The methodology is based on a probabilistic model for the systems, on a specification language derived from temporal logic, and on model-checking algorithms for the verification of system specifications.

We model probabilistic real-time systems as *Markov decision processes* with finite state space [9, 19]. To each state of the Markov decision process is associated a set of actions that can be chosen nondeterministically; the successor of the state is then determined according to the probability distribution arising from the action chosen. Thus, this model can describe both the nondeterministic and the probabilistic components of the system behavior. To each choice of action corresponds a cost, which is interpreted as the amount of time elapsed during the action. In our model, the cost of an action must be either 0 (immediate actions) or 1 (unitary time steps). This system model is closely related to the models proposed in [12, 22, 3].

The specification of system properties is based on the logics pTL and pTL*, and on the use of *instrumentation clocks* to measure the length of intervals of time. The logics pTL and pTL* are obtained by adding the probabilistic

* The research was supported in part by the National Science Foundation under grant CCR-9527927, by the Defense Advanced Research Projects Agency under contract NAG2-892, by ARO under grant DAAH04-95-1-0317, and by ARO under the MURI grant DAAH04-96-1-0341.

operators D and P to the branching-time temporal logics CTL and CTL*. The operator D, introduced in this paper, is used to express bounds on the average time between events. The operator P, already present in the probabilistic logics pCTL and pCTL* [13, 23, 2, 5, 17], is used to express bounds on the probability of system behaviors. The instrumentation clocks, related to the clocks used in timed automata [1], are reset depending on the transitions taken by the system, and their values can be used in the logic to reason about the timing behavior of the system.

To verify whether a system satisfies a specification written in pTL or pTL*, we present model-checking algorithms based on the properties of *stable* subsets of states. A subset of states is stable if there is a choice of nondeterministic actions such that every system behavior that enters the subset will not leave it. The characterization of the properties of stable sets is one of the contributions of this paper, and it leads to uniform algorithms that reduce the model checking problem to the solution of optimization problems on Markov decision processes. The algorithm for the operator P combines the ideas presented in [5] with automata-theoretic constructions, achieving the optimal complexity bound of the earlier algorithm of [6] while exhibiting a relatively simple structure. As discussed in [9, 8, 4], the optimization problems can then be solved by reducing them to linear programming problems. For the logic pTL*, this approach yields model-checking algorithms with time-complexity doubly exponential in the size of the specification, and polynomial in the size of the state space and in the number of bits used to encode the probabilities.

We conclude the paper by discussing an extension of pTL and pTL* that increases the expressive power of the logics by allowing the operator D to refer to arbitrary past formulas.

2 Timed Probabilistic Nondeterministic Systems

A *timed probabilistic nondeterministic system* (TPNS) is a Markov decision process in which the cost of the actions is either 0 or 1 [9].

Definition1 (TPNS). A TPNS $\Pi = (\mathcal{P}, S, Acts, \kappa, p, c, s_{in})$ consists of the following components.

1. A set $\mathcal{P}$ of propositional symbols.
2. A finite state space S. Every state $s \in S$ assigns truth value $s[\![x]\!]$ to every symbol $x \in \mathcal{P}$.
3. A finite set of actions $Acts$.
4. A function κ, which associates with each $s \in S$ the non-empty set $\kappa(s) \subseteq Acts$ of actions that can be taken at state s.
5. A probability distribution p, that for all $s, t \in S$ and $a \in \kappa(s)$ gives the probability $p(t \mid s, a)$ of a transition from s to t under action a. For all $s \in S$ and $a \in \kappa(s)$, we require $\sum_{t \in S} p(t \mid s, a) = 1$.
6. a function c that associates with each $s \in S$ and $a \in \kappa(s)$ the cost (equal to the elapsed time) $c(s, a) \in \{0, 1\}$ of performing a at s.
7. an initial state $s_{in} \in S$. □

Given a state $s \in S$, the successor of s is chosen in two steps: first, an action $a \in \kappa(s)$ is selected nondeterministically; second, a successor state $t \in S$ is chosen according to the probability distribution $p(t \mid s, a)$. This process, iterated, gives rise to the *behaviors* of a TPNS.

Definition 2 (behaviors of TPNS). A *behavior* of a TPNS Π is an infinite sequence $\omega : s_0 a_0 s_1 a_1 \cdots$ such that $s_i \in S$, $a_i \in \kappa(s_i)$ and $p(s_{i+1} \mid s_i, a_i) > 0$ for all $i \geq 0$. Given a behavior $\omega : s_0 a_0 s_1 a_1 \cdots$, we denote by ω_i the state s_i, by ω_i^a the action a_i, and by $\omega_{\geq i}$ the behavior $s_i a_i s_{i+1} a_{i+1} \cdots$. □

Policies and probability of behaviors. For every state $s \in S$, we denote by Ω_s the set of behaviors starting from s, and we let $\mathcal{B}_s \subseteq 2^{\Omega_s}$ be the σ-algebra of *measurable* subsets of Ω_s, following the classical definition of [14]. To be able to talk about the probability of system behaviors, we would like to associate to each $\Delta \in \mathcal{B}_s$ its probability measure $\mu(\Delta)$. However, this measure is not well-defined, since the probability that a behavior $\omega \in \mathcal{B}_s$ belongs to Δ may depend on the criterion by which the actions are chosen.

