

PDF hosted at the Radboud Repository of the Radboud University Nijmegen

The following full text is a preprint version which may differ from the publisher's version.

For additional information about this publication click this link.

<http://hdl.handle.net/2066/117116>

Please be advised that this information was generated on 2017-12-05 and may be subject to change.

More Or Less True: DCTL for Continuous-Time MDPs

David N. Jansen

Radboud Universiteit, Model-based System Development,
Nijmegen, The Netherlands. dnjansen@cs.ru.nl

Abstract. Discounted Computation Tree Logic is a logic that measures utility (as a real value in the interval $[0,1]$) instead of discrete truth (only 0 or 1). It is able to express properties that give more weight to the near future than to the far future. This article extends earlier work on DCTL with time, to continuous-time Markov chains and continuous-time Markov decision processes. It presents model checking algorithms for the two possible semantics of DCTL.

This technical report is an extended version of [4].

1 Introduction

In what context is it appropriate to announce: “It is going to rain”? Unless the world goes away in the meantime, one could say that this statement is always true: it will eventually rain again, perhaps this afternoon, perhaps next week, or perhaps in three months. However, we are tempted to say that the statement is more appropriate in a situation where we expect rain very soon.

Similarly, all computer systems will fail eventually. Therefore, a requirement like “The system will forever fulfil its task” is, in some sense, never satisfied. Models to verify requirements on reactive systems (which engage in an ongoing interaction with their environment and are not meant to terminate) typically assume that the system has infinitely long behaviours, while every implementation will stop fulfilling its task at some time. In such situations, it is more appropriate to abandon speaking about absolute truth and rather look at the *utility* of a system. The longer a system fulfils its task, the higher its utility.

In another situation, namely testing, engineers are satisfied if the system fulfils its requirements for a certain amount of time. An error appearing after tenthousand successful tests is perhaps ascribed to mistyping some input, or it may just end up in the list of known bugs. An error appearing in the third test will cause a much stronger response: the system will have to be repaired.

In these situations, the point is that utility requirements give more weight to the near future than to the far future. The logic Discounted Computation Tree Logic (DCTL) was defined in [1] to express this kind of utility requirements. DCTL is interpreted in a quantitative setting, where any real number between 0 and 1 is a truth value (0 corresponding to “false” or “useless” and 1 to “true” or “most useful”), similar to a “degree of truth” in fuzzy logic [8]. To achieve the difference in weight between near and far future, it introduces *discounting*: an influence on the utility (a rainy state, an error in the system under test) is weighted by the length of the path to reach it.

Another possibility to regard DCTL properties is to think of an *impatient* game player. Assume somebody plays a (single-player) game that may involve both random choices and strategic or nondeterministic choices, to be resolved by the player. In some states, the player wins a reward. The player, however, is not ready to play the game infinitely long, but decides at any moment to abort the game with some probability. The utility expressed by a suitable DCTL property is a measure of the expected reward from the game.

DCTL was defined in [1] for discrete-time Markov decision processes (MDP), and their special cases discrete-time Markov chains (DTMC) and labelled transition systems (LTS). In these models, the only way to measure the length of a path is to count the number of transitions. While this works well when each step takes approximately the same amount of time, it becomes unexact when step timings differ.

Contributions. In this article, I therefore undertake to extend the work of [1] to continuous-time Markov decision processes (CTMDP) and their special case continuous-time Markov chains (CTMC). These formalisms equip each transition with a *rate* indicating how fast (on average) the transition is taken.

DCTL temporal operators, interpreted over DTMCs, were shown in [1] to have two differing semantics: a *fixpoint* and a *path* semantics. I show that these variants also exist in the continuous-time model and provide model checking algorithms for both. To resolve the nondeterminism in CTMDPs, one needs a scheduler or a strategy. The fixpoint semantics by definition only allows so-called positional schedulers, i. e. schedulers that always choose the same action in a given state. In the path semantics, in principle, several classes of schedulers are possible. However, as in the discrete-time case, I only provide a model checking algorithm for CTMCs, systems without nondeterminism.

2 Preliminaries

Notations. We denote the minimum of two values by $r \sqcap s := \min\{r, s\}$ and their maximum by $r \sqcup s := \max\{r, s\}$. For a $n \times n$ -matrix A , let $A|_{i\dots j}$ be the $n \times n$ -matrix where all rows except the rows $i, i+1, \dots, j$ have been set to 0.

