Model Checking of Probabilistic and Nondeterministic Systems*

Andrea Bianco[1] and Luca de Alfaro[2]

[1] Politecnico di Torino, Italy. bianco@polito.it
[2] Stanford University, USA. luca@cs.stanford.edu

Abstract. The temporal logics pCTL and pCTL* have been proposed as tools for the formal specification and verification of probabilistic systems: as they can express quantitative bounds on the probability of system evolutions, they can be used to specify system properties such as reliability and performance. In this paper, we present model-checking algorithms for extensions of pCTL and pCTL* to systems in which the probabilistic behavior coexists with nondeterminism, and show that these algorithms have polynomial-time complexity in the size of the system. This provides a practical tool for reasoning on the reliability and performance of parallel systems.

1 Introduction

Temporal logic has been successfully used to specify the behavior of concurrent and reactive systems. These systems are usually modeled as nondeterministic processes: at any moment in time, more than one future evolution may be possible, but a probabilistic characterization of their likelihood is normally not attempted. While many important system properties can be studied in this setting, others, such as reliability and performance, require instead a probabilistic characterization of the system.

The first applications of temporal logic to probabilistic systems consisted in studying which temporal logic properties are satisfied with probability 1 by systems modeled either as finite Markov chains [14, 18, 12, 1, 20] or as augmented Markov models exhibiting both nondeterministic and probabilistic behavior [22, 19, 5, 20].

Subsequently, [10, 2] considered systems modeled by discrete Markov chains, and introduced the logics pCTL and pCTL*, that can express quantitative bounds on the probability of system evolutions. These logics can thus be used to reason on the reliability and performance of systems. They are obtained by adding to the branching time logics CTL and CTL* a probabilistic operator $\mathbb{P}$,

* This research was supported in part by the National Science Foundation under grant CCR-92-23226, by the Advanced Research Projects Agency under NASA grant NAG2-892, by the United States Air Force Office of Scientific Research under grant F49620-93-1-0139, by Department of the Army under grant DAAH04-95-1-0317, and by the Italian National Research Council.

such that the formula $\mathbb{P}_{\geq a}\phi$ is true at a given point of the system evolution if, starting from that point, the probability that a future evolution satisfies ϕ is at least a.

The model-checking algorithms presented in [10, 2] can be used to determine the validity of pCTL and pCTL* formulas on systems modeled by finite Markov chains. Moreover, [2] considers generalized Markov processes, representing families of Markov chains, and shows that the decision problem for pCTL* formulas on generalized Markov processes is decidable using results from the theory of real closed fields. However, no efficient computational method is given for this latter problem.

In this paper, we extend the logics pCTL and pCTL* to systems in which nondeterministic and probabilistic behavior coexist. We model these systems by *probabilistic-nondeterministic systems*, similar to the augmented Markov models of [19, 20]. Due to the presence of nondeterminism it is not possible, in general, to talk about the probability with which a formula is satisfied, but only about the lower and upper bounds of such probability. Therefore, according to our definition, the formula $\mathbb{P}_{\geq a}\phi$ (resp. $\mathbb{P}_{\leq a}\phi$) is true at a given point of the system evolution if a system evolution starting from that point satisfies ϕ with a probability bounded from below (resp. above) by a. We then present model-checking algorithms that verify whether a system satisfies a specification written in pCTL or pCTL* in polynomial time in the size of the description of the system.

The logics pCTL and pCTL*, together with these model-checking algorithms, provide a practical tool for the formal specification and verification of the performance and reliability of parallel systems. Nondeterminism, as already recognized by [22, 19, 5, 20], is in fact the key to the natural modeling of parallel probabilistic systems by interleaving, as it allows us to model the choice of which system in the parallel composition takes a transition. Nondeterminism also gives the flexibility of leaving some transition probabilities unspecified. This leads to simpler system models, and it is necessary when some transition probabilities are unknown. Leaving some transition probabilities unspecified can also be useful when it is not desirable that a correctness proof of the system with respect to some specification depends on the value of those probabilities.

2 Probabilistic-Nondeterministic Systems

Following an approach similar to [19, 20], we use *Probabilistic-Nondeterministic Systems* (PNS) to model systems in which probabilistic and nondeterministic components of the behavior coexist. To give a formal definition of PNS, we first introduce *next-state probability distributions*.

Definition 1 (next-state probability distribution). If S is the state space of a system, a *next-state probability distribution* is a function $p : S \mapsto [0, 1]$ such that $\sum_{s \in S} p(s) = 1$. For $s \in S$, $p(s)$ represents the probability of making a direct transition to s from the current state. $\square$

A PNS can then be defined as follows.

Definition 2 (PNS). A PNS is a quadruple $\Pi = (S, s_{\text{in}}, V, \tau)$, where:

1. S is the denumerable or finite state space of the system;
2. $s_{\text{in}} \in S$ is the initial state;
3. V is a labeling function that associates with each $s \in S$ the set $V(s) \subseteq \mathcal{P}$ of propositional variables that are true in s;
4. τ is a function that associates with each $s \in S$ the set $\tau(s) = \{p_1^s, \ldots, p_{k_s}^s\}$ of next-state probability distributions from s. We denote $|\tau(s)|$ by k_s. □

The successor of a state $s \in S$ is chosen according to a two-phase process: first, a next-state probability distribution $p_i^s \in \tau(s)$ is selected nondeterministically among $p_1^s, \ldots, p_{k_s}^s$; second, a successor state $t \in S$ is chosen according to the probability distribution p_i^s on S.

This model, based on the one proposed in [19], generalizes the approach of [22] by allowing a simpler encoding of the parallel composition of systems. To see how parallelism can be modeled by a PNS, consider as an example the parallel composition of m Markov chains $A_1, \ldots, A_m$. In a PNS Π representing $A_1 \parallel A_2 \parallel \ldots \parallel A_m$, we can associate with each state $s \in S$ the next-state distributions $\tau(s) = \{p_1^s, \ldots, p_m^s\}$, where the distribution p_i^s arises from a move taken by the chain A_i, $1 \leq i \leq m$. In this way, the probabilistic information on the behavior of each chain is preserved in Π, and the choice of the Markov chain that takes a transition is nondeterministic.