To represent these choice criteria, we use the concept of *policy* [9, 19]. A policy η is a set of conditional probabilities $Q_\eta(a \mid s_0 a_0 s_1 \cdots s_n)$, where $a \in \kappa(s_n)$. A policy dictates the probabilities with which the actions are chosen: according to policy η, after the finite prefix $s_0 a_0 s_1 \cdots s_n$ starting at the root $s = s_0$ of Ω_s, action $a \in \kappa(s_n)$ is chosen with probability $Q_\eta(a \mid s_0 a_0 s_1 \cdots s_n)$. Thus, the probability of a direct transition to $t \in S$ after $s_0 \cdots s_n$ is given by

$$\Pr_s^\eta(t \mid s_0 a_0 s_1 \cdots s_n) = \sum_{a \in \kappa(s_n)} p(t \mid s_n, a)\, Q_\eta(a \mid s_0 a_0 s_1 \cdots s_n) \,.$$

These transition probabilities give rise to a unique probability measure μ_s^η on $\mathcal{B}_s$. We write $\Pr_s^\eta(A)$ to denote the probability of event A in Ω_s under policy η and probability measure μ_s^η, and we adopt the usual conventions to denote conditional probabilities and expectations.

Non-Zeno systems. We say that a TPNS $\Pi = (\mathcal{P}, S, Acts, \kappa, p, c, s_{in})$ is *non-Zeno* if a behavior from s_{in} follows with probability 1 infinitely many time steps, under any policy. Formally, we require that for any policy η, $\Pr_{s_{in}}^\eta\left(\sum_{i=0}^\infty c(\omega_1, \omega_i^a) = \infty\right) = 1$. In general, we are only interested in non-Zeno systems, and the proof of the above property should precede the proof of any other system specification. We will present algorithms to check whether a TPNS is non-Zeno.

Modeling a real-time system with TPNS. While TPNS provide a general model for real-time probabilistic systems, they model systems at a fairly low level. The report [7] introduces *stochastic real-time systems* (SRTS), which provide a more usable modeling language, and it describes how to translate SRTS into TPNS. Translation of stochastic process calculi into models related to TPNS have also been presented in [12]. Since the main focus of this paper are the specification language and the model-checking algorithms, we have omitted the description of these higher-level languages.

3 Specification Language: pTL and pTL*

The specification of performance and reliability properties of TPNS is based on the use of *instrumentation clocks* to measure the length of intervals of time, and on the probabilistic temporal logics pTL and pTL*, that extend pCTL and pCTL* by introducing an operator D to express bounds on the average time between events [13, 2, 5].

3.1 Instrumentation Clocks

An *instrumentation clock* ξ is defined by a propositional formula ξ_τ over $\mathcal{P} \cup \mathcal{P}'$, where $\mathcal{P}' = \{x' \mid x \in \mathcal{P}\}$. The formula ξ_τ specifies a transition relation $\{(s,s') \in S^2 \mid (s,s') \models \xi_\tau\}$, where (s,s') interprets $x \in \mathcal{P}$ as $s[\![x]\!]$ and $x' \in \mathcal{P}'$ as $s'[\![x]\!]$. When a transition from s to s' under action $a \in \kappa(s)$ occurs, clock ξ is incremented by $c(s,a)$ if $(s,s') \not\models \xi_\tau$, and is reset if $(s,s') \models \xi_\tau$. Thus, clock ξ measures the time elapsed since the last state transition that satisfies ξ_τ.

In previous approaches, the specification of timing properties of probabilistic systems relied on temporal operators augmented by time bounds [13, 12, 3]. The instrumentation clocks, derived from the clocks used in timed automata [1], and clocked transition systems [15], lead to a simpler definition of the logic and to a more compact presentation of the model-checking algorithms.

3.2 Syntax of pTL and pTL*

We distinguish two classes of pTL and pTL* formulas: the class *Stat* of *state formulas* (whose truth value is evaluated on the states), and the class *Seq* of *sequence formulas* (whose truth value is evaluated on infinite sequences of states). Given a set $\mathcal{P}$ of predicate symbols and a set C of instrumentation clocks, the classes *Stat* and *Seq* for pTL* are defined inductively as follows.

State formulas:

$$p \in \mathcal{P} \implies p \in \mathit{Stat} \qquad \xi \in C \implies \xi > k, \xi = k \in \mathit{Stat} \quad (1)$$

$$\phi, \psi \in \mathit{Stat} \implies \phi \wedge \psi, \neg\phi \in \mathit{Stat} \qquad \phi \in \mathit{Seq} \implies \mathrm{A}\phi, \mathrm{E}\phi \in \mathit{Stat} \quad (2)$$

$$\phi \in \mathit{Seq} \implies \mathrm{P}_{\bowtie b}\phi \in \mathit{Stat} \qquad \phi \in \mathit{Stat} \implies \mathrm{D}_{\bowtie d}\phi \in \mathit{Stat}\,. \quad (3)$$

Sequence formulas:

$$\phi \in \mathit{Stat} \implies \phi \in \mathit{Seq} \qquad \phi,\ \psi \in \mathit{Seq} \implies \phi \wedge \psi, \neg\phi \in \mathit{Seq} \quad (4)$$

$$\phi \in \mathit{Seq} \implies \Box\phi,\ \Diamond\phi \in \mathit{Seq} \qquad \phi,\ \psi \in \mathit{Seq} \implies \phi\,\mathcal{U}\,\psi \in \mathit{Seq}\,. \quad (5)$$

