Model Checking Discounted Temporal Properties

Luca de Alfaro[1], Marco Faella[1,3], Thomas A. Henzinger[2], Rupak Majumdar[2], and Mariëlle Stoelinga[1]

[1] CE, Universitity of California, Santa Cruz, USA
[2] EECS, University of California, Berkeley, USA
[3] Università degli Studi di Salerno, Italy

University of California, Santa Cruz
Technical Report UCSC-CRL-03-12

October 29, 2003

Abstract. Temporal logic is two-valued: a property is either true or false. When applied to the analysis of stochastic systems, or of systems with imprecise formal models, temporal logic is therefore fragile: even small changes in the model can lead to opposite truth values for the specification. We present a generalization of the branching-time logic CTL that achieves robustness with respect to model perturbations by giving a quantitative interpretation to predicates and logical operators, and by discounting the importance of events according to how late they occur.

In every state, the value of a formula is a real number in the interval [0,1], where 1 corresponds to truth and 0 to falsehood. The boolean operators and and or are replaced by min and max, the path quantifiers E and A determine sup and inf over all paths from a given state, and the temporal operators F and G specify sup and inf over a given path; a new operator averages all values along a path. Furthermore, all path operators are discounted by a parameter that can be chosen to give more weight to states that are closer to the beginning of the path.

We interpret the resulting logic DCTL over transition systems, Markov chains, and Markov decision processes. We provide examples and robustness theorems that demonstrate the usefulness of DCTL for specifying performance properties of systems. We also present model-checking algorithms, and we show that over probabilistic systems the logic cannot be model-checked via the usual connection to the mu-calculus.

1 Introduction

Boolean state-transition models are useful for the representation and verification of computational systems, such as hardware and software systems. A boolean state-transition model is a labeled directed graph, whose vertices represent system states, whose edges represent state changes, and whose labels represent boolean observations about the system, such as the truth values of state predicates. Behavioral properties of boolean state-transition systems can be specified in temporal logic [CGP99,MP91] and verified using model-checking algorithms [CGP99].

For representing systems that are not purely computational but partly physical, such as hardware and software that interacts with a physical environment, boolean state-transition models are often inadequate. Many quantitative extensions of state-transition models have been proposed for this purpose, such as models that embed state changes into the real time line [AD94] and models that assign probabilities to state changes. These models typically contain real numbers, e.g., for representing time or probabilities. Yet previous research has focused mostly on purely boolean frameworks for the specification and verification of quantitative state-transition models, where observations are truth values of state predicates, and behavioral properties are based on such boolean observations [Han94,BdA95,BHHK00,Kwi03]. These boolean specification frameworks are *fragile* with respect to imprecisions in the model: even arbitrarily small changes in the quantitative models can cause different truth values for the specification.

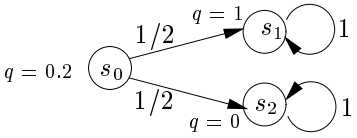
We submit that a proper framework for the specification and verification of quantitative state-transition models should itself be quantitative. To start with, we consider observations that do not have boolean truth values, but real values [Koz83]. Using these quantitative observations, we build a temporal logic for specifying quantitative temporal properties. A CTL-like temporal logic has three kinds of operators. The first kind are boolean operators such as "and" and "or" for locally combining the truth values of boolean observations. These are replaced by "min" and "max" operators for combining the real values of quantitative observations. In addition, a binary "average" operator is useful to generate new quantitative observations. The second kind of constructs are modal operators "always" ($\Box$) and "eventually" ($\Diamond$) for temporally combining the truth values of all boolean observations along an infinite path. These are replaced by "inf" ("lim min") and "sup" ("lim max") operators over infinite sequences of real values. We introduce a

"lim avg" ($\triangle$) operator that captures the long-run average value of a quantitative observation. For nondeterministic models, where there is a choice of future behaviors, there is a third kind of construct: the path quantifiers "for-all-possible-futures" ($\forall$) and "for-some-possible-future" ($\exists$) turn path properties into state properties by quantifying over the paths from a given state. These are replaced by "inf-over-all-possible-futures" and "sup-over-all-possible-futures." Once boolean specifications are replaced by quantitative specifications, it becomes possible to discount the future, that is, to give more weight to the near future than to the far away future. This principle is well-understood in economics and in the theory of optimal control [Ber95], but equally natural in studying quantitative temporal properties of systems. We call the resulting logic DCTL ("Discounted CTL"). While quantitative versions of dynamic logics [Koz83], μ-calculi [HK97,McI98,MM02,dAHM03] and local Hennessy-Milner logics [DEP02] exist, DCTL is the first temporal logic in which the basic operators (the temporal operators $\Diamond$ and $\Box$, along with the new temporal operator $\triangle$, and the path quantifiers $\forall$ and $\exists$) are given a quantitative interpretation.

We propose two semantics for DCTL: a *path* semantics, and a *fixpoint* semantics. In the (undiscounted) path semantics, the $\Diamond$ (resp. $\Box$) operator computes the sup (resp. inf) over a path, and the $\triangle$ operator computes the long-run average. The discounted versions $\Diamond_\alpha$, $\Box_\alpha$, and $\triangle_\alpha$ of these operators weigh the value of a state that occurs k steps in the future by a factor α^k, where $\alpha \leq 1$ is the discount factor. The $\forall$ and $\exists$ operators then combine these values over the paths: in transition systems, $\forall$ and $\exists$ simply associate with each state the inf and sup of the values of the paths leaving a state; in probabilistic systems, $\forall$ and $\exists$ associate with each state the least and greatest expected value of a the value over a path, respectively. Thus, the path semantics of DCTL is obtained by lifting to a quantitative setting the classical interpretation of path and state formulas in CTL.

The *fixpoint* semantics is obtained by lifting to a quantitative setting the connection between CTL and μ-calculus [EL86,CGP99]. In a transition system, given a set r of states, denote by $\exists\mathrm{Pre}(r)$ the set of all states that have a one-step transition to r. Then, the semantics of $\exists\Diamond r$ for a set of states r can be defined as the least fixpoint of the equality $x = r \cup \exists\mathrm{Pre}(x)$, (denoted $\mu x.(r \cup \exists\mathrm{Pre}(x))$). We can lift this definition to a quantitative setting, by interpreting $\cup$ as pointwise maximum, and $\exists\mathrm{Pre}(x)$ as the maximal expected value of x achievable in one step [dAHM03]. The discounted semantics $\exists\Diamond_\alpha r$

is obtained simply by multiplying the next-step expectation by α: $\mu x.(r \cup \alpha \cdot \exists\mathrm{Pre}(x))$.



The path and fixpoint semantics coincide on transition systems, but differ on Markov chains (and consequently on Markov decision processes). This is illustrated by the Markov chain depicted at right. Consider the DCTL formula $\phi : \exists\Diamond_\alpha q$, for $\alpha = 0.8$. According to the path semantics, there are two paths from s_0, each followed with probability $1/2$: the first path has discounted sup equal to 0.8, and the second has discounted sup equal to 0.2; hence, ϕ has value $(0.8+0.2)/2 = 0.5$ at s_0. According to the fixpoint semantics, $[q] \cup 0.8 \cdot \exists\mathrm{Pre}(q)$ has value $\max\{0.2, 0.8 \cdot (1+0)/2)\} = 0.4$ at s_0, and this is also the value of ϕ at s_0. This example highlights the different perspective taken by the two semantics. The path semantics of $\exists\Diamond q$ is an "observational" semantics: if q represents, for instance, the level of water in a vessel (0 is empty, 1 is full), then $\exists\Diamond q$ is the expected value of the maximum level that occurs along a system behavior. Such semantics is well suited to system specification. The fixpoint semantics of $\exists\Diamond q$ is a "controlling" semantics: if q represents the retirement bonus that we receive if we decide to retire, then $\exists\Diamond q$ is the maximal expected bonus we will receive (discounting accounts for inflation). The difference is that in the fixpoint semantics we must decide when to stop: the choice of retiring, or working for one more day, corresponds to the choice between the two sides q and $\exists\mathrm{Pre}(x)$ of the the $\cup$ operator (interpreted as pointwise maximum) in the fixpoint. In the path semantics, on the other hand, we have no control over stopping: we can only observe the value of q over infinite runs, and compute the expected value of the sup it reaches. The fixpoint semantics is better suited to system control: if the goal is to reach a state with high value of q, we must not only reach such a state, but also be able to stop, once satisfied by a sufficiently high value of q, and move on to some subsequent control goal.

In DCTL, discounting serves two purposes. First, it leads to a notion of "quality" with which a specification is satisfied. For example, if we wish to reach a state with high value of q, then the undiscounted formula $\exists\Diamond q$ is valid regardless on when a high value of q is reached, whereas $\exists\Diamond_\alpha q$, for $\alpha < 1$, prizes the case where the high q is reached earlier. Likewise, if q represents the "level of functionality" of a system, then the specification $\forall\Box_\alpha q$ will have a value that is higher, the longer the system functions well, even if the

system will eventually always break. Second, discounting is instrumental in achieving robustness with respect to system perturbations. Indeed, we will show that for discount factors smaller than 1, the value of DCTL formulas in both semantics is a continuous function of the values of the numerical quantities (observations, transition probabilities) of the model.

We present algorithms for model checking both semantics of DCTL over transition systems, Markov chains, and Markov decision processes. Note that DCTL is a quantitative logic even when interpreted over purely boolean state-transition systems, for discount factors less than 1. Over transition systems, the algorithms for $\Box_\alpha$ and $\Diamond_\alpha$ are based on iterating quantitative fixpoint expressions; the main result in this regard is that the iteration always terminates within a finite number of steps which is bounded by the diameter of the system. The algorithm for $\triangle_\alpha$ (discounted long-run average along a path) is more involved, but still polynomial: it builds on both Karp's algorithm for computing minimum mean-weight cycles and a discounted version of Bellman-Ford for computing shortest paths.