We define a *reachability* relation $\rho \subseteq S \times S$ by

$$\rho = \{(s,t) \mid \exists p^s \in \tau(s) \,.\, p^s(t) > 0\} \,.$$

Then, we associate with each state $s \in S$ the set

$$\Omega_s = \{s_0 s_1 s_2 \ldots \mid s = s_0 \wedge \forall n \in \mathbb{N} \,.\, \rho(s_n, s_{n+1})\}$$

of *legal* infinite sequences of states beginning at s. The set of computations of a system Π is thus $\Omega_{s_{\text{in}}}$. For $\omega \in \Omega_s$, we denote with $\omega|_n$ the n-th state of ω, with $\omega|_0 = s$.

Moreover, we let $\mathcal{B}_s \subseteq 2^{\Omega_s}$ be the smallest algebra of subsets of Ω_s that contains all the *basic cylinder sets* $\{\omega \in \Omega_s \mid \omega|_0 = s_0 \wedge \ldots \wedge \omega|_n = s_n\}$ for all $n \geq 0$, $s_0, \ldots, s_n \in S$, and that is closed under complement and countable unions and intersections. This algebra is called the *Borel σ-algebra* of basic cylinder sets, and its elements are the *measurable* sets of sequences, to which it will be possible to assign a probability [13].

Minimal and Maximal Probabilities

Due to the presence of nondeterminism, we cannot define a probability measure on the Borel σ-algebra $\mathcal{B}_s$. However, for each set of sequences $\Delta \in \mathcal{B}_s$, we can define its *maximal probability* $\mu_s^+(\Delta)$ and its *minimal probability* $\mu_s^-(\Delta)$. Intuitively, $\mu_s^+(\Delta)$ (resp. $\mu_s^-(\Delta)$) represents the probability that the system follows a sequence in Δ provided that the nondeterministic choices are as favorable (resp. unfavorable) as possible. To formalize the idea of favorable and unfavorable choices, we introduce the concept of *strategies* (similar to the *schedules* of [22, 19, 5, 20]), that determine which next-state probability distribution is chosen for each state.

If the system reaches the root s of Ω_s following the sequence $s_{\rm in} s_1 \ldots s_n s$, we can assume that a strategy does not depend on the "past" sequence $\omega_p = s_{\rm in} s_1 \ldots s_n$. In fact, we are interested in a strategy that maximizes or minimizes the probability that the system, starting from s, follows a sequence in Δ: as neither Δ nor the next-state distributions depend on ω_p, such strategy also need not depend on ω_p. Formally, a strategy is defined as follows.

Definition 3 (strategy). A *strategy* η is a set of conditional probabilities $Q_\eta(i \mid s_0 s_1 \ldots s_n)$ such that $\sum_{i=1}^{k_{s_n}} Q_\eta(i \mid s_0 s_1 \ldots s_n) = 1$, for all $n \in \mathbb{N}$, $s_0, s_1, \ldots, s_n \in S$, and $1 \leq i \leq k_{s_n}$. □

When a system behaves according to a strategy η in the evolution from $s_0 \in S$, and has reached s_n following the sequence $s_0 \ldots s_n$, it will choose the next-state distribution $p_i^{s_n}$ with probability $Q_\eta(i \mid s_0 s_1 \ldots s_n)$. The probability $\Pr_\eta(t \mid s_0 \ldots s_n)$ that a direct transition to t is taken next is thus equal to $\sum_{i=1}^{k_{s_n}} Q_\eta(i \mid s_0 s_1 \ldots s_n) p_i^{s_n}(t)$.

Therefore, we can associate with each finite sequence $s_0 \ldots s_n$ starting at the root $s = s_0$ of Ω_s the probability $\prod_{i=0}^{n-1} \Pr_\eta(s_{i+1} \mid s_0 \ldots s_i)$. These probabilities for the finite sequences give rise to a unique probability measure $\mu_{s,\eta}$ on $\mathcal{B}_s$ that associates with each $\Delta \in \mathcal{B}_s$ its probability $\mu_{s,\eta}(\Delta)$ [13]. We can then define minimal and maximal probabilities as follows.

Definition 4 (minimal and maximal probability). The *minimal and maximal probabilities* $\mu_s^-(\Delta)$, $\mu_s^+(\Delta)$ of a set of sequences $\Delta \in \mathcal{B}_s$ are defined by

$$\mu_s^-(\Delta) = \inf_\eta \mu_{s,\eta}(\Delta) \qquad \mu_s^+(\Delta) = \sup_\eta \mu_{s,\eta}(\Delta) \qquad \square$$

Thus, $\mu_s^-(\Delta)$ and $\mu_s^+(\Delta)$ represent the probability with which the system follows an evolution $s s_1 s_2 \ldots \in \Delta$ when the nondeterministic choices are as unfavorable or as favorable as possible, respectively. In general, μ^+ and μ^- are not additive on $\mathcal{B}_s$, as the following lemma states.

Lemma 5. *If $\Delta_1, \Delta_2 \in \mathcal{B}_s$, with $\Delta_1 \cap \Delta_2 = \emptyset$, then*

$$\mu_s^-(\Delta_1 \cup \Delta_2) \geq \mu_s^-(\Delta_1) + \mu_s^-(\Delta_2) \qquad \mu_s^+(\Delta_1 \cup \Delta_2) \leq \mu_s^+(\Delta_1) + \mu_s^+(\Delta_2)$$

and equality does not hold in general.

The minimal and maximal probability are related by the following lemma.

Lemma 6. *For $\Delta \in \mathcal{B}_s$, it is $\mu_s^-(\Delta) = 1 - \mu_s^+(\Omega_s - \Delta)$.*

Proof. From $\mu_{s,\eta}(\Delta) = 1 - \mu_{s,\eta}(\Omega_s - \Delta)$, we have $\mu_s^-(\Delta) = \inf_\eta \mu_{s,\eta}(\Delta) = \inf_\eta \big(1 - \mu_{s,\eta}(\Omega_s - \Delta)\big) = 1 - \sup_\eta \mu_{s,\eta}(\Omega_s - \Delta) = 1 - \mu_s^+(\Omega_s - \Delta)$. □