In the above definition, $\bowtie$ stands for one of $\{<, \leq, \geq, >\}$, $k \in \mathbb{N}$, $b \in [0,1]$ and $d \geq 0$. The temporal operators $\Box$, $\Diamond$, $\mathcal{U}$, and the path quantifiers A, E are taken from CTL* [10], the probabilistic operator P is taken from pCTL* [2, 5], and the operator D originates here. As usual, the other propositional connectives are defined in terms of $\neg$, $\wedge$. The logic pTL is a restricted version of pTL*; its definition is obtained by replacing the clauses (4), (5) with the single clause

$$\phi,\ \psi \in \mathit{Stat} \implies \Box\phi,\ \Diamond\phi,\ \phi\,\mathcal{U}\,\psi \in \mathit{Seq}\,. \quad (6)$$

3.3 Semantics

Given a TPNS Π and a set C of instrumentation clocks, the semantics of a formula ϕ of pTL or pTL* with respect to Π and C is defined in two steps. First, we construct an *instrumented* TPNS Π_C^ϕ, whose states keep track of the value of the clocks; second, we define the satisfaction of ϕ on Π_C^ϕ.

Instrumenting the TPNS. Given a TPNS $\Pi = (\mathcal{P}, S, \mathit{Acts}, \kappa, p, c, s_{in})$ and a set $C = \{\xi_1, \ldots, \xi_n\}$ of instrumentation clocks over S, we construct an *instrumented* TPNS Π_C^ϕ, whose states keep track of the value of the clocks. Note that if M_ξ is the largest constant with which the clock $\xi \in C$ is compared to in ϕ, we need to keep track of the value of ξ only up to $M_\xi + 1$, since no inequality of ϕ changes truth value when the value of ξ increases beyond $M_\xi + 1$. In light of this observation, we define the TPNS $\Pi_C^\phi = (\mathcal{P}, S^*, \mathit{Acts}, \kappa^*, p^*, c^*, s^*_{in})$ as follows.

1. $S^* = S \times \prod_{\xi \in C} \{0, \ldots, M_\xi + 1\}$. A state $(s, c_1, \ldots, c_n) \in S^*$ assigns value $(s, c_1, \ldots, c_n)[\![x]\!] = s[\![x]\!]$ to $x \in \mathcal{P}$ and $(s, c_1, \ldots, c_n)[\![\xi_i]\!] = c_i$ to $\xi_i \in C$.
2. For $(s, c_1, \ldots, c_n) \in S^*$, $\kappa^*(s, c_1, \ldots, c_n) = \kappa(s)$.
3. For $(s, c_1, \ldots, c_n), (s', c'_1, \ldots, c'_n) \in S^*$ and $a \in \kappa^*(s, c_1, \ldots, c_n)$, $p^*\big((s', c'_1, \ldots, c'_n) \bigm| (s, c_1, \ldots, c_n), a\big)$ is equal to $p(s' \mid s, a)$ if for all $\xi_i \in C$:
$$c'_i = \begin{cases} 0 & \text{if } (s, s') \models \xi_{i\tau}; \\ \max\{c(s,a) + c_i, M_{\xi_i} + 1\} & \text{otherwise;} \end{cases}$$
and is equal to 0 otherwise.
4. For $(s, c_1, \ldots, c_n) \in S^*$ and $a \in \kappa(s)$, $c^*\big((s, c_1, \ldots, c_n), a\big) = c(s, a)$.
5. $s^*_{in} = (s_{in}, 0, \ldots, 0)$.

Semantics over the instrumented TPNS. The truth value of pTL and pTL* formulas is then defined with respect to the instrumented TPNS $\Pi_C^\phi = (\mathcal{P}^*, S^*, \mathit{Acts}, \kappa^*, p^*, c^*, s^*_{in})$. For $\phi \in \mathit{Stat}$, $\psi \in \mathit{Seq}$, we indicate with $s \models \phi$, $\omega \models \psi$ their satisfaction on $s \in S^*$, $\omega \in \bigcup_{s \in S^*} \Omega_s$ respectively. The base cases (1) and the cases for logical connectives are immediate.

Temporal operators. The truth value of $\omega \models \phi$ for a behavior ω and $\phi \in \mathit{Seq}$ is defined in the usual way (see for example [18]).

Path and probabilistic quantifiers. The semantics of the path and probabilistic quantifiers is defined as in pCTL and pCTL* [5]: for $\phi \in \mathit{Seq}$, $0 \le b \le 1$ and $s \in S^*$,

$$s \models \mathrm{A}\phi \quad \textit{iff} \quad \forall \omega \in \Omega_s \,.\, \omega \models \phi \qquad\qquad s \models \mathrm{E}\phi \quad \textit{iff} \quad \exists \omega \in \Omega_s \,.\, \omega \models \phi \tag{7}$$

$$s \models \mathrm{P}_{\bowtie b}\phi \quad \textit{iff} \quad \forall \eta \,.\, \Pr{}^{\eta}_s(\omega \models \phi) \bowtie b \,. \tag{8}$$

The intuitive meaning of (8) is that $\mathrm{P}_{\bowtie b}\phi$ holds at $s \in S$ if a behavior has probability $\bowtie b$ of satisfying ϕ, regardless of the policy.