For Markov chains and Markov processes, we can model check the fixpoint semantics of DCTL by relying on a mix of results from optimal control [Ber95] and quantitative μ-calculus [dAHM03]. On the other hand, model checking the path semantics of DCTL over Markov chains and Markov decision processes requires novel algorithms. Indeed, in spite of the fact that MDPs have been heavily studied, no algorithms for solving this natural problem — compute the maximal expectation of the sup along a path — were previously known, neither in the discounted, nor in the undiscounted, setting.

In all cases, we show that the model checking problem for DCTL can be solved in time polynomial in the size of the system. For transition systems and Markov chains, the time required is also polynomial in the size of the DCTL formula. For Markov decision processes, the time required is instead exponential in the depth of the DCTL formula, as the bit-wise encoding of valuations is subject to growing at each nesting of the temporal operators.[1]

[1] In practice, unless the algorithms are implemented with arbitrary-precision arithmetic, the time for DCTL model checking over Markov decision processes is polynomial in the size of the DCTL formula.

2 Discounted CTL

2.1 Syntax

Let Σ be a set of propositions and let A be a set of parameters. The DCTL formulas over (Σ, A) are generated by the grammar

$$\begin{array}{rcl} \phi & ::= & r \mid \mathrm{T} \mid \mathrm{F} \mid \phi \vee \phi \mid \phi \wedge \phi \mid \neg\phi \mid \phi \oplus_c \phi \mid \exists\psi \mid \forall\psi \\ \psi & ::= & \Diamond_c \phi \mid \Box_c \phi \mid \triangle_c \phi \end{array}$$

where $r \in \Sigma$ is a proposition and $c \in A$ is a parameter. The formulas generated by ϕ are *state formulas*; the formulas generated by ψ are *path formulas*. The DCTL formulas are the state formulas. We say that the formula ϕ is a *basic formula* if every non-trivial subformula of ϕ is a proposition.

2.2 Semantics for Labeled Transition Systems

We define two semantics for DCTL: the path semantics, and the fixpoint semantics. In the path semantics, the path operators $\Diamond$ and $\Box$ determine the discounted sup and inf values over a path, and the $\exists$ and $\forall$ operators determine the minimum and maximum values of the path formula over all paths from a given state. The fixpoint semantics is defined by lifting to a quantitative setting the usual connection between CTL and μ-calculus.

Discount factors. Let A be a set of parameters. A *parameter interpretation* of A is a function $\langle\cdot\rangle\colon A \to [0,1]$ which assigns to each parameter a real between 0 and 1. If $0 < \langle c\rangle < 1$, then $\langle c\rangle$ is called a *discount factor*. The interpretation $\langle\cdot\rangle$ is *contractive* if $\langle c\rangle < 1$ for all $c \in A$; it is *undiscounted* if $\langle c\rangle = 1$ for all $c \in A$. We write $\mathcal{I}_A$ for the set of parameter interpretations of A. We denote by $|q|_b$ the length of the binary encoding of a number $q \in \mathbb{Q}$, and we denote by $|\langle\cdot\rangle|_b = \sum_{a\in A} |\langle a\rangle|_b$ the size of the interpretation $\langle\cdot\rangle$ of A.

Valuations. Let S be a set of states. A *valuation* on S is a function $v\colon S \to [0,1]$ which assigns to each state a real between 0 and 1. The valuation v is *boolean* if $v(s) \in \{0,1\}$ for all $s \in S$. We write $\mathcal{V}_S$ for the set of valuations on S. We write $\mathbf{0}$ for the valuation that maps all states to 0, and $\mathbf{1}$ for the valuation that maps all states to 1. For two real numbers u_1, u_2 and a discount factor $\alpha \in [0,1]$ we write $u_1 \sqcup u_2$ for $\max\{u_1, u_2\}$, $u_1 \sqcap u_2$ for $\min\{u_1, u_2\}$, and $u_1 +_\alpha u_2$ for $(1-\alpha)\cdot u_1 + \alpha\cdot u_2$. We lift operations on reals to operations on valuations in a pointwise fashion; for example, for two valuations $v_1, v_2 \in \mathcal{V}_S$, by $v_1 \sqcup v_2$ we denote the valuation that maps each state $s \in S$ to $v_1(s) \sqcup v_2(s)$.

Labeled transition systems. A *labeled transition system* (LTS) $\mathcal{S} = (S, \delta, \Sigma, [\cdot])$ consists of a set S of states, a transition relation $\delta\colon S \to 2^S \setminus \emptyset$ which assigns to each state a nonempty set of successor states, a set Σ of propositions, and a function $[\cdot]\colon \Sigma \to \mathcal{V}_S$ which assigns to each proposition a valuation. We denote by $|\delta|$ the value $\sum_{s \in S} |\delta(s)|$. The labeled transition system $\mathcal{S}$ is *boolean* if for all propositions $r \in \Sigma$, the valuation $[r]$ is boolean. A *path* of $\mathcal{S}$ is an infinite sequence $s_0 s_1 s_2 \ldots$ of states such that $s_{i+1} \in \delta(s_i)$ for all $i \geq 0$. Given a state $s \in S$, we write $\mathit{Traj}(s)$ for the set of paths that start in s.

The path semantics. The DCTL formulas over (Σ, A) are evaluated with respect to a labeled transition system $\mathcal{S} = (S, \delta, \Sigma, [\cdot])$ whose propositions are Σ, and with respect to a parameter interpretation $\langle\cdot\rangle \in \mathcal{I}_A$. Every state formula ϕ define a valuation $[\![\phi]\!]^{\mathrm{p}} \in \mathcal{V}_S$:

$$
\begin{array}{lcl}
[\![r]\!]^{\mathrm{p}} & = & [r] \\
[\![\mathrm{T}]\!]^{\mathrm{p}} & = & \mathbf{1} \\
[\![\mathrm{F}]\!]^{\mathrm{p}} & = & \mathbf{0} \\
[\![\neg\phi]\!]^{\mathrm{p}} & = & \mathbf{1} - [\![\phi]\!]^{\mathrm{p}}
\end{array}
\qquad\qquad
\begin{array}{lcl}
[\![\phi_1 \vee \phi_2]\!]^{\mathrm{p}} & = & [\![\phi_1]\!]^{\mathrm{p}} \sqcup [\![\phi_2]\!]^{\mathrm{p}} \\
[\![\phi_1 \wedge \phi_2]\!]^{\mathrm{p}} & = & [\![\phi_1]\!]^{\mathrm{p}} \sqcap [\![\phi_2]\!]^{\mathrm{p}} \\
[\![\phi_1 \oplus_c \phi_2]\!]^{\mathrm{p}} & = & [\![\phi_1]\!]^{\mathrm{p}} +_{\langle c\rangle} [\![\phi_2]\!]^{\mathrm{p}} \\
[\![\exists\psi]\!]^{\mathrm{p}}(s) & = & \sup\{[\![\psi]\!]^{\mathrm{p}}(\rho) \mid \rho \in \mathit{Traj}(s)\} \\
[\![\forall\psi]\!]^{\mathrm{p}}(s) & = & \inf\{[\![\psi]\!]^{\mathrm{p}}(\rho) \mid \rho \in \mathit{Traj}(s)\}
\end{array}
$$

where $r \in \Sigma$. Every path formula ψ assigns a real $[\![\psi]\!]^{\mathrm{p}}(\rho) \in [0, 1]$ to each path ρ of $\mathcal{S}$:

$$
\begin{array}{lcl}
[\![\Diamond_c\phi]\!]^{\mathrm{p}}(s_0 s_1 \ldots) & = & \sup\{\langle c\rangle^i \cdot [\![\phi]\!]^{\mathrm{p}}(s_i) \mid i \geq 0\} \\
[\![\Box_c\phi]\!]^{\mathrm{p}}(s_0 s_1 \ldots) & = & \inf\{1 - \langle c\rangle^i \cdot (1 - [\![\phi]\!]^{\mathrm{p}}(s_i)) \mid i \geq 0\} \\
[\![\triangle_c\phi]\!]^{\mathrm{p}}(s_0 s_1 \ldots) & = & \begin{cases} (1 - \langle c\rangle) \cdot \sum\{\langle c\rangle^i \cdot [\![\phi]\!]^{\mathrm{p}}(s_i) \mid i \geq 0\} & \text{if } \langle c\rangle < 1, \\ \lim_{i \geq 0}(\frac{1}{i+1} \cdot \sum_{0 \leq j \leq i} [\![\phi]\!]^{\mathrm{p}}(s_j)) & \text{if } \langle c\rangle = 1. \end{cases}
\end{array}
$$

Notice that the limit of the first clause for $\triangle_c$ when $\langle c\rangle \to 1$ gives the second clause. If the labeled transition system $\mathcal{S}$ is boolean and the parameter interpretation $\langle\cdot\rangle$ is undiscounted, then 1 can be interpreted as truth, 0 as falsehood, and DCTL without the operator $\triangle$ coincides with CTL.