Definition 1. A *continuous-time Markov decision process (CTMDP)* is a tuple $\mathcal{M} = (S, A, R, AP, L)$ consisting of

- a nonempty set of states S ;
- a nonempty, finite set of actions A ;
- a transition rate matrix $R : S \times A \times S \rightarrow \mathbb{R}_{\geq 0}$, written $R^a(s, s')$.
- a nonempty, finite set of atomic propositions AP ;
- a labelling function $L : S \times AP \rightarrow [0, 1]$ (here we extend the usual definition).

We denote the components of \mathcal{M} by $S_{\mathcal{M}}, A_{\mathcal{M}}, R_{\mathcal{M}}, AP_{\mathcal{M}}, L_{\mathcal{M}}$, respectively, and we omit the subscript \mathcal{M} if \mathcal{M} is clear from the context.

Informally speaking, the behaviour of a CTMDP is as follows: It is always in a state, say s . The atomic proposition p has utility $L(s, p) \in [0, 1]$. Transitions are triggered according to the following rule: The CTMDP first chooses nondeterministically an action a . Then, the probability that the transition to s' becomes enabled within at most t time units is $1 - e^{-R^a(s, s')t}$. If there are several states s' with $R^a(s, s') > 0$, the first transition that becomes enabled is taken. In effect, this leads to an exponential distribution with the so-called exit rate of $E^a(s) := \sum_{s' \in S} R^a(s, s')$. Its cumulative density function is $\Pr\left(s \xrightarrow[a]{\leq t} \mid a\right) = 1 - e^{-E^a(s)t}$ and its probability density function is $\text{pdf}\left(s \xrightarrow[a]{t} \mid a\right) = E^a(s)e^{-E^a(s)t}$.

The probability to go to state s' , if in state s action a is chosen, is then $P^a(s, s') := R^a(s, s')/E^a(s)$. Together, the transition to s' will be taken in time at most t with probability

$$\Pr\left(s \xrightarrow[a]{\leq t} s' \mid a\right) = P^a(s, s')(1 - e^{-E^a(s)t}) \quad .$$

We will also use $E_{\mathcal{M}}(s) := \max_{a \in A} E_{\mathcal{M}}^a(s)$ and $E_{\mathcal{M}} := \max_{s \in S} E_{\mathcal{M}}(s)$.

It may happen that for some states and actions, $E^a(s) = 0$. In that case, we say that s is *a-absorbing*. If the CTMDP is in state s and chooses action a , the behaviour stops, and the CTMDP stays in s forever. If state s is *a-absorbing* for each action $a \in A$, we call s (*strictly*) *absorbing*.

A *continuous-time Markov chain (CTMC)* is a CTMDP that does not contain nondeterministic choices, i. e., a CTMDP \mathcal{M} with $|A_{\mathcal{M}}| = 1$.

Example 2. Figure 1 depicts an example of a simple continuous-time Markov chain. The proposition *black* denotes the “blackness” of a state. There is, a. o., a transition from s to t with rate $R^a(s, t) = 3$. State u is absorbing. The unique action has been omitted from the figure.

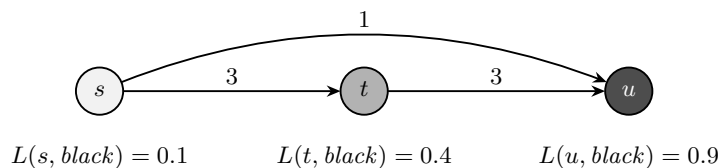


Fig. 1. Example CTMC

A *path* is a (finite or infinite) sequence of the form

$$s_0 \xrightarrow[a_0]{t_0} s_1 \xrightarrow[a_1]{t_1} s_2 \rightarrow \dots$$

where $s_i \in S$, $a_i \in A$ such that $R^{a_i}(s_i, s_{i+1}) > 0$, and $t_i \in \mathbb{R}_{\geq 0}$. It represents a possible behaviour of the CTMDP. We denote the set of paths by $Path$ and the set of maximal paths by $Path^{\max}$. (Maximal paths are paths that are infinite or that end in an absorbing state.)

A *cylinder set* is a set of maximal paths of the following form:

$$C(s_0, a_0, I_0, s_1, a_1, I_1, \dots, a_{n-1}, I_{n-1}, s_n) := \left\{ s_0 \xrightarrow[a_0]{t_0} s_1 \xrightarrow[a_1]{t_1} \dots \xrightarrow[a_{n-1}]{t_{n-1}} s_n \rightarrow \dots \mid t_i \in I_i \text{ for all } i < n \right\}$$

where I_i are intervals in $\mathbb{R}_{\geq 0}$.