Average-time operator. Given a behavior $\omega : s_0 a_0 s_1 a_1 \cdots$ and $\phi \in \mathit{Stat}$, let $T_{\omega,\phi} = \min\{i \mid \omega_i \models \phi\}$ be the first position at which ϕ holds along ω, with

$T_{\omega,\phi} = \infty$ if $\forall i \,.\, \omega_i \not\models \phi$. The *first-passage* cost from s to T under policy η is then defined by

$$C^T_{s,\eta} = \mathrm{E}_{s,\eta}\Big\{ \sum_{i=0}^{T_{\omega,\phi}-1} c(\omega_i, \omega_i^a) \Big\} ,$$

where $\mathrm{E}_{s,\eta}\{\cdot\}$ denotes as usual the expectation with respect to the measure μ^η_s. For $s \in S^*$ and $d \geq 0$ we define

$$s \models \mathrm{D}_{\bowtie d}\phi \quad \textit{iff} \quad \forall \eta \,.\, C^T_{s,\eta} \bowtie d \,. \tag{9}$$

The intuitive meaning of (9) is that $\mathrm{D}_{\bowtie d}\phi$ holds at $s \in S$ if the TPNS reaches a ϕ-state in average time $\bowtie d$, regardless of the policy. Note that this definition relies on the fact that the TPNS in non-Zeno: otherwise, the behaviors on which time never advances would affect the value of the first-passage cost.

Definition 3. Given a TPNS Π, a set C of instrumentation clocks and a specification $\phi \in \mathit{Stat}$, let Π^ϕ_C be the instrumented TPNS, and let s^*_{in} be its initial state. We say that Π instrumented with C satisfies ϕ, written $\Pi \models_C \phi$, iff $s^*_{in} \models \phi$. □

4 Model Checking

In this section, we present algorithms to decide whether an instrumented TPNS Π^ϕ_C is non-Zeno, and whether it satisfies a specification ϕ written in pTL or pTL*. Since the logics pTL and pTL* are obtained by adding the operators P and D to CTL and CTL*, we need to examine only the cases corresponding to these additional operators. The algorithms we introduce are based on the properties of certain subsets of states of a TPNS, called *stable sets*.

4.1 Stable Sets

Intuitively, a subset of the state space of a TPNS is *stable* if there is a policy such that all behaviors that enter the subset will never leave it [5]. Let $\mathrm{Supp}(s,a) = \{t \in S \mid p(t \mid s,a) > 0\}$. Stable sets are defined as follows.

Definition 4 (stable sets). Consider a TPNS $\Pi = (\mathcal{P}, S, \mathit{Acts}, \kappa, p, c, s_{in})$ and a subset $B \subseteq S$. We say that B is *stable* if, for all $s \in B$, there is $a \in \kappa(s)$ such that $\mathrm{Supp}(s,a) \subseteq B$. If B is stable, we define the relation

$$\rho_B = \Big\{ (s,t) \in B \times B \;\Big|\; \exists a \in \kappa(s) \,.\, \big(t \in \mathrm{Supp}(s,a) \wedge \mathrm{Supp}(s,a) \subseteq B\big) \Big\} .$$

Intuitively, if $(s,t) \in \rho_B$, then there is an action $a \in \kappa(s)$ that leads from s to t with non-zero probability while leading outside of B with probability 0. If the graph (B, ρ_B) is strongly connected, B is said to be a *strongly connected stable set* (SCSS). Given a subset $C \subseteq S$, we say that B is a *maximal* stable set in C (resp. a maximal SCSS in C) if B is stable (resp. a SCSS) and if there is no other $B' \subseteq C$ such that B' is stable (resp. a SCSS) and $B \subset B'$. □

An equivalent definition of SCSS is provided by the following remark, that we state without proof.

Remark (SCSS and closed recurrent classes). *A set B of states is an SCSS iff there is a Markovian policy η such that B is a closed recurrent class of the Markov chain arising from η.*

The following lemmas summarize the relevant properties of stable sets and SCSS.

Lemma 5. *The following assertions hold.*

1. *Let B be a stable set. Then, there is a policy such that any behavior that enters B will stay in B forever with probability 1.*
2. *Let B be an SCSS. Then, there is a policy such that any behavior that enters B with probability 1 will stay in B forever and will visit all states of B infinitely often.*

Given an infinite behavior ω, we denote by $\mathit{inft}(\omega)$ the set of states that appear infinitely often along ω. The next lemma states that this set is stable with probability 1, under any policy.