The fixpoint semantics. In this semantics, the DCTL formulas are evaluated with respect to a labeled transition system and to a contractive parameter interpretation $\langle\cdot\rangle \in \mathcal{I}_A$. Given a valuation $x \in \mathcal{V}_S$, we denote by $\exists\mathrm{Pre}(x) \in \mathcal{V}_S$ the valuation defined by $\exists\mathrm{Pre}(x)(s) = \max\{x(t) \mid t \in \delta(s)\}$, and we denote by $\forall\mathrm{Pre}(x) \in \mathcal{V}_S$ the valuation defined by $\forall\mathrm{Pre}(x)(s) =$

$\min\{x(t) \mid t \in \delta(s)\}$. The fixpoint semantics $[\![\cdot]\!]^{\mathrm{f}}$ for the propositions, the constants $\mathtt{T}$ and $\mathtt{F}$ and the boolean operators is similar to the path semantics, where $[\![\cdot]\!]^{\mathrm{p}}$ is substituted by $[\![\cdot]\!]^{\mathrm{f}}$. The other operators are defined as follows:

$$\begin{array}{rcl}
[\![\exists\Diamond_c\phi]\!]^{\mathrm{f}} & = & \mu x.([\![\phi]\!]^{\mathrm{f}} \sqcup (\mathbf{0} +_{\langle c\rangle} \exists\mathrm{Pre}(x))) \\
[\![\forall\Diamond_c\phi]\!]^{\mathrm{f}} & = & \mu x.([\![\phi]\!]^{\mathrm{f}} \sqcup (\mathbf{0} +_{\langle c\rangle} \forall\mathrm{Pre}(x))) \\
[\![\exists\Box_c\phi]\!]^{\mathrm{f}} & = & \mu x.([\![\phi]\!]^{\mathrm{f}} \sqcap (\mathbf{1} +_{\langle c\rangle} \exists\mathrm{Pre}(x))) \\
[\![\forall\Box_c\phi]\!]^{\mathrm{f}} & = & \mu x.([\![\phi]\!]^{\mathrm{f}} \sqcap (\mathbf{1} +_{\langle c\rangle} \forall\mathrm{Pre}(x))) \\
[\![\exists\triangle_c\phi]\!]^{\mathrm{f}} & = & \mu x.([\![\phi]\!]^{\mathrm{f}} +_{\langle c\rangle} \exists\mathrm{Pre}(x))) \\
[\![\forall\triangle_c\phi]\!]^{\mathrm{f}} & = & \mu x.([\![\phi]\!]^{\mathrm{f}} +_{\langle c\rangle} \forall\mathrm{Pre}(x)))
\end{array}$$

Above, for $F : \mathcal{V}_S \to \mathcal{V}_S$, the notation $\mu x.F(x)$ indicates the unique (as $\langle c\rangle < 1$) valuation x_* such that $x_* = F(x_*)$.

2.3 Semantics for Markov Processes

Given a finite set S, let $\mathrm{Distr}(S)$ be the set of probability distributions over S; for $a \in \mathrm{Distr}(S)$, we denote by $\mathrm{Supp}(a) = \{s \in S \mid a(s) > 0\}$ the support of a. A probability distribution a over S is *deterministic* if $a(s) \in \{0, 1\}$ for all $s \in S$.

Markov decision processes. A *Markov decision process* (MDP) $\mathcal{S} = (S, \tau, \Sigma, [\cdot])$ consists of a set S of states, a probabilistic transition relation $\tau\colon S \to 2^{\mathrm{Distr}(S)} \setminus \emptyset$, which assigns to each state a finite nonempty set of probability distributions over the successor states, a set Σ of propositions, and a function $[\cdot]\colon \Sigma \to \mathcal{V}_S$ which assigns to each proposition a valuation. The Markov decision process $\mathcal{S}$ is *boolean* if for all propositions $r \in \Sigma$, the valuation $[r]$ is boolean. A finite (resp. infinite) *path* of $\mathcal{S}$ is a finite (resp. infinite) sequence $s_0 s_1 s_2 \ldots s_m$ (resp. $s_0 s_1 s_2 \ldots$) of states such that for all $i < m$ (resp. $i \in \mathbb{N}$) there is $a_i \in \tau(s_i)$ with $s_{i+1} \in \mathrm{Supp}(a_i)$. We denote by *FTraj* and *Traj* the sets of finite and infinite paths of $\mathcal{S}$; for $s \in S$, we denote by Traj_s the infinite paths starting from s.

We denote by $|\tau|_b$ the length of the binary encoding of τ, defined by $\sum_{s\in S}\sum_{a\in\tau(s)}\sum_{t\in\mathrm{Supp}(a)}|a(t)|_b$, and we denote by $|[\cdot]|_b = \sum_{q\in\Sigma}\sum_{s\in S}|[q](s)|_b$ the size of the binary encoding of $[\cdot]$. Then, the binary size of $\mathcal{S}$ is given by $|\mathcal{S}|_b = |\tau|_b + |[\cdot]|_b$.

A *strategy* π for $\mathcal{S}$ is a mapping $\mathit{FTraj} \to \mathrm{Distr}(\bigcup_{s\in S}\tau(s))$: once the MDP has followed the path $s_0 s_1 \ldots s_m \in \mathit{FTraj}$, the strategy π prescribes the probability $\pi(s_0 s_1 \ldots s_m)(a)$ of using a next-state distribution

$a \in \tau(s_m)$. For all $s_0 s_1 \ldots s_m \in \mathit{FTraj}$ and all $a \in \mathrm{Distr}(S)$, we require that $\mathrm{Supp}(\pi(s_0 s_1 \ldots s_m)) \subseteq \tau(s_m)$. Thus, under strategy π, after following a finite path $s_0 s_1 \ldots s_m$ the MDP takes a transition to state s_{m+1} with probability $\sum_{a \in \tau(s_m)} a(s_{m+1})\, \pi(s_0 s_1 \ldots s_m)(a)$. We denote by Π the set of all strategies for $\mathcal{S}$. The transition probabilities corresponding to strategy π, together with an initial state s, give rise to a probability space $(\mathit{Traj}_s, \mathcal{B}_s, \Pr_s^\pi)$, where $\mathcal{B}_s$ is the set of measurable subsets of 2^{Traj_s}, and $\Pr_s^\pi$ is the probability measure over $\mathcal{B}_s$ induced by the next-state transition probabilities described above [KSK66,Wil91]. For $i \in \mathbb{N}$, the random variable $Z_i : \mathit{Traj}_s \to S$ defined by $Z_i(s_0 s_1 \ldots) = s_i$ yields state of the stochastic process after i steps. Given a random variable X over this probability space, we denote its expected value by $\mathrm{E}_s^\pi[X]$.

Special cases of MDPs: Markov chains and transition systems. Markov chains and transition systems can be defined as special cases of Markov decision processes. An MDP $\mathcal{S} = (S, \tau, \Sigma, [\cdot])$ is a *Markov chain* if $|\tau(s)| = 1$ for all $s \in S$. It is customary to specify the probabilistic structure of a Markov chain via its *probability transition matrix* $P = [p_{s,t}]_{s,t \in S}$, defined for all $s, t \in S$ by $p_{s,t} = a(t)$, where a is the unique distribution $a \in \tau(s)$. An initial state $s \in S$ completely determines a probability space $(\mathit{Traj}_s, \mathcal{B}_s, \Pr_s)$, and for a random variable X over this probability space, we let $\mathrm{E}_s[X]$ denote its expectation. An MDP $\mathcal{S} = (S, \tau, \Sigma, [\cdot])$ is a transition system if, for all $s \in S$ and all $a \in \tau(s)$, the distribution a is deterministic; in that case, we define $\delta : S \mapsto 2^S$ by $\delta(s) = \{t \in S \mid \exists a \in \tau(s).a(t) = 1\}$ for all $s \in S$.

The path semantics. The DCTL formulas over (Σ, A) are evaluated with respect to a Markov decision process $\mathcal{S} = (S, \tau, \Sigma, [\cdot])$ and with respect to a parameter interpretation $\langle\cdot\rangle \in \mathcal{I}_A$. The semantics $[\![\psi]\!]^{\mathrm{p}}$ of a path formula ψ is defined as for transition systems; we note that $[\![\psi]\!]^{\mathrm{p}}$ is a random variable over the probability space $(\mathit{Traj}_s, \mathcal{B}_s, \Pr_s)$. Every state formula ϕ defines a valuation $[\![\phi]\!]^{\mathrm{p}} \in \mathcal{V}_S$: the cases for propositions, T, F, $\vee$, $\wedge$, and $\neg$ are as for transition systems; the case for $\exists$ and $\forall$ is as follows:

$$[\![\exists\psi]\!]^{\mathrm{p}}(s) = \sup\{\mathrm{E}_s^\pi([\![\psi]\!]^{\mathrm{p}}) \mid \pi \in \Pi\}, \qquad [\![\forall\psi]\!]^{\mathrm{p}}(s) = \inf\{\mathrm{E}_s^\pi([\![\psi]\!]^{\mathrm{p}}) \mid \pi \in \Pi\}.$$

The fixpoint semantics. Given a valuation $x : S \to [0,1]$, we denote by $\exists\mathrm{Pre}(x) : S \to [0,1]$ the valuation defined by $\exists\mathrm{Pre}(x)(s) = \max_{a \in \tau(s)} \sum_{t \in S} x(t)a(t)$, and we denote by $\forall\mathrm{Pre}(x) : S \to [0,1]$ the valuation defined by $\forall\mathrm{Pre}(x)(s) = \min_{a \in \tau(s)} \sum_{t \in S} x(t)a(t)$. With this notation,

the fixpoint semantics $[\![\cdot]\!]^{\mathrm{f}}$ is defined by the same clauses as for transition systems.

2.4 Properties of DCTL

Basic equivalences. For all state formulas ϕ_1, ϕ_2 over (Σ, A), all MDPs with propositions Σ, and all contractive parameter interpretations of A and $* \in \{\mathrm{p}, \mathrm{f}\}$, we have the following equivalences: $[\![\neg\exists\Diamond_c\phi]\!]^* = [\![\forall\Box_c\neg\phi]\!]^*$, $[\![\neg\exists\Box_c\phi]\!]^* = [\![\forall\Diamond_c\neg\phi]\!]^*$. and $[\![\neg\exists\triangle_c\phi]\!]^* = [\![\forall\triangle_c\neg\phi]\!]^*$ In particular, we see that $\triangle_c$ is self-dual and that a minimalist definition of DCTL will omit one of $\{\mathrm{T}, \mathrm{F}\}$, one of $\{\vee, \wedge\}$, and one of $\{\exists, \forall, \Diamond, \Box\}$.