3 Probabilistic Temporal Logic

Syntax. The logics pCTL and pCTL* are derived from the branching-time logics CTL and CTL* [6] by introducing a probabilistic operator $\mathbb{P}$, with the intuitive reading that $\mathbb{P}_{\geq a}\phi$ (resp. $\mathbb{P}_{\leq a}\phi$) means that the probability of ϕ holding in the future evolution of the system is at least (resp. at most) a [10, 11, 9, 2]. Formally, we distinguish two classes of 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). For pCTL*, the classes *Stat* and *Seq* are defined as follows:

$$\mathcal{P} \subseteq \mathit{Stat} \tag{1}$$

$$\phi,\ \psi \in \mathit{Stat} \implies \phi \wedge \psi,\ \neg\phi \in \mathit{Stat} \tag{2}$$

$$\phi \in \mathit{Seq} \implies \mathrm{A}\phi,\ \mathrm{E}\phi,\ \mathbb{P}_{\bowtie a}\phi \in \mathit{Stat} \tag{3}$$

$$\phi \in \mathit{Stat} \implies \phi \in \mathit{Seq} \tag{4}$$

$$\phi,\ \psi \in \mathit{Seq} \implies \phi \wedge \psi,\ \neg\phi,\ \Box\phi,\ \Diamond\phi,\ \phi\, \mathcal{U}\, \psi \in \mathit{Seq}\ . \tag{5}$$

In the above definition, $\bowtie$ stands for one of $<, \leq, \geq, >$, and $a \in [0,1]$. The logic pCTL is a restricted version of pCTL*, and its definition can be obtained by replacing the clauses (4), (5) in the above definition with the single clause

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

Semantics. For a formula $\phi \in \mathit{Stat}$, we indicate with $s \models \phi$ its satisfaction on state $s \in S$, and for $\phi \in \mathit{Seq}$ we indicate with $\omega \models \phi$ its satisfaction on the infinite state sequence ω. The semantics of the logical connectives and of the temporal operators is defined in the usual way; the semantics of A, E and $\mathbb{P}$ are defined as follows:

$$s \models \mathrm{A}\phi \ \ \mathit{iff}\ \ \forall \omega \in \Omega_s \,.\, \omega \models \phi \tag{7}$$

$$s \models \mathrm{E}\phi \ \ \mathit{iff}\ \ \exists \omega \in \Omega_s \,.\, \omega \models \phi \tag{8}$$

$$s \models \mathbb{P}_{\geq a}\phi \ \ \mathit{iff}\ \ \mu_s^-(\{w \in \Omega_s \mid \omega \models \phi\}) \geq a \tag{9}$$

$$s \models \mathbb{P}_{\leq a}\phi \ \ \mathit{iff}\ \ \mu_s^+(\{w \in \Omega_s \mid \omega \models \phi\}) \leq a\ . \tag{10}$$

The semantics of $s \models \mathbb{P}_{>a}\phi$, $s \models \mathbb{P}_{<a}\phi$ are defined in a similar way. This definition has a very intuitive reading: if $s \models \mathbb{P}_{\geq a}\phi$, it means that regardless of the choices made in nondeterministic states, the probability that the future evolution satisfies ϕ is at least a (and similarly for $s \models \mathbb{P}_{\leq a}\phi$).

To see that the semantics is well-defined, it is possible to show by induction on the structure of ϕ that $\{\omega \in \Omega_s \mid \omega \models \phi\} \in \mathcal{B}_s$ for every $\phi \in \mathit{Seq}$ [22]. We say that a formula $\phi \in \mathit{Stat}$ is satisfied by a PNS Π, written $\Pi \models \phi$, if $s_{\mathrm{in}} \models \phi$.

4 Model Checking

We now present algorithms to decide whether a PNS Π with finite state space S satisfies a specification ϕ written in pCTL or pCTL*. We will prove that these algorithms have polynomial time complexity in the size of the description of Π. We first give the algorithm for pCTL, and then we examine the one for pCTL*.

The algorithms share the same basic structure of those proposed in [8, 7] for CTL and CTL*. Given a formula $\phi \in \mathit{Stat}$, they recursively evaluate the truth-values of the state subformulas $\psi \in \mathit{Stat}$ of ϕ at all states $s \in S$, starting from the propositional formulas of ϕ and following the recursive definitions (1)–(3) of state formulas, until the truth-value of ϕ itself can be computed at all $s \in S$.

In fact, since pCTL and pCTL* differ from CTL and CTL* only for the presence of the $\mathbb{P}$ operator, we can use the same techniques proposed for CTL and CTL* to deal with the operators $\wedge$, $\neg$, A, E. In the algorithms below, therefore, we need to examine only the case corresponding to $\mathbb{P}$.

4.1 pCTL Formulas

Let $\mathrm{Pr}_s^+ \phi \stackrel{\mathrm{def}}{=} \mu_s^+(\{w \in \Omega_s \mid \omega \models \phi\})$, $\mathrm{Pr}_s^- \phi \stackrel{\mathrm{def}}{=} \mu_s^-(\{w \in \Omega_s \mid \omega \models \phi\})$. From (9), (10) we see that in order to check whether $s \models \mathbb{P}_{\bowtie a}\phi$ it suffices to compute $\mathrm{Pr}_s^+ \phi$, $\mathrm{Pr}_s^- \phi$. Using $\Box\psi \leftrightarrow \neg\Diamond\neg\psi$, $\Diamond\psi \leftrightarrow \mathbf{true}\,\mathcal{U}\,\psi$, and the relations

$$\mathrm{Pr}_s^+ \neg\phi = 1 - \mathrm{Pr}_s^- \phi \qquad \mathrm{Pr}_s^- \neg\phi = 1 - \mathrm{Pr}_s^+ \phi \ ,$$

derived from Lemma 6, we need only to consider the case of $\phi = \gamma\,\mathcal{U}\,\psi$. Let $S_d = \{s \in S \mid s \models \psi\}$ be the set of "destination" states, and let $S_p = \{s \in S \mid s \models \gamma\}$ be the set of "intermediate" states.

Computation of $\mathrm{Pr}_s^- \phi$. It is useful to determine, first of all, for which states $s \in S$ is $\mathrm{Pr}_s^- \phi > 0$. To this end, let the monotone set function $\Lambda : 2^S \mapsto 2^S$ be such that, for $A \subseteq S$,

$$\Lambda(A) = A \cup \big\{ s \in S_p \;\big|\; \forall i \in \{1, \ldots, k_s\} \,.\, \exists t \,.\, \big(t \in A \wedge p_i^s(t) > 0\big) \big\} \ .$$

As S is finite, the fixpoint $\Lambda^\infty(A) = \bigcup_{i=0}^{\infty} \Lambda^i(A)$ is computable in at most $|S|$ iterations. Let $S_{>0} = \Lambda^\infty(S_d)$. The following lemma states that this is exactly the set of states from which ϕ can be true with probability greater than 0.