Lemma 6. *For all $s \in S$ and all policies η, $\Pr_s^\eta(\mathit{inft}(\omega)$ is an SCSS$) = 1$.*

Proof. Assume, towards the contradiction, that $\Pr_s^\eta(\mathit{inft}(\omega)$ *is an SCSS*$) < 1$. Then, there is a subset $B \subseteq S$ which is not at SCSS, and such that $\Pr_s^\eta(\mathit{inft}(\omega) = B) > 0$. Define $\rho_B = \big\{(t,t') \;\big|\; \exists a \in \kappa(t)\,.\,(t' \in \mathrm{Supp}(t,a) \wedge \mathrm{Supp}(t,a) \subseteq B)\big\}$, as for SCSS. Since B is not an SCSS, there are $t_1, t_2 \in B$ such that there is no path from t_1 to t_2 in (B, ρ_B). Define $\Omega_s^B = \{\omega \in \Omega_s \mid \mathit{inft}(\omega) = B\}$, and let

$$q = \min\Big\{ \sum_{t' \notin B} p(t' \mid t,a) \;\Big|\; t \in B \wedge a \in \kappa(t) \wedge \mathrm{Supp}(t,a) \not\subseteq B \Big\}\,.$$

The lack of a path from t_1 to t_2 in (B, ρ_B) implies that at most a fraction $1-q$ of the behaviors that pass from t_1 can then reach t_2 without leaving B. Since every behavior $\omega \in \Omega_s^B$ contains infinitely many disjoint subsequences from t_1 to t_2, we have that $\Pr_s^\eta(\omega \in \Omega_s^B) \leq (1-q)^k$ for all $k > 0$. As $q > 0$, this implies $\Pr_s^\eta(\mathit{inft}(\omega) = B) = 0$, leading to the required contradiction. □

The following corollary states that if a behavior is eventually confined in a set C, with probability 1 it is confined in the union of the SCSS maximal in C.

Corollary 7. *Consider any subset C of states of a TPNS. Let $D_1, \ldots, D_n$ be the SCSS that are maximal in C, and let $D = \bigcup_{i=1}^n D_i$. For any $s \in S$ and any policy η, $\Pr_s^\eta(\mathit{inft}(\omega) \subseteq C \wedge \mathit{inft}(\omega) \not\subseteq D) = 0$.*

Proof. Assume, towards the contradiction, that $\Pr_s^\eta(\mathit{inft}(\omega) \subseteq C \wedge \mathit{inft}(\omega) \not\subseteq D) > 0$. Then, by Lemma 6 there is an SCSS $B \subseteq C$ such that $B \not\subseteq D$ and $\Pr_s^\eta(\mathit{inft}(\omega) = B) > 0$. This contradicts the fact that D is the union of all SCSS maximal in C. □

Define the size $|\Pi|$ of a TPNS $\Pi = (\mathcal{P}, S, Acts, \kappa, p, c, s_{in})$ to be the length of its encoding, where we assume that p and c are represented by listing, for all $s, t \in S$ and $a \in \kappa(s)$, the values of $p(t \mid s, a)$ and $c(s, a)$ as fixed-precision binary numbers. Given a subset $C \subseteq S$, the following algorithm computes the maximal stable set $B \subseteq C$ and the maximal SCSS in C in time polynomial in $|\Pi|$.

Algorithm 8 (stable sets and SCSS).

Input: TPNS $\Pi = (\mathcal{P}, S, Acts, \kappa, p, c, s_{in})$, and a subset $C \subseteq S$.

Output: The stable set B maximal in C, and the list $E_1, \ldots, E_n$ of SCSS maximal in C.

Procedure: Define the functional $\Lambda : 2^S \mapsto 2^S$ by $\Lambda(D) = \{s \in D \mid \exists a \in \kappa(s) \,.\, \mathrm{Supp}(s, a) \subseteq D\}$. Then $B = \lim_{k \to \infty} \Lambda^k(C) = \Lambda^\infty(C)$, and the computation of the limit requires at most $|C|$ iterations, since the functional is monotonic. The SCSS $E_1, \ldots, E_n$ can be computed by computing the maximal strongly connected components of the graph (B, ρ_B). □

4.2 Checking Non-Zenoness

To check whether a TPNS is non-Zeno, we introduce *Zeno* stable sets.

Definition 9 (Zeno stable sets). Given a TPNS $\Pi_c^\phi = (\mathcal{P}, S, Acts, \kappa, p, c, s_{in})$, a *Zeno stable set* (ZSS) is a stable subset $B \subseteq S$ such that for every $s \in B$, there is an action $a \in \kappa(s)$ with $\mathrm{Supp}(s, a) \subseteq B$ and $c(s, a) = 0$. A Zeno stable set is *maximal* if it is not the proper subset of any other Zeno stable set. □

Note that since the union of two ZSS is still a ZSS, every TPNS has a single (possibly empty) maximal ZSS. The maximal ZSS can be computed in time polynomial in the size of the TPNS by the following algorithm.

Algorithm 10 (computation of maximal ZSS).

Input: A TPNS $(\mathcal{P}, S, s_{in}, \kappa, p, c)$.

Output: The maximal ZSS B of the TPNS.

Procedure: Define the functional $\Lambda : 2^S \mapsto 2^S$ by $\Lambda(D) = \left\{s \in D \;\middle|\; \exists a \in \kappa(s).\big(\mathrm{Supp}(s, a) \subseteq D \wedge c(s, a) = 0\big)\right\}$. Then $B = \Lambda^\infty(S)$, and the computation requires at most $|S|$ iterations. □

The following theorem states that to check that the TPNS Π_C^ϕ is non-Zeno it suffices to check that there are no reachable ZSS, which by the above results can be done in time polynomial in $|\Pi_C^\phi|$.