Comparing both semantics. We show that the path and fixpoint semantics coincide over transition systems, and over Markov systems with boolean propositions (for non-nested formulas), but do not coincide in general over (non-boolean) Markov chains. This result is surprising, as it indicates that the standard connection between CTL and μ-calculus breaks down as soon as we consider *both* probabilistic systems and quantitative valuations. Since discounting plays no role in the proof of the theorem, so that a similar result would hold also for the logic without operator $\triangle$ under no discounting. On the other hand, the theorem states that the two semantics always coincide for the $\triangle_c$ operator.

Theorem 1. *The following assertions hold:*

1. *For all labeled transition systems with propositions Σ, all contractive parameter interpretations of A, and all* DCTL *formulas ϕ over (Σ, A), we have $[\![\phi]\!]^{\mathrm{p}} = [\![\phi]\!]^{\mathrm{f}}$.*
2. *For all boolean Markov decision processes with propositions Σ, all contractive parameter interpretations of A, and all* DCTL *formulas ϕ over (Σ, A) that contain no nesting of path quantifiers, we have $[\![\phi]\!]^{\mathrm{p}} = [\![\phi]\!]^{\mathrm{f}}$.*
3. *There is a Markov chain $\mathcal{S}$ with propositions Σ, a contractive parameter interpretation A, and a* DCTL *formula ϕ over (Σ, A) such that $[\![\phi]\!]^{\mathrm{p}} \neq [\![\phi]\!]^{\mathrm{f}}$.*

Lemma 1. *For all MDPs with propositions Σ, all contractive parameter interpretations of A, and all $r \in \Sigma$, we have $[\![\exists\triangle_c r]\!]^{\mathrm{p}} = [\![\exists\triangle_c r]\!]^{\mathrm{f}}$ and $[\![\forall\triangle_c r]\!]^{\mathrm{p}} = [\![\forall\triangle_c r]\!]^{\mathrm{f}}$.*

Robustness. Let $\mathcal{S} = (S, \tau, \Sigma, [\cdot])$ and $\mathcal{S}' = (S, \tau', \Sigma, [\cdot]')$ be two MDPs on the same state space S and set of atomic propositions Σ. $\|\mathcal{S}, \mathcal{S}'\| = \max_{s \in S}\{\max_{r \in \Sigma} |[r](s) - [r'](s)|, \max_{a \in \tau(s)} \min_{b \in \tau'(s)} \sum_{s' \in S} |a(s') - b(s')|, \max_{b \in \tau'(s)} \min_{a \in \tau(s)} \sum_{s' \in S} |a(s') - b(s')|\}$. It is not difficult to see that $\|, \|$ is a metric. For an MDP $\mathcal{S}$, we write $[\![\cdot]\!]^{\mathrm{f}}_{\mathcal{S}}$ and $[\![\cdot]\!]^{\mathrm{p}}_{\mathcal{S}}$ to denote the semantics functions defined on $\mathcal{S}$.

Theorem 2. *Let $\langle\cdot\rangle$ be a contractive parameter evaluation.*

1. *For all $\epsilon > 0$, there is a $\delta > 0$ such that for all formulas φ of DCTL and all states $s \in S$ we have $|[\![\varphi]\!]^{\mathrm{f}}_{\mathcal{S}}(s) - [\![\varphi]\!]^{\mathrm{f}}_{\mathcal{S}'}(s)| \leq \epsilon$ for all MDPs $\mathcal{S}, \mathcal{S}'$ with $\|\mathcal{S}, \mathcal{S}'\| \leq \delta$.*
2. *Let Φ be any set of DCTL formulas such that the maximum nesting depth of any formula in Φ is k. For all $\epsilon > 0$, there is a $\delta > 0$ such that for all formulas $\varphi \in \Phi$ and all states $s \in S$ we have $|[\![\varphi]\!]^{\mathrm{p}}_{\mathcal{S}}(s) - [\![\varphi]\!]^{\mathrm{p}}_{\mathcal{S}'}(s)| \leq \epsilon$ for all MDPs $\mathcal{S}, \mathcal{S}'$ with $\|\mathcal{S}, \mathcal{S}'\| \leq \delta$.*

Notice that we get the continuity statement for the path semantics only for sets of formulas with bounded nesting depth. For example, consider a three state Markov chain $\mathcal{S} = (\{s_0, s_1, s_2\}, \tau, \{r\}, [\cdot])$ such that $\tau(s_0)$ is the distribution that chooses s_1 with probability $1 - \epsilon$ and chooses s_2 with probability ϵ, and $\tau(s_i)$ chooses s_i with probability 1 for $i = 1, 2$. Let $[r](s_0) = [r](s_1) = 0$ and $[r](s_2) = 1$. Consider the Markov chain $\mathcal{S}'$ which differs from $\mathcal{S}$ in that $\tau(s_0)$ chooses s_1 with probability 1. Then $\|\mathcal{S}, \mathcal{S}'\| = \epsilon$. However, consider the formulas $(\exists\Diamond_c)^n r$, for $n \geq 1$. Let $x_n = [\![(\exists\Diamond_c)^n r]\!]^{\mathrm{p}}_{\mathcal{S}}(s_0)$ (for $\langle c\rangle = 1$). Then $x_{n+1} = (1 - \epsilon)x_n + \epsilon$, and the limit as $n \to \infty$ is 1. On the other hand, $[\![(\exists\Diamond)^n r]\!]^{\mathrm{p}}_{\mathcal{S}'}(s_0) = 0$ for all n.

3 Model Checking DCTL over Transition Systems

The model-checking problem of a DCTL formula ϕ over an LTS $\mathcal{S}$ asks to compute the value $[\![\phi]\!](s)$ for all states s of $\mathcal{S}$ (since both semantics of DCTL coincide over LTSs, we write $[\![\cdot]\!]$ without superscript). Similar to CTL model checking [CES83], we recursively consider one of the basic subformulas ψ of ϕ and compute the valuation $[\![\psi]\!]$. Then we replace ψ in ϕ by a new proposition p_ψ with $[p_\psi] = [\![\psi]\!]$. Because of duality, it suffices to focus on model checking basic formulas of the forms $\exists\Diamond_c r$, $\forall\Diamond_c r$, and $\forall\triangle_c r$, for a proposition $r \in \Sigma$. We fix an LTS $\mathcal{S} = (S, \delta, \Sigma, [\cdot])$ and a discount interpretation $\langle\cdot\rangle$, and we write $[r] = q$ and $\langle c\rangle = \alpha$.

3.1 Model Checking $\Diamond$ (and $\Box$)

The fixpoint semantics of DCTL suggests iterative algorithms for evaluating formulas. In particular, $[\![\exists\Diamond_c r]\!]^{\mathrm{f}} = \lim_{n\to\infty} v_n$, where $v_0(s) = q(s)$, and $v_{n+1}(s) = q(s) \sqcup \alpha \max\{v_n(s') \mid s' \in \delta(s)\}$ for all $n \geq 0$. Over LTSs, the fixpoint is reached in a finite number of steps, namely, $[\![\exists\Diamond_c r]\!] = v_{|S|}$. To see this, observe that the value $[\![\exists\Diamond_c r]\!]^{\mathrm{f}}(s)$, the maximal (discounted) maximum over all paths from s, is obtained at a state in an acyclic prefix of some path from s. The argument that $[\![\forall\Diamond_c r]\!] = v_{|S|}$, where $v_{n+1}(s) = q(s) \sqcap \alpha \max\{v_n(s') \mid s' \in \delta(s)\}$, is slightly more involved. The value $[\![\forall\Diamond_c r]\!]^{\mathrm{f}}(s)$, the minimal (discounted) maximum over all paths from s, is again obtained at a state s' in an acyclic prefix of some path ρ from s. This is because if some state s'' were repeated on ρ before s', then the path ρ' that results from ρ by infinitely visiting s'' (and never visiting s') would achieve a smaller (discounted) maximum than ρ.

Lemma 2. *The evaluation of the fixpoint formulas for $[\![\forall\Diamond_c r]\!]^{\mathrm{f}}$ and $[\![\exists\Diamond_c r]\!]^{\mathrm{f}}$ terminates after at most $|S|$ iterations.*

3.2 Model Checking $\triangle$

Computing $[\![\forall\triangle_c r]\!](s)$ consists in minimizing the (discounted) average $[\![\triangle_c r]\!]$ over the paths from s. The (undiscounted) case $\alpha = 1$ is covered in Theorem 4.1 of [ZP96]: the value $[\![\forall\triangle_1 r]\!](s)$ is the minimum mean weight of a cycle reachable from s, which can be found using Karp's algorithm in $\mathcal{O}(|S|\cdot|\delta|)$ time. For $\alpha < 1$, the reasoning of [ZP96] can be used to show that the minimal discounted average is obtained on a path ρ' from s which, after some prefix ρ keeps repeating some simple cycle ℓ. Hence ℓ contains at most $|S|$ states. To find ρ', we use two steps. In the first phase, we find for each state s the simple cycle ℓ starting at s with the minimal discounted average. In the second phase, we find the best prefix-cycle combination $\rho\ell^\omega$.

Phase 1. We need to compute $L_\alpha(s) = \min\{[\![\triangle_c r]\!]^{\mathrm{p}}(\rho) \mid \rho \in \mathit{Traj}(s), \rho = (s_0 s_1 s_2 \ldots s_{n-1})^\omega, n \leq |S|\}$, where the value $[\![\triangle_c r]\!]^{\mathrm{p}}(\rho)$ is given by $\frac{1-\alpha}{1-\alpha^n} \cdot \sum_{i=0}^{n-1} \alpha^i \cdot q(s_i)$. Consider the recursion $v_0(s,s') = 0$ and $v_{n+1}(s,s') = q(s) + \alpha \cdot \min\{v_n(t,s') \mid t \in \delta(s)\}$. Then $v_n(s,s')$ minimizes $\sum_{i=0}^{n-1} \alpha^i \cdot q(s_i)$ over all finite paths $s_0 s_1 \ldots s_n$ with $s_0 = s$ and $s_n = s'$. Hence

$$L_\alpha(s) = (1-\alpha) \cdot \min\left\{\tfrac{v_1(s,s)}{1-\alpha^1}, \tfrac{v_2(s,s)}{1-\alpha^2}, \ldots, \tfrac{v_{|S|-1}(s,s)}{1-\alpha^{|S|-1}}\right\}.$$

For a fixed s', computing $\min\{v_n(t,s') \mid t \in \delta(s)\}$ for all $s \in S$ can be done in $\mathcal{O}(|\delta|)$ time. Therefore, v_{n+1} is obtained from v_n in $\mathcal{O}(|S|^2 + |S|\cdot|\delta|) = \mathcal{O}(|S|\cdot|\delta|)$ time. Hence, the computation of $v_{|S|}$ and L_α requires $\mathcal{O}(|S|^2\cdot|\delta|)$ time.