Lemma 7. $s \in S - S_{>0}$ *implies* $\mathrm{Pr}_s^- \phi = 0$, $\;s \in S_{>0}$ *implies* $\mathrm{Pr}_s^- \phi > 0$, $\;s \in S_d$ *implies* $\mathrm{Pr}_s^- \phi = 1$.

We still have to determine the value of $\mathrm{Pr}_s^- \phi$ for the states in $S_p' \stackrel{\mathrm{def}}{=} S_{>0} - S_d$. Each $s \in S_p'$ will choose the next-state distribution $p_i^s : 1 \leq i \leq k_s$ that minimizes the probability of getting to S_d. Thus, for all $s \in S_p'$ we have:

$$\mathrm{Pr}_s^- \phi = \min_{1 \leq i \leq k_s} \left[\sum_{t \in S_p'} p_i^s(t) \mathrm{Pr}_t^- \phi + \sum_{t \in S_d} p_i^s(t) \right] . \tag{11}$$

We can find a solution for the above set of equations by solving a linear programming problem, as the following lemma states.

Lemma 8. *To determine* $\Pr_s^- \phi$ *for all* $s \in S'_p$*, it suffices to find the set of values* $\{x_s : s \in S'_p\}$ *that maximizes* $\sum_{s \in S'_p} x_s$ *subject to the set of constraints*

$$x_s \leq \sum_{t \in S'_p} p_i^s(t) x_t + \sum_{t \in S_d} p_i^s(t)$$

for all $s \in S'_p$ *and* $1 \leq i \leq k_s$*. Then, it is simply* $\Pr_s^- \phi = x_s$*, for all* $s \in S'_p$*. These values are well-defined, as the above problem admits a unique optimal solution.*

To solve the above linear programming problem, it is possible to use well-known algorithms, such as the simplex method. To state the results about the complexity of pCTL model checking, assume that Π is described by listing all the next-state distributions for all states as vectors of rational numbers, each represented as the ratio of two integers. The *size of* Π, denoted by $|\Pi|$, will be simply the length of this description, considered as a string. Using algorithms based on the ellipsoid method, the above linear programming problem can be solved in polynomial time in $|\Pi|$ [21]. Therefore, we have the following theorem.

Theorem 9. *If the truth-values of* γ, ψ *are known at all* $s \in S$*, the truth-value of* $\mathbb{P}_{\leq a}(\gamma\, \mathcal{U}\, \psi)$ *at all* $s \in S$ *can be computed in polynomial time in* $|\Pi|$*.*

Computation of $\Pr_s^+ \phi$. In the case of $\Pr_s^+ \phi$, the set $S_{>0} = \{s \in S \mid \Pr_s^+ \phi > 0\}$ is simply the set of states of the directed graph $(S_d \cup S_p, \rho)$ from which it is possible to reach S_d following a path belonging to the graph itself. Again, $\Pr_s^+ \phi = 0$ for $s \in S - S_{>0}$, and $\Pr_s^+ \phi = 1$ for $S \in S_d$. Letting $S'_p \stackrel{\text{def}}{=} S_{>0} - S_d$, for all $s \in S'_p$ we can write, in analogy to (11),

$$\Pr_s^+ \phi = \max_{1 \leq i \leq k_s} \left[\sum_{t \in S'_p} p_i^s(t) \Pr_t^+ \phi + \sum_{t \in S_d} p_i^s(t) \right].$$

Again, we can compute $\Pr_s^+ \phi$ for all $s \in S'_p$ by solving a linear programming problem, and the analogous of Theorem 9 holds for $\mathbb{P}_{\geq a} \phi$.

Complexity of pCTL model checking. Combining the results about the complexity of CTL model checking [4] with Theorem 9, we get the following theorem about the complexity of pCTL model checking on PNS.

Theorem 10. *Model checking of pCTL formulas over a PNS* Π *can be done in time polynomial in* $|\Pi|$ *and linear in the size of the formula.*

4.2 pCTL* Formulas

We now turn to the problem of computing $\Pr_s^- \phi$ and $\Pr_s^+ \phi$ for a general pCTL* path formula $\phi \in \mathit{Seq}$. As $\Pr_s^- \phi = 1 - \Pr_s^+ \neg\phi$ by Lemma 6, we need to consider only the case of $\Pr_s^+ \phi$. As usual, we assume that the truth-values of all state subformulas of ϕ have already been evaluated at all states of the system.

The algorithm we propose consists of three steps. First, we put the formula ϕ in a canonical form ϕ''. Second, we construct from Π a new system Π', such that the states of Π' keep track of the truth-values of the subformulas of ϕ'', and the probability of sets of sequences in Π is equal to the probability of the corresponding sets of sequences in Π'. Third, we show that computing $\Pr_s^+ \phi$ in Π corresponds to computing the probability of reaching certain sets of states of Π', and this can be done using the method previously outlined for pCTL.[3]

Canonical form for ϕ. Let $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$ be the set of *maximal state subformulas* of ϕ, that is, the set of state subformulas of ϕ that are not proper subformulas of any other state subformula of ϕ. For each γ_i, we introduce a new propositional variable r_i, and let $\phi' = \phi[r_1/\gamma_1] \cdots [r_n/\gamma_n]$ be the result of replacing each occurrence of γ_i in ϕ with r_i, for all $1 \leq i \leq n$. As for each state $s \in S$ we have already computed whether $s \models \gamma_i$, we can extend the labeling V by letting $\tilde{V}(s) = V(s) \cup \{r_i \mid s \models \gamma_i,\ 1 \leq i \leq n\}$.

The resulting formula ϕ' is a *linear-time* temporal formula constructed with the propositional connectives and the temporal operators $\Box$, $\Diamond$, $\mathcal{U}$ on the propositional variables $r_1, \ldots, r_n$ [17]. By the results of [16, 3], $\neg\phi'$ can be put into the canonical form $\bigwedge_{i=1}^{l}(\Box\Diamond\chi_i \vee \Diamond\Box\lambda_i)$ for some *past* temporal formulas $\chi_1, \ldots, \chi_l$, $\lambda_1, \ldots, \lambda_l$ built with propositional connectives and the *past* temporal operators $\mathcal{S}$ (*since*) and $\ominus$ (*previous*) [15, 17]. Thus, ϕ' can be put into the form

$$\phi'' : \bigvee_{i=1}^{l} \Diamond\Box(\delta_i \wedge \Diamond\psi_i) \ ,$$

where again $\delta_1, \ldots, \delta_l, \psi_1, \ldots, \psi_l$ are past temporal formulas. Moreover, the size of ϕ'' is at most doubly exponential in the size of ϕ.