Theorem 11 (checking non-Zenoness). *A TPNS is non-Zeno iff it does not contain any non-empty ZSS reachable in (S, ρ_S) from the initial state.*

Proof. If there is a ZSS reachable from the initial state, it is easy to see that there is also a policy under which the system is non-Zeno. In the other direction, assume that the TPNS does not contain any ZSS reachable from the initial state, and consider any policy η. By Lemma 6, $\Pr_{s_{in}}^\eta(\mathit{inft}(\omega)$ *is a SCSS and not a ZSS*$) = 1$. From this it can be shown that every path must take with probability 1 infinitely many actions with cost bounded away from 0, leading to the desired conclusion. □

4.3 Model Checking of pTL* Formulas

The model checking algorithms we present share the same basic structure of those proposed in [11] for CTL and CTL*. Given a TPNS Π_C^ϕ and a formula $\phi \in \mathit{Stat}$, the algorithms recursively evaluate the truth values of the state subformulas of ϕ at all states $s \in S$, following the recursive definitions (1)–(3), until the truth value of ϕ itself can be computed at all $s \in S$. For brevity, we will present algorithms only the logic pTL*, since pTL model checking can be done by combining the results of [5] with the methods presented for the D operator.

Model Checking for the Operator P

From definition (8), to compute whether $s \models \mathrm{P}_{\bowtie b}\psi$ for a state s and a formula $\psi \in \mathit{Seq}$ it suffices to consider the minimum and maximum probabilities with which a computation from s satisfies ψ. Even though these probabilities can be computed using the algorithm presented in [6], we will follow here a different approach. The algorithm we present relies on the properties of stable sets, and shares the insights of the one presented in [5]. However, by relying on the determinization of ω-automata instead of on canonical forms for temporal formulas, the algorithm achieves the optimal complexity bound of the one presented in [6] while exhibiting a relatively simple structure. This algorithm has been recently extended by [17] to logics with fairness assumptions on the policies.

The algorithm. By the results of [6, 5] there are optimal policies η^- and η^+ that minimize and maximize, respectively, the probability $\Pr_s^\eta(\omega \models \psi)$. From (8), to compute whether $s \models \mathrm{P}_{\bowtie b}\psi$ it suffices to compute either the minimum probability $\Pr_s^-(\omega \models \psi) = \Pr_s^{\eta^-}(\omega \models \psi)$ or the maximum one $\Pr_s^+(\omega \models \psi) = \Pr_s^{\eta^+}(\omega \models \psi)$, depending on the direction of the inequality $\bowtie$. Since $\Pr_s^-(\omega \models \psi) = 1 - \Pr_s^+(\omega \models \neg\psi)$, it suffices to give an algorithm for the computation of $\Pr_s^+(\omega \models \psi)$.

Let $\alpha_1, \ldots, \alpha_n \in \mathit{Stat}$ be the maximal state subformulas of ψ, i.e. the state subformulas of ψ that are not proper subformulas of any other state subformula of ψ. Define $\psi' = \psi[r_1/\alpha_1] \ldots [r_n/\alpha_n]$ to be the result of replacing each α_i with a new propositional symbol r_i. The formula ψ' is therefore a linear-time temporal formula constructed from $r_1, \ldots, r_n$ using the temporal operators $\Box$, $\Diamond$, $\mathcal{U}$. As the truth values of $\alpha_1, \ldots, \alpha_n$ have already been computed at all states, we define the *label* $l(t)$ of $t \in S$ by $l(t) = \{r_i \mid 1 \leq i \leq n \wedge t \models \alpha_i\}$.

It is known from automata theory that ψ' can be translated into a deterministic Rabin automaton $DR_{\psi'} = (Q, q_{in}, \Sigma, \gamma, U)$ with state space Q, initial state $q_{in} \in Q$, alphabet $\Sigma = 2^{\{r_1,\ldots,r_n\}}$, transition relation $\gamma : Q \times \Sigma \mapsto Q$, and acceptance condition U [24, 20, 21]. The acceptance condition is a list $U = \{(H_1, L_1), \ldots, (H_m, L_m)\}$ of pairs of subsets of Q. An infinite sequence $\sigma : b_0 b_1 b_2 \cdots$ of symbols of Σ is accepted by $DR_{\psi'}$ if it induces a sequence $\omega_\sigma : q_0 q_1 q_2 \cdots$ of states of Q s.t. $q_0 = q_{in}$, $\gamma(q_i, b_i) = q_{i+1}$ for all $i \geq 0$ and, for some $1 \leq j \leq m$, it is $\mathit{inft}(\omega_\sigma) \subseteq H_j$ and $\mathit{inft}(\omega_\sigma) \cap L_j \neq \emptyset$.

Given $\Pi_C^\phi = (\mathcal{P}, S, \mathit{Acts}, \kappa, p, c, s_{in})$, $DR_{\psi'} = (Q, q_{in}, \Sigma, \gamma, U)$ and $s \in S$ we construct the *product TPNS* $\Pi' = (\mathcal{P}, S', \mathit{Acts}, \kappa', p', c', s'_{in})$, where:

1. $S' = S \times Q$, where $(t,q)[\![p]\!] = t[\![p]\!]$ for all $(t,q) \in S'$ and $p \in \mathcal{P}$.
2. For $(t,q) \in S'$, $\kappa'(t,q) = \kappa(t)$.
3. For each $t \in S$ and $a \in \kappa(t)$, the probability $p'((t',q') \mid (t,q),a)$ of a transition to $(t',q') \in S'$ is equal to $p(t' \mid t,a)$ if $\gamma(q,l(t')) = q'$, and is equal to 0 otherwise.
4. For $(t,q) \in S'$ and $a \in \kappa(t)$, $c((t,q),a) = c(t,a)$.
5. $s'_{in} = (s, \gamma(q_{in}, l(s)))$.