Phase 2. After a prefix of length n, the cost $L_\alpha(s)$ of repeating a cycle at state s has to be discounted by α^n, which is exactly the factor by which we discount $q(s)$ after taking that prefix. Hence, we modify the original LTS $\mathcal{S}$ into an LTS $\mathcal{S}^+$, as follows. For every state $s \in S$, we add a copy $\widehat{s}$ whose weight $w^+(\hat{s})$ we set to $L_\alpha(s)$; the weights $w^+(s)$ of states $s \in S$ remain $q(s)$. Moreover, for every $t \in S$ and $s \in \delta(t)$, we add $\widehat{s}$ as a successor to t, that is, $\delta^+(t) = \delta(t) \cup \{\hat{s} \mid s \in \delta(t)\}$ and $\delta^+(\hat{s}) = \{\hat{s}\}$. Taking the transition from t to $\widehat{s}$ corresponds to moving to s and repeating the optimal cycle from there. We can find the value of the optimal prefix-cycle combination starting from s as the *discounted distance* from s to $\hat{S} = \{\hat{s} \mid s \in S\}$ in the modified graph $\mathcal{S}^+$ with weigths w^+. Formally, given an LTS $\mathcal{S}$, a state s, a weight function $w\colon S \to \mathbb{R}^{\geq 0}$, a discount factor α, and a target set T, the minimal discounted distance from s to T is $\min\{\sum_{i=0}^{n-1} \alpha^i \cdot w(s_i) \mid s_0 s_1 \ldots s_{n-1} \in \mathit{FTraj}(s), s_{n-1} \in T\}$. This can be computed by a discounted version of the Bellman-Ford algorithm for finding shortest paths:

```
function DiscountedDistance(S, w, α, T):
  for every s ∈ S do
    if s ∈ T then d(s) := w(s) else d(s) := ∞;
  for i := 1 to |S| − 1 do
    for each s' ∈ δ(s) do
      if d(s) > w(s) + α · d(s') then d(s) := w(s) + α · d(s');
  return d.
```

Like the standard version, discounted Bellman-Ford runs in $\mathcal{O}(|S|\cdot|\delta|)$ time. Thus, the complexity of computing $[\![\forall \triangle_c r]\!]$ is dominated by the first phase. We conclude that the overall complexity of model checking a DCTL formula is polynomial in the size of the system and the size of the formula.

Theorem 3. *Given a* DCTL *formula* ϕ*, an LTS* $\mathcal{S} = (S, \delta, P, [\cdot])$*, and a discount interpretation* $\langle\cdot\rangle$*, the problem of model checking* ϕ *over* $\mathcal{S}$ *w.r.t.* $\langle\cdot\rangle$ *can be solved in time* $\mathcal{O}(|S|^2\cdot|\delta| \cdot |\phi|)$*.*

4 Model Checking DCTL over Markov Chains

Model checking $\Diamond$ and $\Box$: First Algorithm. Given a Markov chain $(S, \tau, \Sigma, [\cdot])$, with $r \in \Sigma$, and a parameter interpretation $\langle\cdot\rangle$, we wish to evaluate $[\![\exists\Diamond_c r]\!]^{\mathrm{p}}(s)$, for all states $s \in S$. As before, let $q = [r]$ and $\alpha = \langle c\rangle$. We give the algorithm for the case $\alpha < 1$: the case for $\alpha = 1$ can be solved along similar lines. When evaluating $[\![\exists\Diamond_c r]\!]^{\mathrm{p}}$ in a state s, we can start with the initial estimate of $q(s)$. If s is the state $s_{\max}$ with the maximum value of q, the initial estimate is the correct value. If s has the second greatest value for q, the estimate can only be improved if $s_{\max}$ is hit within a certain number l of steps, namely before the discount α^l becomes smaller than $q(s)/q(s_{\max})$. This argument can be recursively applied to all states.

Let $s_1, \ldots, s_n$ be an ordering of the elements of S such that $q(s_1) \geq q(s_2) \geq \ldots \geq q(s_n)$. Let P be the stochastic matrix associated with the chain, with $P(i,j) = p_{s_i,s_j}$. For all $0 \leq j < i \leq n$, such that $q(s_i) > 0$, let $k_{i,j} = \lfloor \log_\alpha \frac{q(s_i)}{q(s_j)} \rfloor$, with the convention that $\log_\alpha 0 = \infty$. Let $v(s_i) = [\![\exists\Diamond_\alpha r]\!]^{\mathrm{p}}(s_i)$. Then, $v(s_1) = q(s_1)$, and we can express the value of $v(s_i)$ in terms of the values $v(s_1), \ldots, v(s_{i-1})$. Let $K = \max\{k_{i,j} \mid k_{i,j} < \infty\}$, and for all $l > 0$, let $B^i_l = \{s_j \mid j < i \text{ and } k_{i,j} \geq l\}$. Intuitively, B^i_l contains those states that, if hit after l steps from s_i, can influence (increase) the value of $v(s_i)$. For the generic state s_i, the following holds.

$$v(s_i) = q(s_i) \cdot stay^i + \sum_{j=1}^{i-1} v(s_j) \cdot \sum_{l=1}^{k_{i,j}} \alpha^l go^i_{j,l}, \tag{1}$$

where $stay^i = \Pr_{s_i}\left[\bigwedge_{l>0} Z_l \notin B^i_l\right]$ and $go^i_{j,l} = \Pr_{s_i}\left[Z_l = s_j \wedge \bigwedge_{m=1}^{l-1} Z_m \notin B^i_m\right]$. It is easy to check that $stay^i + \sum_{j=1}^{i-1}\sum_{l=1}^{k_{i,j}} go^i_{j,l} = 1$. In the first phase, we deal with states s_i such that $q(s_i) > 0$. Since the sequence $(B^i_l)_{l>0}$ is decreasing, it can have at most $|S|$ different values. It follows that there exist integers $b^i_1 \leq \ldots \leq b^i_m \in \mathbb{N}$ and sets $X^i_1, \ldots, X^i_m \subseteq S$, such that $b^i_1 = 1$, $b^i_m = K$ and, for all $k = 1, \ldots, m-1$ and for all $b^i_k \leq l < b^i_{k+1}$, $B^i_l = X^i_k$. Notice that we can compute the actual value of the indices $b^i_1, \ldots, b^i_m$ in time $\mathcal{O}(|S| \cdot \log|S|)$, by ordering the values $k_{i,j}$ in a non-decreasing fashion and getting rid of the duplicates. Let P^i_k be the matrix obtained from P by turning the states in X^i_k into absorbing

states (sinks). Then,

$$go^i_{j,l} = \left((P^i_1)^{b^i_1} \cdot (P^i_2)^{b^i_2 - b^i_1} \cdot \ldots \cdot (P^i_{k-1})^{b^i_{k-1} - b^i_{k-2}} \cdot (P^i_k)^{l - b^i_k}\right)(i,j), \quad \text{for } b^i_k \leq l < b^i_{k+1}.$$

$$\begin{aligned}
\sum_{l=1}^{k_{i,j}} \alpha^l go^i_{j,l} &= \sum_{k=1}^{m} \sum_{l=b^i_k}^{b^i_{k+1}-1} \alpha^l go^i_{j,l} \\
&= \Big(\sum_{k=1}^{m} \alpha^{b^i_k} (P^i_1)^{b^i_1} \cdot (P^i_2)^{b^i_2 - b^i_1} \cdot \ldots \cdot (P^i_{k-1})^{b^i_{k-1} - b^i_{k-2}} \cdot \sum_{l=0}^{b^i_{k+1} - b^i_k - 1} \alpha^l (P^i_k)^l\Big)(i,j) \\
&= \Big(\sum_{k=1}^{m} \alpha^{b^i_k} (P^i_1)^{b^i_1} \cdot (P^i_2)^{b^i_2 - b^i_1} \cdot \ldots \cdot (P^i_{k-1})^{b^i_{k-1} - b^i_{k-2}} \cdot \frac{I - (\alpha P^i_k)^{b^i_k - b^i_{k-1} + 1}}{I - \alpha P^i_k}\Big)(i,j).
\end{aligned}$$

Each matrix $(P^i_k)^h$ can be computed by repeated squaring in time $\mathcal{O}(|S|^3 \cdot \log h)$. Some further calculations show that, for a fixed i, both $\sum_{l=1}^{k_{i,j}} \alpha^l go^i_{j,l}$ and $\sum_{l=1}^{k_{i,j}} go^i_{j,l}$ can be computed in time $\mathcal{O}(|S|^4 \cdot \log K)$. The value $stay^i$ is given by $1 - \sum_{j,l} go^i_{j,l}$. The total complexity of this phase is thus $\mathcal{O}(|S|^5 \cdot \log K)$.