Construction of Π'. The truth-value of a past formula at point s_k of a sequence $s_0, s_1, s_2, \ldots$ depends only on the finite "past" sequence $s_0, s_1, \ldots, s_k$. Therefore, it is possible to construct from $\Pi = (S, s_{\rm in}, V, \tau)$ a system $\Pi' = (S', s'_{\rm in}, V', \tau')$ whose states keep track of the truth-values of the past formulas in ϕ'' as Π follows a sequence of states.

To do so, let $\theta_1, \ldots, \theta_m$ be the set of past subformulas of ϕ'' having $\mathcal{S}$ or $\ominus$ as the main connective, ordered in such a way that no θ_i is a subformula of θ_j for $i > j$. The space state of Π' is $S' = S \times \{\mathbf{true}, \mathbf{false}\}^m$, so that a state $s' = \langle s, u_1, \ldots, u_m\rangle \in S'$ consists of a state s of Π and of a sequence $u_1, \ldots, u_m$

[3] An alternative approach, not pursued in this paper, would have been to construct Π' from Π and from a deterministic Street automaton for $\neg\phi$.

of truth-values of $\theta_1, \ldots, \theta_m$. Any state in S' can be taken as the initial state $s'_{\rm in}$ of Π'. We define the projection function $\pi : S' \mapsto S$ by $\pi(\langle s, u_1, \ldots, u_m\rangle) = s$. Let $q_1, \ldots, q_m$ be new propositional variables, that will be used to replace the formulas $\theta_1, \ldots, \theta_m$. The labeling function V' is defined by

$$V'(\langle s, u_1, \ldots, u_m\rangle) = \tilde{V}(s) \cup \{q_i \mid u_i = \mathbf{true},\ 1 \le i \le m\} .$$

For $1 \le i \le m$, define $\hat{\theta}_i = \big(\ldots(\theta_i[q_{i-1}/\theta_{i-1}])\ldots[q_1/\theta_1]\big)$ to be the formulas resulting from successively substituting in θ_i $q_{i-1}, \ldots, q_1$ for $\theta_{i-1}, \ldots, \theta_1$. For $1 \le k \le l$, define

$$\hat{\delta}_k = \big(\ldots(\delta_k[q_m/\theta_m])\ldots[q_1/\theta_1]\big) \qquad \hat{\psi}_k = \big(\ldots(\psi_k[q_m/\theta_m])\ldots[q_1/\theta_1]\big)$$

to be the propositional formulas resulting from successively substituting $q_m, \ldots,$ q_1 for $\theta_m, \ldots, \theta_1$ in δ_k, ψ_k. Note that $\hat{\theta}_i$ does not contain any q_j for $1 \le i \le j \le m$. Define also $\hat{\phi}$ to be the formula obtained by orderly replacing each δ_i, ψ_i in ϕ'' with $\hat{\delta}_i$, $\hat{\psi}_i$, $1 \le i \le l$, respectively.

The definition of the reachability relation ρ' in Π' encodes the semantics of the past operators. Recall that a formula $\alpha\,\mathcal{S}\,\beta$ holds at a given state of a sequence if β holds at that state, or if α holds at that state and $\alpha\,\mathcal{S}\,\beta$ holds at the previous one; a formula $\ominus\alpha$ holds at a given state if α holds at the previous one. Consider any two states $s' = \langle s, u_1, \ldots, u_m\rangle$, $t' = \langle t, v_1, \ldots, v_m\rangle$ of Π'. As u_i, v_i represent the truth-values of $\hat{\theta}_i$ at s', t' respectively, we let t' be reachable from s', written $\rho'(s', t')$, if $\rho(s, t)$ and, for all $1 \le i \le m$:

1. if $\hat{\theta}_i$ has the form $\ominus\alpha$, $v_i = \mathbf{true}$ *iff* $s' \models \alpha$;
2. if $\hat{\theta}_i$ has the form $\alpha\,\mathcal{S}\,\beta$, $v_i = \mathbf{true}$ *iff* $[t' \models \beta$ *or* $(u_i = \mathbf{true}$ *and* $t' \models \alpha)]$.

The next-state probability distributions for Π' are then defined, for $s' \in S'$, $k_{s'} = k_{\pi(s')}$, and for $1 \le i \le k_{s'}$, by:

$$p_i^{s'}(t') = \begin{cases} p_i^{\pi(s')}(\pi(t')) & \text{if } \rho'(s', t'); \\ 0 & \text{otherwise.} \end{cases}$$

The fact that the above equation defines next-state probability distributions is a consequence of the following lemma.

Lemma 11. *Given $s \in S$ and $s' \in S'$ such that $s = \pi(s')$, for every $t \in S$ such that $\rho(s, t)$ there is exactly one $t' \in S'$ such that $t = \pi(t')$ and $\rho'(s', t')$.*

Proof. Let $t' = \langle r, v_1, \ldots, v_m\rangle$ be a state in S' such that $t = \pi(t')$ and $\rho'(s', t')$. The value of r is uniquely determined by $r = t$. For $1 \le i \le m$, the truth value of v_i is determined by s', t and by the truth values of $v_1, \ldots, v_{i-1}$. Hence, t' is uniquely determined. $\square$

Relationship between Π and Π'. A formula $\ominus\alpha$ is always false on the first state of a sequence. A formula $\alpha \,\mathcal{S}\, \beta$ holds on the first state of a sequence if that state satisfies β. Thus, in order for $u_1, \ldots, u_m$ to represent the truth-value of $\hat{\theta}_1, \ldots, \hat{\theta}_m$, a sequence in Π that starts at the state $s \in S$ should start in Π' at a state $\xi(s) = \langle s, u_1, \ldots, u_m \rangle$ such that, for all $1 \leq i \leq m$, u_i is true iff $\hat{\theta}_i$ has the form $\alpha \,\mathcal{S}\, \beta$ and $\xi(s) \models \beta$. As the above requirement uniquely determines $\xi(s)$, it defines a one-to-one function $\xi : S \mapsto S'$.