Each pair (H_i, L_i), $1 \leq i \leq m$, induces a related pair (H'_i, L'_i) defined by $H'_i = S \times H_i$, $L'_i = S \times L_i$. For each pair (H'_i, L'_i), $1 \leq i \leq m$, we let $B_1^{(i)}, \ldots, B_{n_i}^{(i)}$ be the SCSS maximal in H'_i and having non-empty intersection with L'_i, and we let $T = \bigcup_{i=1}^{m} \bigcup_{j=1}^{n_i} B_j^{(i)}$. By the previous results, the set T can be computed in time polynomial in $|\Pi'|$. The following theorem states that to compute $\Pr_s^+(\omega \models \psi)$ it suffices to compute the maximum probability of reaching T from s'_{in} in Π'. As discussed in [9, 6], maximum reachability probabilities can be computed by solving a linear programming problem in time polynomial in $|\Pi'|$.

Theorem 12. $\Pr_s^+(\omega \models \psi) = \sup_\eta \Pr_{s'_{in}}^\eta(\exists k \,.\, \omega_k \in T)$.

Proof. Let $R_{\omega,T} \equiv \exists k \,.\, \omega_k \in T$. By construction of Π', it is $\Pr_s^+(\psi) = \sup_\eta \Pr_{s'_{in}}^\eta(\omega \models \psi')$. We can write

$$\Pr_{s'_{in}}^\eta(\omega \models \psi') \;=\; \Pr_{s'_{in}}^\eta(R_{\omega,T} \wedge \omega \models \psi') + \Pr_{s'_{in}}^\eta(\neg R_{\omega,T} \wedge \omega \models \psi') \;. \qquad (10)$$

A behavior $\omega \in \Omega_{s'_{in}}$ that satisfies ψ' must, for some $1 \leq i \leq m$, (a) be eventually confined to H'_i, (b) visit infinitely often L'_i. From (a), by Corollary 7, with probability 1 the behavior is eventually confined to the union of the SCSS in H'_i. From (b), the behavior can be eventually confined only to the SCSS in H'_i that have non-empty intersection with L'_i, that is, to $B_1^{(i)}, \ldots, B_{n_i}^{(i)}$. Since $\bigcup_{j=1}^{n_i} B_j^{(i)} \subseteq T$, a behavior that satisfies ψ' will enter T with probability 1, so that the second term on the right side of (10) is 0, and (10) reduces to $\Pr_{s'_{in}}^\eta(\omega \models \psi') = \Pr_{s'_{in}}^\eta(R_{\omega,T} \wedge \omega \models \psi')$. Taking $\sup_\eta$ of both sides and using Lemma 5, we have

$$\sup_\eta \Pr_{s'_{in}}^\eta(\omega \models \psi') \;=\; \sup_\eta \Pr_{s'_{in}}^\eta(R_{\omega,T} \wedge \omega \models \psi') \;=\; \sup_\eta \Pr_{s'_{in}}^\eta(R_{\omega,T}) \;,$$

and the result follows. □

Model Checking for the Operator D

Given $\psi \in \mathit{Stat}$ and $b \geq 0$, consider the problem of computing the truth value of $\mathrm{D}_{\bowtie b}\psi$ at state $s \in S$ of a TPNS Π_C^ϕ. We assume that Π_C^ϕ is non-Zeno, and that the truth value of ψ has already been computed at all states of S. Let $T = \{s \in S \mid s \models \psi\}$ be the set of states satisfying ψ. From (9), to decide whether $s \models \mathrm{D}_{\bowtie b}\psi$ we need to compute $\inf_\eta C_{s,\eta}^T$, $\sup_\eta C_{s,\eta}^T$. This corresponds to the computation of the minimum and maximum first-passage costs of a Markov decision process. As discussed in [9, 8, 4], these costs can be computed by solving linear-programming problems, which require time polynomial in $|\Pi_C^\phi|$.

Complexity of Model Checking

Combining the results of the previous sections with the results of [6, 5], we get the following theorem.

Theorem 13. *Given a TPNS Π, the following assertions hold:*

1. *Checking whether Π is non-Zeno can be done in polynomial time in $|\Pi|$.*
2. *Model checking a pTL formula ϕ with set of instrumentation clocks C has time-complexity linear in $|\phi|$ and polynomial in $|\Pi|$ and $\prod_{\xi \in C}(M_\xi + 1)$.*
3. *Model checking a pTL* formula ϕ with set of instrumentation clocks C has time-complexity doubly exponential in $|\phi|$, and polynomial in $|\Pi|$ and $\prod_{\xi \in C}(M_\xi + 1)$.*

5 Extending the D Operator to Past Formulas

To conclude, we discuss an extension of the logics pTL and pTL* that increases the expressive power of the logics by allowing the formula ϕ in $\mathrm{D}_{\bowtie b}\phi$ to be a *past* temporal formula, instead of a state formula. A past temporal formula is a formula constructed from state formulas in *Stat* using the temporal operators $\boxminus$, $\diamondminus$ and $\mathcal{S}$ [18].[2]

The semantics of this extension can be defined as follows. Given a behavior ω and a past formula ϕ, let $T_{\omega,\phi} = \min\{i \mid \omega_0 \cdots \omega_i \models_i \phi\}$, where $\omega_0 \cdots \omega_i \models_i \phi$ indicates that ϕ holds at the last position i of the finite sequence of states $\omega_0 \cdots \omega_i$. The truth value of $\mathrm{D}_{\bowtie b}\phi$ can be defined as in (9).