In the second phase we consider those states s_i such that $q(s_i) = 0$. Let u be the smallest index i such that $q(s_i) = 0$, For each $i \geq u$, (1) becomes:

$$v(s_i) = \sum_{j=1}^{u-1} v(s_j) \cdot \sum_{l=1}^{\infty} \alpha^l go^i_{j,l}.$$

In this case, $go^i_{j,l}$ is the probability of hitting s_j after exactly l steps, while in the meanwhile avoiding all states with indices smaller than u. To efficiently compute $v(s_i)$, we define a stochastic matrix P_0 from P by adding an absorbing state t and using t to turn all states s_j with $j < u$ into transient states. Also, we set $\bar{v}$ to be the column vector containing the correct value $v(s_i)$, if $i < u$, and zero otherwise. Then,

$$v(s_i) = \sum_{j=1}^{u-1} v(s_j) \cdot \sum_{l=1}^{\infty} \alpha^l \cdot (P_0)^l(i,j) = ((I - \alpha P_0)^{-1} \cdot \bar{v})(i), \tag{2}$$

where I denotes the identity matrix. Solving the system (2) takes time $\mathcal{O}(|S|^3)$ using LUP decomposition. The time spent in the two phases amounts to $\mathcal{O}(|S|^5 \cdot \log K)$.

Second Algorithm. We present a different version of the first phase. The symbols $s_1, \ldots, s_n$, $k_{i,j}$, B^i_l, $stay^i$, and $go^i_{j,l}$ are defined as before.

Consider again equation 1. For $i = 1, \ldots, n$, let $K_i = \max\{k_{i,1}, \ldots, k_{i,i-1}\}$. For states s_i such that $q(s_i) > 0$, we can compute the values $go^i_{j,l}$ as follows. For $l > 0$, let C^i_l be the event "$Z_l \notin B^i_l$". It holds that $go^i_{j,l} = \Pr_{s_i}[Z_l = s_j \cap C^i_1 \cap \ldots \cap C^i_{l-1}] = \Pr_{s_i}[Z_l = s_j \mid C^i_1 \cap \ldots \cap C^i_{l-1}] \cdot \Pr_{s_i}[C^i_{l-1} \mid C^i_1 \cap \ldots \cap C^i_{l-2}] \cdot \ldots \cdot \Pr_{s_i}[C^i_1]$. For each $j = 1, \ldots, i-1$ and $l > 0$, let $p(s_j, l) = \Pr_{s_i}[Z_l = s_j \mid \bigcap_{m=1}^{l-1} Z_m \notin B^i_m]$. In words, $p(s_j, l)$ is the probability that, starting in s_i, the system reaches s_j after exactly l steps, given that in each previous step it does not hit states that can influence $v(s_i)$. For all $j = 1, \ldots, n$ and $0 < l \leq K$, we can compute $\Pr_{s_i}[C^i_l \mid C^i_1 \cap \ldots \cap C^i_{l-1}]$ together with $p(s_j, l)$ using the following recursion:

$$
\begin{aligned}
p(s_j, 1) &= P(i, j) \\
\Pr_{s_i}[C^i_1] &= \sum\{p(s_t, l) \mid s_t \notin B^i_1\} \\
p(s_j, l+1) &= \frac{1}{\Pr_{s_i}[C^i_l \mid C^i_1 \cap \ldots \cap C^i_{l-1}]} \cdot \sum\{P(t, j) \cdot p(s_t, l) \mid s_t \notin B^i_l\} \\
\Pr_{s_i}[C^i_{l+1} \mid C^i_1 \cap \ldots \cap C^i_l] &= \sum\{p(s_t, l) \mid s_t \notin B^i_{l+1}\}
\end{aligned}
$$

For a fixed i, the previous recursion takes time $\mathcal{O}(|S|^2 \cdot K)$. Then,

$$
go^i_{j,l} = p(s_j, l) \cdot \prod_{m=1}^{l-1} \Pr_{s_i}\left[C^i_m \mid C^i_1 \cap \ldots \cap C^i_{m-1}\right]. \tag{3}
$$

It follows that, for a fixed i, all values $go^i_{j,l}$ can be computed in time $\mathcal{O}(|S|^2 \cdot K)$. The value $stay^i$ is given by $1 - \sum_{j,l} go^i_{j,l}$. Running this procedure for every state with nonzero value of q takes thus $\mathcal{O}(|S|^3 \cdot K)$.

If we use the first algorithm for the states s with $q(s) = 0$, we obtain an alternative algorithm for Model Checking $[\![\exists \Diamond_c r]\!]^{\mathrm{p}}$, whose complexity is $\mathcal{O}(|S|^3 \cdot K)$. Notice that if the matrix P is represented as an $|S| \times |S|$ array, the previous complexity is less than quadratic in the size of the input. Which algorithm performs better on a given instance of the problem depends clearly on the ratio $\frac{|S|^2 \cdot \log K}{K}$, the first algorithm being preferable when the ratio is smaller than 1.

Model checking $\triangle$. Let P be the stochastic matrix representing a Markov chain $(S, \tau, \Sigma, [\cdot])$ and let $\langle\cdot\rangle$ be a parameter interpretation. As before, let $r \in \Sigma$, $q = [r]$ and $\alpha = \langle c \rangle$. If we let $[\![\exists \triangle_c r]\!]$ and q denote column vectors, we obtain the following classical equation [FV97]:

$$
[\![\exists \triangle_c r]\!] = (1 - \alpha) \cdot \sum_{i \geq 0} \alpha^i P^i q = (1 - \alpha) \cdot (I - \alpha P)^{-1} q,
$$

where I is the identity matrix. Thus, we can compute the value $[\![\exists\triangle_c r]\!](s)$ for each state $s \in S$ by solving a linear system with $|S|$ variables. This takes time $\mathcal{O}(|S|^{\log_2 7})$ using Strassen's algorithm or $\mathcal{O}(|S|^3)$ using LUP decomposition.

Complexity of DCTL Model Checking over Markov Chains. The overall complexity is polynomial in the size of the system and the size of the formula. For this result, we assume that the basic operations such as addition and multiplication can be done in constant time.

Theorem 4. *Given a* DCTL *formula* ϕ*, a Markov chain* $\mathcal{S} = (S, \tau, P, [\cdot])$*, and a discount interpretation* $\langle\cdot\rangle$*, the problem of model checking* $\mathcal{S}$ *w.r.t.* ϕ *and* $\langle\cdot\rangle$ *can be solved in time polynomial in* $|S|$*,* $|[\cdot]|_b$*,* $|\langle\cdot\rangle|_b$ *and* $|\phi|$*.*

5 Model Checking DCTL over Markov Decision Processes

For Markov decision processes, the path and the fixpoint semantics do not coincide, as stated by Theorem 1: hence, we need to provide algorithms for both semantics. In view of the duality laws for negation, and in view of the recursive definition of DCTL, it suffices to provide algorithms for computing $\exists\Diamond_c r$, $\forall\Diamond_c r$, $\exists\triangle_c r$, and $\forall\triangle_c r$, for a predicate r. We consider a Markov decision process $\mathcal{S} = (S, \tau, \Sigma, [\cdot])$ and a discount interpretation $\langle\cdot\rangle$; to simplify the notation, we let $[r] = q$ and $\langle c\rangle = \alpha$.

5.1 Model Checking $\Diamond$ in the path Semantics

If $\alpha = 0$, then trivially $[\![\exists\Diamond_c r]\!]^{\mathrm{p}}(s) = [\![\forall\Diamond_c r]\!]^{\mathrm{p}}(s) = q(s)$ at all $s \in S$, so in the following we assume $0 < \alpha \leq 1$. The problem of computing $[\![\exists\Diamond_c r]\!]^{\mathrm{p}}$ on an MDP can be viewed as an optimization problem, where the goal is to maximize the expected value of the sup of q over a path. As a preliminary step to solve the problem, we note that in general the optimal strategy is *history dependent,* that is, the choice of distribution at a state depends in general on the past sequence of states visited by the path.

Example 1. Consider the system in Figure 1 and assume $\alpha = 1$. The optimal choice in state t_2 depends on whether t_1 was hit or not. If it was, the current sup is 0.8 and the right choice is a_1, because with probability $\frac{1}{2}$ the sup will increase to 1. If t_2 was not hit, the right choice is a_2, because it gives a certain gain of 0.8.

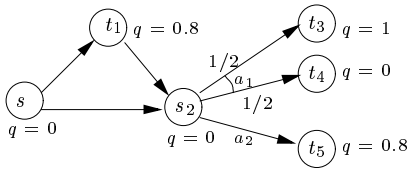


Fig. 1. An MDP requiring a memory strategy for $[\![\exists \Diamond_c r]\!]^{\mathrm{p}}(s)$.

While the above example indicates that the optimal strategy is in general history-dependent, it also suggests that all a strategy needs to remember is the sup value that has occurred so far along the path. For $\pi \in \Pi$, $s \in S$, and $x \in \mathbb{R}$ we define

$$\mathrm{Esup}^{\pi}(s, x) = \mathrm{E}^{\pi}_{s}[x \sqcup \sup_{i>0} \alpha^{i} q(Z_i)].$$

The term x corresponds to the (appropriately discounted) sup value that has occurred so far in the past of a path. Obviously, $[\![\exists \Diamond_c r]\!]^{\mathrm{p}}(s) = \sup_{\pi \in \Pi} \mathrm{Esup}^{\pi}(s, q(s))$ and $[\![\forall \Diamond_c r]\!]^{\mathrm{p}}(s) = \inf_{\pi \in \Pi} \mathrm{Esup}^{\pi}(s, q(s))$. For a strategy π and $s \in S$, let $\Pr(t \mid s, \pi) = \sum_{a \in \tau(s)} \pi(s)(a)\, a(t)$ be the probability of a transition from s to t under π, and let $\pi[s]$ to be the strategy defined by $\pi[s](\rho) = \pi(s\rho)$ for all $\rho \in \mathit{Traj}$. In words, $\pi[s]$ is the strategy that behaves like π after an initial transition from s. For all $s \in S$ and $x \in \mathbb{R}$, the quantity $\mathrm{Esup}^{\pi}(s, x)$ satisfies the following recurrence equation:

$$\mathrm{Esup}^{\pi}(s, x) = \alpha \sum_{t \in S} \mathrm{Esup}^{\pi[s]}(t, \frac{x}{\alpha} \sqcup q(t)) \Pr(t \mid s, \pi). \tag{4}$$

Intuitively, this recursion can be understood as follows. Under strategy π, at a state $s = s_m$ of a path $s_0 s_1 \ldots$, the quantity $\mathrm{Esup}^{\pi}(s_m, x)$ represents the value of $\sup_{i>0} E^{\pi}_{s_m}[\alpha^i q(Z_i)]$ given that $\sup_{0 \le i \le m} \alpha^{-i} q(s_{m-i}) = x$. The recursion (4) then relates $\mathrm{Esup}^{\pi}(s, x)$ to $\mathrm{Esup}^{\pi[s]}(t, y)$ at the successorts t of s, where at t we consider the new conditioning $y = x/\alpha \sqcup q(t)$, thus discounting x by α^{-1} (as s is one step before t), and taking into account the value $q(s)$ seen at s.