Moreover, for all $s \in S$, there is a bijective correspondence between the legal sequences of Π that start at $s \in S$ and those of Π' that start at $\xi(s)$. This correspondence relates each legal sequence $\omega : s_0, s_1, s_2, \ldots$ of Π with the unique legal sequence $\zeta(\omega) : s'_0, s'_1, s'_2, \ldots$ of Π' such that $\xi(s_0) = s'_0$, and $\pi(s'_i) = s_i$ for all $i > 0$. If $\Delta \in \Omega_s$ is a set of sequences of Π, denote with $\zeta(\Delta)$ the set of ζ-related sequences in Π'. The following lemma follows from the construction of Π' and $\hat{\phi}$.

Lemma 12. *$\omega \models \phi$ iff $\zeta(\omega) \models \hat{\phi}$, so that*

$$\zeta(\{\omega \in \Omega_s \mid \omega \models \phi\}) = \{\omega' \in \Omega'_{\xi(s)} \mid \omega' \models \hat{\phi}\} \ .$$

Proof. Given two corresponding sequences $\omega : s_0, s_1, s_2, \ldots$, $\zeta(\omega) : t_0, t_1, t_2, \ldots$ with labelings $\tilde{V}$, V' respectively, ϕ holds at s_0 iff ϕ'' holds at s_0. By induction on i it can be proved that θ_i holds at s_k iff $\hat{\theta}_i$ holds at t_k iff $u_i = \mathbf{true}$ at t_k, for $1 \leq i \leq m$, $k \geq 0$. Hence ϕ'' holds at s_0 iff $\hat{\phi}$ holds at t_0, and this concludes the proof. $\square$

Furthermore, there is a correspondence between the strategies of Π and Π'. To η for Π corresponds η' for Π' such that

$$Q_{\eta'}(i \mid s'_0 \ldots s'_n) = Q_\eta(i \mid \pi(s'_0) \ldots \pi(s'_n)) \ , \tag{12}$$

for all $n \geq 0$, all sequences $s'_0 \ldots s'_n$ of states of Π', and $1 \leq i \leq k_{s'}$. Related sets of sequences starting from related states of Π, Π' have thus the same probability, as the following lemma states.

Lemma 13. *If $\Delta \in \mathcal{B}_s$ is a measurable set and η, η' are related as in (12), $\mu_{s,\eta}(\Delta) = \mu_{\xi(s),\eta'}(\zeta(\Delta))$. Therefore, by definition of maximal measure,*

$$\mu^+_s(\Delta) = \mu^+_{\xi(s)}(\zeta(\Delta)) \ .$$

Proof. The result follows easily from the definition of next step probabilities in Π' and from the fact that ξ is one-to-one and ζ is bijective. $\square$

Computing in Π'. From the above relations, in order to compute $\mathrm{Pr}^+_s\,\phi$ in Π it suffices to compute $\mathrm{Pr}^+_{\xi(s)}\,\hat{\phi}$ in Π'; and to compute this we can take advantage of the special form of $\hat{\phi} : \bigvee_{i=1}^{l} \Diamond\Box(\hat{\delta}_i \wedge \Diamond\hat{\psi}_i)$.

For $1 \leq i \leq l$, define $C_i = \{s \in S' \mid s \models \hat{\delta}_i\}$, set $B_i := C_i$, and iterate the following three-step procedure until no more states can be removed from B_i.

1. Define, for each $s \in B_i$, the set of indices

$$M_s = \left\{ j \in \{1, \ldots, k_s\} \;\middle|\; \{t \in S' \mid p_j^s(t) > 0\} \subseteq B_i \right\}$$

of next-state distributions that do not lead any computation outside B_i.
2. Consider the directed graph $G = (B_i, E)$, where

$$E = \big\{ (s,t) \mid \exists j \in M_s \,.\, p_j^s(t) > 0 \big\} \,.$$

3. Remove from B_i all states s that cannot reach a state in $\{s \in B_i \mid s \models \hat{\psi}_i\}$ by a path in G of length at least 1.

Note that the above procedure is iterated $N_i \leq |S'|$ times.

For $1 \leq i \leq l$, let F_i be the subsets of B_i obtained, and let $F = \bigcup_{i=1}^{l} F_i$. For $A \subseteq S'$, $s \in S'$, define $\Gamma_s(A) = \{\omega \in \Omega'_s \mid \exists k \,.\, \omega|_k \in A\}$ to be the set of sequences that reach A from s. The following theorem allows us to compute $\Pr_s^+ \phi$.

Theorem 14. *For $s \in S$, $\Pr_s^+ \phi = \mu^+_{\xi(s)}\big(\Gamma_{\xi(s)}(F)\big)$.*

The quantity $\mu^+_{\xi(s)}(\Gamma_{\xi(s)}(F))$ can then be computed with the algorithm given in the previous section for pCTL, taking F as S_d and S' as S_p. The proof of the theorem uses the two following lemmas.

Lemma 15. *For all $s \in S'$, there is a strategy η such that a sequence $\omega \in \Gamma_s(F)$ satisfies $\hat{\phi}$ with probability 1, i.e.*

$$\mu_s^+ \left(\Gamma_s(F)\right) = \mu_s^+ \left(\{\omega \in \Gamma_s(F) \mid \omega \models \hat{\phi}\}\right) \,.$$

Moreover, this strategy does not need to depend on the portion of $\omega \in \Gamma_s(F)$ outside F.

Proof. Assume that $t_0 \in F$ is the first state at which $\omega \in \Gamma_s(F)$ enters F. Let $i = \min\big\{m \mid 1 \leq m \leq l \wedge t_0 \in F_m\big\}$. For $t \in F_i$, let

$$M_t = \left\{ j \in \{1, \ldots, k_t\} \;\middle|\; \{t' \in S' \mid p_j^t(t') > 0\} \subseteq F_i \right\}$$

be the set of indices of next-state distributions that do not leave F_i. The strategy η, at $t \in F_i$, will choose one of the $j \in M_t$ with equal probabilities. Note that while the strategy depends on the state t_0 of first entry in F, it does not depend on the portion of ω outside F. After the entry in F, the sequence is confined to F_i; from each $t \in F_i$ there is a path to a state of F_i where $\hat{\psi}_i$ holds; and F_i has finite size. Therefore, the sequence ω will satisfy $\Diamond(\Box\hat{\delta}_i \wedge \Box\Diamond\hat{\psi}_i)$, and $\hat{\phi}$, with probability 1. $\Box$