This extended version of the D operator can be model checked by combining the techniques of [5] with the algorithms presented in the previous section. Specifically, given a TPNS Π_C^ϕ and a subformula $\mathrm{D}_{\bowtie b}\psi$ of ϕ, it is possible to construct a TPNS $\Pi_C^{\psi,\phi}$ in which the states keep track of the truth values of the past subformulas of ψ (ψ itself included). The truth value of $\mathrm{D}_{\bowtie b}\psi$ can be computed by applying the algorithms of Section 4.3 to the TPNS $\Pi_C^{\psi,\phi}$. Since the complexity of the model checking is dominated by the doubly-exponential dependency arising from the operator P, the bounds expressed by Theorem 13 apply also to this extended version of the logic.

Acknowledgments. We wish to thank Andrea Bianco for many inspiring discussions and suggestions.

References

1. R. Alur and D. Dill. The theory of timed automata. In *Real-Time: Theory in Practice*, volume 600 of *Lect. Notes in Comp. Sci.*, pages 45–73. Springer-Verlag, 1991.
2. A. Aziz, V. Singhal, F. Balarin, R.K. Brayton, and A.L. Sangiovanni-Vincentelli. It usually works: The temporal logic of stochastic systems. In *Computer Aided Verification*, volume 939 of *Lect. Notes in Comp. Sci.* Springer-Verlag, 1995.

[2] The use of past temporal operators in non-probabilistic branching-time logics has been discussed in depth in [16].

3. D. Beauquier and A. Slissenko. Polytime model checking for timed probabilistic computation tree logic. Technical Report TR-96-08, Dept. of Informatics, Univ. Paris-12, April 1996.
4. D.P. Bertsekas. *Dynamic Programming.* Prentice-Hall, 1987.
5. A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In *Found. of Software Tech. and Theor. Comp. Sci.*, volume 1026 of *Lect. Notes in Comp. Sci.*, pages 499–513. Springer-Verlag, 1995.
6. C. Courcoubetis and M. Yannakakis. Markov decision processes and regular events. In *ICALP'90*, volume 443 of *Lect. Notes in Comp. Sci.*, pages 336–349. Springer-Verlag, 1990.
7. L. de Alfaro. Formal verification of performance and reliability of real-time systems. Technical Report STAN-CS-TR-96-1571, Stanford University, June 1996.
8. E.V. Denardo. Computing a bias-optimal policy in a discrete-time markov decision problem. *Operations Research*, 18:279–289, 1970.
9. C. Derman. *Finite State Markovian Decision Processes.* Acedemic Press, 1970.
10. E.A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, chapter 16, pages 995–1072. Elsevier Science Publishers (North-Holland), Amsterdam, 1990.
11. E.A. Emerson and C.L. Lei. Modalities for model checking: Branching time strikes back. In *Proc. 12th ACM Symp. Princ. of Prog. Lang.*, pages 84–96, 1985.
12. H. Hansson. *Time and Probability in Formal Design of Distributed Systems.* Elsevier, 1994.
13. H. Hansson and B. Jonsson. A framework for reasoning about time and reliability. In *Proc. of Real Time Systems Symposium*, pages 102–111. IEEE, 1989.
14. J.G. Kemeny, J.L. Snell, and A.W. Knapp. *Denumerable Markov Chains.* D. Van Nostrand Company, 1966.
15. Y. Kesten, Z. Manna, and A. Pnueli. Verifying clocked transition systems. In *Hybrid Systems III*, volume 1066 of *Lect. Notes in Comp. Sci.*, pages 13–40. Springer-Verlag, 1996.
16. O. Kupferman and A. Pnueli. Once and for all. In *Proc. 10th IEEE Symp. Logic in Comp. Sci.*, pages 25–35, 1995.
17. M. Kwiatkowska and C. Baier. Model checking for a probabilistic branching time logic with fairness. Technical Report CSR-96-12, University of Birmingham, June 1996.
18. Z. Manna and A. Pnueli. Models for reactivity. *Acta Informatica*, 30:609–678, 1993.
19. M.L. Puterman. *Markov Decision Processes.* John Wiley and Sons, 1994.
20. S. Safra. On the complexity of ω-automata. In *Proc. 29th IEEE Symp. Found. of Comp. Sci.*, 1988.
21. S. Safra. Exponential determinization for ω-automata with strong-fairness acceptance condition. In *Proc. ACM Symp. Theory of Comp.*, pages 275–282, 1992.
22. R. Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems.* PhD thesis, MIT, June 1995. Technical Report MIT/LCS/TR-676.
23. R. Segala and N.A. Lynch. Probabilistic simulations for probabilistic processes. In *CONCUR'94*, volume 836, pages 481–496. Springer-Verlag, 1994.
24. M.Y. Vardi and P. Wolper. An automata-theoretic approach to automatic program verification. In *Proc. First IEEE Symp. Logic in Comp. Sci.*, pages 332–344, 1986.