The recursion (4) can be proved as follows.

$$\begin{aligned}
&\alpha \sum_{t\in S} \mathrm{Esup}^{\pi[s]}(t, \frac{x}{\alpha} \sqcup q(t)) \Pr(t \mid s, \pi) \\
=&\alpha \sum_{t\in S} \mathrm{E}_t^{\pi[s]}\Big[\frac{x}{\alpha} \sqcup q(t) \sqcup \sup_{i>0} \alpha^i q(Z_i)\Big] \Pr(t \mid s, \pi) \\
=&\sum_{t\in S} \mathrm{E}_t^{\pi[s]}\Big[x \sqcup \alpha q(t) \sqcup \sup_{i>0} \alpha^{i+1} q(Z_i)\Big] \Pr(t \mid s, \pi) \\
=&\sum_{t\in S} \mathrm{E}_t^{\pi[s]}\Big[x \sqcup \sup_{i\geq 0} \alpha^{i+1} q(Z_i)\Big] \Pr(t \mid s, \pi) \\
=&\,\mathrm{E}_s^{\pi}\Big[x \sqcup \sup_{i>0} \alpha^i q(Z_i)\Big].
\end{aligned}$$

The idea behind the computation of $\sup_{\pi\in\Pi} \mathrm{Esup}^{\pi}(s, q(s))$ is to turn the recurrence equation (4) into an optimization problem, where at each $s \in S$ we seek the strategy π that maximizes the right hand side. The optimization problem to compute these quantities is phrased in terms of the variables $v(s, x)$, representing the value of $\mathrm{Esup}(s, x)$. Since we are ultimately interested in the value of $\mathrm{Esup}(s, q(s))$ for $s \in S$, and since if $x \geq 1$ we have $\mathrm{Esup}^{\pi'}(t, x) = x$ for all $t \in S$ and $\pi' \in \Pi$, it suffices to consider values for x that belong to the finite set $X = \{q(s)/\alpha^k \mid s \in S \sqcap k \in \mathbb{N} \sqcap q(s) < \alpha^k\}$. We set up the following set of equations in the variables $\{v(s, x) \mid s \in S \sqcap x \in X\}$:

$$v(s, x) = \begin{cases} x & \text{if } x \geq 1; \\ x \sqcup \alpha \max\limits_{a\in\tau(s)} \sum\limits_{t\in S} v(t, \frac{x}{\alpha} \sqcup q(t))\, a(t) & \text{otherwise.} \end{cases} \tag{5}$$

The following theorem relates the least fixpoint of (5) to $[\![\exists \Diamond_c r]\!]^{\mathrm{p}}$.

Theorem 5. *Let $\{v^*(s, x) \mid s \in S \sqcap x \in X\}$ be the least (pointwise) fixed point of the set of equations (5). Then, we have $[\![\exists \Diamond_c r]\!]^{\mathrm{p}}(s) = v^*(s, q(s))$ for all $s \in S$.*

Proof. We consider an iterative evaluation of the least fixpoint (5), given by $v_0(s, x) = x$ and, for $k \geq 0$, by

$$v_{n+1}(s, x) = \begin{cases} x & \text{if } x \geq 1; \\ \alpha \max\limits_{a\in\tau(s)} \sum\limits_{t\in S} v_n(t, \frac{x}{\alpha} \sqcup q(t))\, a(t) & \text{otherwise.} \end{cases} \tag{6}$$

The proof consists of two parts: (i) showing that for all $s \in S$ and $x \in X$ there is a strategy $\pi^* \in \Pi$ such that $\mathrm{E}_s^{\pi^*}[x \sqcup \sup_{0<i\leq n} \alpha^i q(Z_i)] = v_n(s,x)$, and (ii) showing that for all $\pi \in \Pi$, all $s \in S$, and all $x \in X$ we have $\mathrm{E}_s^{\pi}[x \sqcup \sup_{0<i\leq n} \alpha^i q(Z_i)] \leq v_n(s,x)$. Once (i) and (ii) are proved, the result follows from

$$\lim_{n\to\infty} v_n = v^* \qquad \lim_{n\to\infty} \mathrm{E}_s^{\pi}[x \sqcup \sup_{0\leq i\leq n} \alpha^i q(Z_i)] = \mathrm{E}_s^{\pi}[x \sqcup \sup_{i\geq 0} \alpha^i q(Z_i)].$$

We prove only (i), since the proof of (ii) is similar. First, notice that if we define $X' = \{q(s)/\alpha^k \mid s \in S \sqcap k \in \mathbb{N}\}$, (6) can be written as

$$v_{n+1}(s,x) = \alpha \max_{a\in\tau(s)} \sum_{t\in S} v_n(t, \frac{x}{\alpha} \sqcup q(t))\, a(t) \tag{7}$$

for all $s \in S$ and $x \in X'$. The strategy π^* is in general a function of $\langle s, x\rangle \in S \times X'$. We define it inductively: π_0^* is arbitrary; for $n \geq 0$, π_{n+1}^* first chooses a distribution $a \in \tau(s)$ that realizes the maximum in (6), and then upon a transition from s to some $t \in S$, proceeds as π_n^* from $\langle t, \frac{x}{\alpha} \sqcup q(t)\rangle$. By induction, proceeding in analogy with the proof of (4), we have for all $n \geq 0$, all $s \in S$ and all $x \in X'$:

$$\begin{aligned}
v_{n+1}(s,x) &= \alpha \max_{a\in\tau(s)} \sum_{t\in S} v_n(t, \frac{x}{\alpha} \sqcup q(t))\, a(t) \\
&= \alpha \max_{a\in\tau(s)} \sum_{t\in S} \mathrm{E}_t^{\pi_n^*}\Big[\frac{x}{\alpha} \sqcup q(t) \sqcup \sup_{0<i\leq n} \alpha^i q(Z_i)\Big]\, a(t) \\
&= \max_{a\in\tau(s)} \sum_{t\in S} \mathrm{E}_t^{\pi_n^*}\Big[x \sqcup \alpha q(t) \sqcup \sup_{0<i\leq n} \alpha^{i+1} q(Z_i)\Big]\, a(t) \\
&= \max_{a\in\tau(s)} \sum_{t\in S} \mathrm{E}_t^{\pi_n^*}\Big[x \sqcup \sup_{0\leq i\leq n} \alpha^{i+1} q(Z_i)\Big]\, a(t) \\
&= \mathrm{E}_s^{\pi_{n+1}^*}\Big[x \sqcup \sup_{0<i\leq n} \alpha^i q(Z_i)\Big],
\end{aligned}$$

leading to the desired result. ∎

To compute $[\![\forall\Diamond_c r]\!]^{\mathrm{p}}$, we simply replace $\max_{a\in\tau(s)}$ with $\min_{a\in\tau(s)}$ in (5), and again consider the least fixed point. The least fixed points for $[\![\exists\Diamond_c r]\!]^{\mathrm{p}}$ and $[\![\forall\Diamond_c r]\!]^{\mathrm{p}}$ can be computed by linear programming, following a standard approach.

Theorem 6. *The following assertions hold.*

1. *Consider the following linear programming problem in the set of variables* $\{v(s,x) \mid s \in S \sqcap x \in X\}$*: minimize* $\sum_{s\in S}\sum_{x\in X} v(s,x)$ *subject to*

$$v(s,x) \geq x \qquad v(s,x) \geq \alpha \sum_{t\in S} \tilde{v}(t, \frac{x}{\alpha} \sqcup q(t))\, a(t)$$

   for all $s \in S$*, all* $x \in X$*, and all* $a \in \tau(s)$*, where* $\tilde{v}(t,x)$ *is 1 if* $x \geq 1$ *and is* $v(t,x)$ *otherwise. Denote by* $\{\hat{v}(s,x) \mid s \in S \sqcap x \in X\}$ *an optimal solution. Then,* $\hat{v}(s,q(s)) = v^*(s,q(s)) = [\![\exists\Diamond_c r]\!]^{\mathrm{p}}(s)$*.*
2. *Consider the following linear programming problem in the set of variables* $\{v(s,x), u(s,x) \mid s \in S \sqcap x \in X\}$*: minimize* $\sum_{s\in S}\sum_{x\in X}(v(s,x) - u(s,x))$ *subject to*

$$v(s,x) \geq x \qquad v(s,x) \geq u(s,x) \qquad u(s,x) \leq \alpha \sum_{t\in S} \tilde{v}(t, \frac{x}{\alpha} \sqcup q(t))\, a(t)$$

   for all $s \in S$*, all* $x \in X$*, and all* $a \in \tau(s)$*, where* $\tilde{v}(t,x)$ *is 1 if* $x \geq 1$ *and is* $v(t,x)$ *otherwise. Denote by* $\{\hat{v}(s,x), \hat{u}(s,x) \mid s \in S \sqcap x \in X\}$ *an optimal solution. Then,* $\hat{v}(s,q(s)) = v^*(s,q(s)) = [\![\forall\Diamond_c r]\!]^{\mathrm{p}}(s)$*.*

The linear programming problems in the above theorem consist of at most $2\cdot|S|\cdot|X|$ variables. If $\alpha = 1$, then $|X| = |S|$; otherwise, $|X| = -|S|\log_\alpha q_{min}$, where $q_{min} = \min\{q(s) \mid s \in S \sqcap q(s) > 0\}$. Finally, notice that $|X|$ is linear in the size of the input encoding of the MDP, if q-values are encoded in binary notation. This leads to the following result.