Lemma 16. *For $1 \leq i \leq l$, $s \in S'$, and for any strategy η, the measure of the set of sequences from s that satisfy $\Diamond(\Box\hat{\delta}_i \wedge \Box\Diamond\hat{\psi}_i)$ without ever entering F_i is 0, i.e.*

$$\mu_{s,\eta}\big(\{\omega \in \Omega'_s \mid \omega \models \Diamond(\Box\hat{\delta}_i \wedge \Box\Diamond\hat{\psi}_i)\} - \Gamma_s(F_i)\big) = 0 \,.$$

Proof. For $1 \leq j \leq N_i$, let D_j be the set of states that have been removed from B_i at the j-th iteration of the procedure; let also $D_0 = S' - C_i$ be the set of states that does not satisfy $\hat{\delta}_i$. Let $D_{<j} = \bigcup_{k=0}^{j-1} D_j$, $D_{>j} = \bigcup_{k=j+1}^{N_i} D_j$. Moreover, call a $\hat{\psi}_i$-state any state $t \in S'$ such that $t \models \hat{\psi}_i$. Define also

$$b = \inf\big\{p_m^s(t) \bigm| s, t \in S' \wedge 1 \leq m \leq k_s \wedge p_m^s(t) > 0\big\}$$

and note that $b > 0$, as S' has finite size. We will prove the following assertion by complete induction on j, from N_i down to 1:

> *For $1 \leq j \leq N_i$, a sequence passing from $s \in D_j$, never entering F_i and satisfying $\Box\Diamond\hat{\psi}_i$ will contain a state in $D_{<j}$ with probability 1.*

Clearly, this assertion implies the result stated by the lemma.

Consider the case of j, $1 \leq j \leq N_i$, and assume that the assertion has been proved for all j', $j < j' \leq N_i$.

Let a_1 be the fraction of sequences passing through $s \in D_j$ and reaching a $\hat{\psi}_i$-state without leaving $D_j \cup D_{>j}$. Since s has been removed from B_i, each of these sequences, before reaching the $\hat{\psi}_i$-state, must pass through a *critical point*, i.e. a point where the strategy η has chosen a next-state probability distribution p such that $\{t \in S' \mid p(t) > 0\} \not\subseteq D_j \cup D_{>j}$. Therefore, $a_1 \leq 1 - b$, as at most $1 - b$ sequences that pass through a critical point remain in $D_j \cup D_{>j}$.

The $\hat{\psi}_i$-state reached by the a_1 sequences belongs to either D_j or $D_{>j}$. If it belongs to D_j, we say that the first cycle is concluded. Otherwise, by the induction hypothesis we know that the sequences that pass through $D_{>j}$ and satisfy $\Box\Diamond\hat{\psi}_i$ without entering F_i eventually go to a state in $D_j \cup D_{<j}$ with probability 1. For these sequences, the first cycle is concluded when they reach $D_j \cup D_{<j}$. In either case, at most a_1 sequences complete the first cycle without leaving D_j or $D_{>j}$.

A fraction a_2 of the sequences that complete the first cycle without leaving $D_j \cup D_{>j}$ will reach another $\hat{\psi}_i$-state without leaving $D_j \cup D_{>j}$. As they must pass again through a critical point, $a_2 \leq 1 - b$. In general, the fraction of sequences that goes through k cycles without leaving $D_j \cup D_{>j}$ is at most $\prod_{m=1}^{k} a_m \leq (1-b)^k$. Therefore, the set of sequences passing through $s \in D_j$ that satisfy $\Box\Diamond\hat{\psi}_i$ without leaving $D_j \cup D_{>j}$ has measure 0. □

Corollary 17. *For any $s \in S'$ and η, $\mu_{s,\eta}\big(\{\omega \in \Omega'_s \mid \omega \models \hat{\phi}\} - \Gamma_s(F)\big) = 0$.*

Proof. From Lemma 16 we have

$$\begin{aligned} &\mu_{s,\eta}\big(\{\omega \in \Omega'_s \mid \omega \models \hat{\phi}\} - \Gamma_s(F)\big) \\ &\qquad \leq \sum_{i=1}^{l} \mu_{s,\eta}\big(\{\omega \in \Omega'_s \mid \omega \models \Diamond(\Box\hat{\delta}_i \wedge \Box\Diamond\hat{\psi}_i)\} - \Gamma_s(F_i)\big) = 0\,. \end{aligned}$$

□

Proof of Theorem 14. For $s' \in S'$, by the definition of maximal probability we have $\Pr^+_{s'}\hat{\phi} = \sup_\eta \mu_{s',\eta}(\{\omega \in \Omega'_{s'} \mid \omega \models \hat{\phi}\})$. By Corollary 17 we have, for any strategy η,

$$\begin{aligned}
&\mu_{s',\eta}(\{\omega \in \Omega'_{s'} \mid \omega \models \hat{\phi}\}) \\
&\qquad = \mu_{s',\eta}(\{\omega \in \Gamma_{s'}(F) \mid \omega \models \hat{\phi}\}) + \mu_{s',\eta}(\{\omega \in \Omega'_{s'} \mid \omega \models \hat{\phi}\} - \Gamma_{s'}(F)) \\
&\qquad = \mu_{s',\eta}(\{\omega \in \Gamma_{s'}(F) \mid \omega \models \hat{\phi}\}) \ .
\end{aligned}$$

Hence, by Lemma 15, $\Pr^+_{s'}\hat{\phi} = \sup_\eta \mu_{s',\eta}(\{\omega \in \Gamma_{s'}(F) \mid \omega \models \hat{\phi}\}) = \mu^+_{s'}(\Gamma_{s'}(F))$. From Lemmas 12 and 13 we finally have $\Pr^+_s \phi = \Pr^+_{\xi(s)}\hat{\phi} = \mu^+_{\xi(s)}(\Gamma_{\xi(s)}(F))$, as was to be proved. □

Complexity of pCTL* model checking. By combining results about the complexity of CTL* model checking [7], pCTL model checking, and an analysis of the above algorithm, we get the following result, that summarizes the complexity of pCTL* model checking for PNS.

Theorem 18. *Model checking of pCTL* formulas over a PNS Π can be done in polynomial time in $|\Pi|$.*

On the other hand, from the results of [5] we know that determining whether a linear-time temporal formula is satisfied with probability 1 by a PNS requires at least doubly exponential time in the size of the formula. As this problem can be reduced to pCTL* model checking, we have the following result.

Theorem 19. *Model checking of pCTL* formulas over PNS has a time complexity that is at least doubly exponential in the size of the formula.*

In the algorithm we presented, we can trace the source of this complexity to the step that computes the canonical form of a temporal formula, and to the construction of Π'. In fact, $|\Pi'|$ is triply-exponential in $|\phi|$, in the worst case.

Strategies for pCTL and pCTL*. We say that a strategy η is *deterministic* if $Q_\eta(i \mid s_0 \dots s_n)$ is either 0 or 1 for all $1 \le i \le k_{s_n}$, $n \ge 0$ and all sequences $s_0 \dots s_n$ of states of S. We say that a strategy is *Markovian* if

$$Q_\eta(i \mid s_0 \dots s_n) = Q_\eta(i \mid s_n)$$

for all $n \ge 0$ and all sequences $s_0 \dots s_n$ of states of S.

Given a system Π, and $\phi \in Seq$, $s \in S$, say that a strategy η is *most favorable* (resp. *most unfavorable*) if $\mu_{s,\eta}(\{\omega \in \Omega_s \mid \omega \models \phi\}) = \Pr^+_s \phi$ (resp. if $\mu_{s,\eta}(\{\omega \in \Omega_s \mid \omega \models \phi\}) = \Pr^-_s \phi$). The following corollary, derived from an analysis of the model-checking algorithms, gives us a characterization of the most favorable and unfavorable strategies corresponding to pCTL and pCTL* formulas.

Corollary 20. *The following results hold.*

1. *For all PNS Π and all pCTL formulas $\phi \in Seq$, there are Markovian and deterministic strategies that are most favorable and most unfavorable for all $s \in S$.*
2. *For all PNS Π, all pCTL* formulas $\phi \in Seq$ and all $s \in S$, there are most favorable and most unfavorable strategies that are deterministic. However, there are PNS Π, $s \in S$, and pCTL* formulas $\phi \in Seq$ such that there are no most favorable nor most unfavorable strategies that are Markovian.*

The second part of this corollary shows that nondeterminism cannot be encoded by leaving some transition probabilities of a Markov chain unspecified, if pCTL* is used as the specification language.

5 Conclusions

It is known from [10, 2] that pCTL and pCTL* model checking on Markov chains can be done in polynomial time in the size of the system. It is interesting to note that adding nondeterminism still preserves the polynomial time bound, provided the size of the system takes into account not only the number of states, but also the encoding of the transition probabilities.

The situation is different for the time bounds expressed in terms of the size of the formula. Model checking of pCTL formulas can be done in linear time on the size of the formula both for Markov chains [10] and PNS. However, while pCTL* model checking on Markov chains can be done in single exponential time in the size of the formula [5, 2], pCTL* model checking on PNS requires at least doubly exponential time in the size of the formula. In our algorithm, the complexity of putting formulas in canonical form is partially mitigated by the fact that many common formulas used in system specification can be efficiently put into canonical form.

Acknowledgements. We would like to thank Anca Browne, Anil Kamath, Zohar Manna, Serge Plotkin and Amir Pnueli for helpful comments and suggestions.

References

1. R. Alur, C. Courcoubetis, and D. Dill. Verifying automata specifications of probabilistic real-time systems. In *Real Time: Theory in Practice*, Lecture Notes in Computer Science 600, pages 28–44. Springer-Verlag, 1992.
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, 7th International Workshop*, volume 939 of *Lect. Notes in Comp. Sci.* Springer-Verlag, 1995.

3. E. Chang, Z. Manna, and A. Pnueli. The safety-progress classification. In *Logic, Algebra, and Computation*, NATO ASI Series, Subseries F: Computer and System Sciences. Springer-Verlag, Berlin, 1992.
4. E.M. Clarke, E.A. Emerson, and A.P. Sistla. Automatic verification of finite state concurrent systems using temporal logic. In *Proc. 10th ACM Symp. Princ. of Prog. Lang.*, 1983.
5. C. Courcoubetis and M. Yannakakis. Verifying temporal properties of finite-state probabilistic programs. In *Proc. 29th IEEE Symp. Found. of Comp. Sci.*, 1988.
6. E.A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume E, chapter 16, pages 995–1072. Elsevier Science Publishers (North-Holland), Amsterdam, 1990.
7. 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.
8. E.A. Emerson and A.P. Sistla. Deciding branching time logic. In *Proc. 16th ACM Symp. Theory of Comp.*, pages 14–24, 1984.
9. H. Hansson. *Time and Probabilities in Formal Design of Distributed Systems.* Real-Time Safety Critical Systems. Elsevier, 1994.
10. 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.
11. H. Hansson and B. Jonsson. A logic for reasoning about time and probability. *Formal Aspects of Computing*, 6(5):512–535, 1994.
12. S. Hart and M. Sharir. Probabilistic temporal logic for finite and bounded models. In *Proc. 16th ACM Symp. Theory of Comp.*, pages 1–13, 1984.
13. J.G. Kemeny, J.L. Snell, and A.W. Knapp. *Denumerable Markov Chains.* D. Van Nostrand Company, 1966.
14. D. Lehman and S. Shelah. Reasoning with time and chance. *Information and Control*, 53(3):165–198, 1982.
15. O. Lichtenstein, A. Pnueli, and L. Zuck. The glory of the past. In *Proc. Conf. Logics of Programs*, volume 193 of *Lect. Notes in Comp. Sci.*, pages 196–218. Springer-Verlag, 1985.
16. O. Maler and A. Pnueli. Tight bounds on the complexity of cascaded decomposition of automata. In *Proc. 31th IEEE Symp. Found. of Comp. Sci.*, pages 672–682, 1990.
17. Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Specification.* Springer-Verlag, New York, 1991.
18. A. Pnueli. On the extremely fair treatment of probabilistic algorithms. In *Proc. 15th ACM Symp. Theory of Comp.*, pages 278–290, 1983.
19. A. Pnueli and L. Zuck. Probabilistic verification by tableaux. In *Proc. First IEEE Symp. Logic in Comp. Sci.*, pages 322–331, 1986.
20. A. Pnueli and L.D. Zuck. Probabilistic verification. *Information and Computation*, 103:1–29, 1993.
21. A. Schrijver. *Theory of Linear and Integer Programming.* J. Wiley & Sons, 1987.
22. M.Y. Vardi. Automatic verification of probabilistic concurrent finite-state systems. In *Proc. 26th IEEE Symp. Found. of Comp. Sci.*, pages 327–338, 1985.