Corollary 1. *For an MDP* $\mathcal{S} = (S, \tau, \Sigma, [\cdot])$ *and* $r \in \Sigma$*, the valuations* $[\![\exists\Diamond_c r]\!]^{\mathrm{p}}$ *and* $[\![\forall\Diamond_c r]\!]^{\mathrm{p}}$ *can be computed in time polynomial in* $|\mathcal{S}|_b$ *and* $|\langle c\rangle|_b$*.*

In addition to linear programming, it is possible to compute $[\![\exists\Diamond_c r]\!]^{\mathrm{p}}$ and $[\![\forall\Diamond_c r]\!]^{\mathrm{p}}$ also by value iteration, using (6), as well as by policy iteration, adapting standard algorithms to the task (see e.g. [Ber95]).

5.2 Model Checking $\Diamond$ in the fixpoint semantics.

The computation of $[\![\exists\Diamond_c r]\!]^{\mathrm{f}}$ and $[\![\forall\Diamond_c r]\!]^{\mathrm{f}}$ on an MDP can be performed by transforming the fixpoints into linear programming problems, following a standard approach. For example, for $[\![\forall\Diamond_c r]\!]^{\mathrm{f}}$ we consider the following linear programming problem in the set of variables $\{v(s), u(s) \mid s \in S\}$:

minimize $\sum_{s\in S}(v(s)-u(s))$ subject to $v(s)\geq q(s)$, $v(s)\geq u(s)$, and $u(s)\leq \alpha\sum_{t\in S}v(t)a(t)$ for all $s\in S$ and all $a\in\tau(s)$. Denoting with $\{v^*(s),u^*(s)\in\mathbb{R}\mid s\in S\}$ an optimal solution, we have $[\![\forall\Diamond_c r]\!]^{\mathrm{f}}(s)=v^*(s)$ at all $s\in S$. Again, this can be solved in time polynomial in $|\mathcal{S}|_b$ and $|\alpha|_b$.

Theorem 7. *The following assertions hold.*

1. *Consider the following linear programming problem in the set of variables $\{v(s)\mid s\in S\}$: minimize $\sum_{s\in S}v(s)$ subject to $v(s)\geq q(s)$ and $v(s)\geq\alpha\sum_{t\in S}v(t)a(t)$ for all $s\in S$ and all $a\in\tau(s)$. Denoting with $\{v^*(s)\in\mathbb{R}\mid s\in S\}$ an optimal solution, we have $[\![\exists\Diamond_c r]\!]^{\mathrm{f}}(s)=v^*(s)$ at all $s\in S$.*
2. *Consider the following linear programming problem in the set of variables $\{v(s),u(s)\mid s\in S\}$: minimize $\sum_{s\in S}(v(s)-u(s))$ subject to $v(s)\geq q(s)$, $v(s)\geq u(s)$, and $u(s)\leq\alpha\sum_{t\in S}v(t)a(t)$ for all $s\in S$ and all $a\in\tau(s)$. Denoting with $\{v^*(s),u^*(s)\in\mathbb{R}\mid s\in S\}$ an optimal solution, we have $[\![\forall\Diamond_c r]\!]^{\mathrm{f}}(s)=v^*(s)$ at all $s\in S$.*

Corollary 2. *For an MDP $\mathcal{S}=(S,\tau,\Sigma,[\cdot])$ and $r\in\Sigma$, the valuations $[\![\exists\Diamond_c r]\!]^{\mathrm{f}}$ and $[\![\forall\Diamond_c r]\!]^{\mathrm{f}}$ can be computed in time polynomial in $|\mathcal{S}|_b$ and $|\langle c\rangle|_b$.*

5.3 Model Checking $\triangle$.

In an MDP, for $\alpha<1$, we have that $[\![\exists\triangle_c r]\!]^{\mathrm{p}}=[\![\exists\triangle_c r]\!]^{\mathrm{f}}$; hence, a single model-checking algorithm suffices for both semantics. Notice also that $[\![\forall\triangle_c r]\!]^{\mathrm{p}}=\mathbf{1}-[\![\exists\triangle_c\neg r]\!]^{\mathrm{p}}$, so that we need to consider only the path quantifier $\exists$. On the other hand, we must distinguish the case where $\alpha<1$ from the case $\alpha=1$ (meaningful only for the path semantics), since the dynamic programming algorithms required for solving these problems are rather different. For $\alpha<1$, the problem can be solved by the standard methods used for discounted long-run average problems [Ber95].

Proposition 1. *Assume $\alpha<1$, and consider the following linear programming problem in the set of variables $\{v(s)\mid s\in S\}$: minimize $\sum_{s\in S}v(s)$ subject to $v(s)\geq(1-\alpha)q(s)+\alpha\sum_{t\in S}v(t)a(t)$ for all $s\in S$ and all $a\in\tau(s)$. Denoting by $\{v^*(s)\mid s\in S\}$ an optimal solution, we have $[\![\exists\triangle_c r]\!]^{\mathrm{p}}(s)=v^*(s)$ for all $s\in S$.*

If $\alpha=1$, the problem consists in computing the maximal long-run average reward (or cost) of an MDP. The classical solutions for this problem, unfortunately, rely on structural assumptions about the MDP [Ber95]: i.e. every state

is reachable with positive probability from every other state. These restrictions can be overcome, using the methods of [dA97]. The idea is to separately analyze each *maximal end component,* and then combine the results as an instance of an undiscounted reachability problem.

5.4 Complexity of DCTL model checking over MDPs.

The following result summarizes the complexity of model-checking DCTL formulas over MDPs.

Theorem 8. *Given a* DCTL *formula ϕ, an MDP $\mathcal{S}$, and a discount interpretation $\langle\cdot\rangle$, we can compute $[\![\phi]\!]^{\mathrm{p}}$ and $[\![\phi]\!]^{\mathrm{f}}$ over $\mathcal{S}$ for $\langle\cdot\rangle$ in time polynomial in $|\mathcal{S}|_b$ and $|\langle\cdot\rangle|_b$ and exponential in $|\phi|$.*

The exponential dependency on the length of ϕ is due to the fact that model checking $\exists\Diamond r$ and $\exists\triangle r$ requires polynomial time in $||r||_b$, and may produce valuations $[\![\exists\Diamond r]\!]$ and $[\![\exists\triangle r]\!]$ whose binary encoding size is polynomial in $||r||_b$, leading to a potential exponential blow-up of the binary encodings of the valuations.

References

[AD94] R. Alur and D.L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.

[BdA95] 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.

[Ber95] D.P. Bertsekas. *Dynamic Programming and Optimal Control.* Athena Scientific, 1995. Volumes I and II.

[BHHK00] C. Baier, B. R. Haverkort, H. Hermanns, and J.-P. Katoen. Model checking continuous-time markov chains by transient analysis. In *Computer Aided Verification*, pages 358–372, 2000.

[CES83] 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.

[CGP99] E.M. Clarke, O. Grumberg, and D. Peled. *Model Checking.* MIT Press, 1999.

[dA97] L. de Alfaro. *Formal Verification of Probabilistic Systems.* PhD thesis, Stanford University, 1997. Technical Report STAN-CS-TR-98-1601.

[dAHM03] L. de Alfaro, T.A. Henzinger, and R. Majumdar. Discounting the future in systems theory. In *Proc. 30th Int. Colloq. Aut. Lang. Prog.*, Lect. Notes in Comp. Sci. Springer-Verlag, 2003.

[DEP02] J. Desharnais, A. Edalat, and P. Panangaden. Bisimulation for labelled markov processes. *Information and Computation*, 179(2):163–193, 2002.

[EL86] E.A. Emerson and C.L. Lei. Efficient model checking in fragments of the propositional μ-calculus. In *Proc. First IEEE Symp. Logic in Comp. Sci.*, pages 267–278, 1986.

[FV97] J. Filar and K. Vrieze. *Competitive Markov Decision Processes.* Springer-Verlag, 1997.

[Han94] H. Hansson. *Time and Probabilities in Formal Design of Distributed Systems.* Real-Time Safety Critical Systems Series. Elsevier, 1994.

[HK97] M. Huth and M. Kwiatkowska. Quantitative analysis and model checking. In *Proc. 12th IEEE Symp. Logic in Comp. Sci.*, pages 111–122, 1997.

[Koz83] D. Kozen. A probabilistic PDL. In *Proc. 15th ACM Symp. Theory of Comp.*, pages 291–297, 1983.

[KSK66] J.G. Kemeny, J.L. Snell, and A.W. Knapp. *Denumerable Markov Chains.* D. Van Nostrand Company, 1966.

[Kwi03] M. Kwiatkowska. Model checking for probability and time: From theory to practice. In *Proc. 18th Annual IEEE Symposium on Logic in Computer Science (LICS'03)*, pages 351–360. IEEE Computer Society Press, 2003.

[McI98] A. McIver. Reasoning about efficiency within a probabilitic μ-calculus. In *Proc. of PROBMIV*, pages 45–58, 1998. Technical Report CSR-98-4, University of Birmingham, School of Computer Science.

[MM02] A. McIver and C. Morgan. Games, probability, and the quantitative μ-calculus. In *LPAR 02: Logic Programming, Artificial Intelligence, and Reasoning*, LNCS 2514, pages 292–310. Springer, 2002.

[MP91] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Specification.* Springer-Verlag, New York, 1991.

[Wil91] D. Williams. *Probability With Martingales.* Cambridge University Press, 1991.

[ZP96] U. Zwick and M. S. Paterson. The complexity of mean payoff games on graphs. *Theoretical Computer Science*, 158(1–2):343–359, 1996.