The Probability Space of Paths. A *scheduler* is a function that resolves the nondeterministic choices in a CTMDP. A CTMDP together with a measurable scheduler induces a fully probabilistic process, where a probability space can be defined.

A scheduler in principle bases its choices on the history of the system. Restricted scheduler classes only check a part of the history. The most powerful schedulers, history-dependent schedulers, are functions from non-maximal paths to actions $D : (Path \setminus Path^{\max}) \rightarrow A$. See [7] for details. However, in the rest of this article, we will mostly look at (time-abstract) *positional* schedulers, which base the choice on the current state only. They are functions $D : S \rightarrow A$.

A CTMDP (S, A, R, AP, L) and a positional scheduler D together induce a CTMC $(S, \{*\}, \tilde{R}, AP, L)$. Its transition relation is defined by $\tilde{R}^*(s, s') = R^{D(s)}(s, s')$. From this, one can define a probability space $(Path^{\max}, \mathcal{C}, \Pr_s^D)$. Here, \mathcal{C} is the smallest σ -algebra that contains all cylinder sets. Its probability measure is the unique measure induced by:

$$\begin{aligned} \Pr_{s_0}^D(C(s_0)) &:= 1 \\ \Pr_{s_0}^D(C(s_0, D(s_0), I_0, s_1, a_1, I_1, \dots, a_n, I_n, s_{n+1})) &:= \\ &:= \Pr\left(s_0 \xrightarrow[D(s_0)]{\in I_0} s_1 \mid D(s_0)\right) \cdot \Pr_{s_1}^D(C(s_1, a_1, I_1, \dots, a_n, I_n, s_{n+1})). \end{aligned}$$

Lemma 3. *The function defined above is a measure.*

Proof. This lemma has been shown in [7, Theorem 2]. □

A CTMDP \mathcal{M} is *uniform* if all exit rates are equal, i. e. for all states s and actions a , $E_{\mathcal{M}}^a(s) = E_{\mathcal{M}}$. For a CTMDP \mathcal{M} , we can find a uniform CTMDP having a similar probability distribution as follows.

We first define the CTMDP $\mathcal{M}_{\text{unif}} = (S, A, \tilde{R}, AP, L)$ which is almost the same as \mathcal{M} , except that we add to each state a self-loop such that the total exit rate becomes $E \geq E_{\mathcal{M}}$ for every state and action:

$$\tilde{R}^a(s, s') := \begin{cases} R^a(s, s') & \text{if } s \neq s' \\ E - \sum_{t \neq s} R^a(s, t) & \text{if } s = s' \end{cases} .$$

For a uniform (or uniformised) CTMDP, we can define its *embedded* discrete-time MDP as follows: $\mathcal{P} = \mathcal{P}_{\mathcal{M}}$ is given by (S, A, P, AP, L) , where the transition probabilities $P^a(s, s')$ are given by $P^a(s, s') = R^a(s, s')/E_{\mathcal{M}}$.

Lemma 4. *Positional schedulers are invariant under uniformisation, i. e. for every scheduler on \mathcal{M} there is one on $\mathcal{M}_{\text{unif}}$ and vice versa, such that the probability measures of the respective induced CTMCs are equal up to stuttering in the same state.*

Proof. As the $\mathcal{M}_{\text{unif}}$ -scheduler corresponding to D on \mathcal{M} , we choose D again. This is possible as the state space and the set of actions do not change. Note that the CTMC induced by D on $\mathcal{M}_{\text{unif}}$ only differs from the CTMC induced by D on \mathcal{M} by self-loops. Weak bisimulation on CTMCs [3a] ignores self-loops, so the two are weakly bisimilar. \square

3 The Logic DCTL

DCTL consists of state formulas, denoted φ, ψ , and path formulas, denoted Ψ . The syntax of state and path formulas is given by the following grammar:

$$\begin{aligned}\varphi &::= 1 \mid p \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \oplus_w \psi \mid \exists\Psi \mid \forall\Psi \\ \Psi &::= \diamond_{\alpha}\varphi \mid \square_{\alpha}\varphi \mid \Delta_{\alpha}\varphi\end{aligned}$$

where p is an atomic proposition, $w \in [0, 1]$ is a weight, and $\alpha \in \mathbb{R}_{>0}$ is a discount rate. We call $\diamond_{\alpha}\varphi$ the *discounted maximum*, $\square_{\alpha}\varphi$ the *discounted minimum*, and $\Delta_{\alpha}\varphi$ the *discounted average* over paths.

3.1 Semantics

The semantics of a DCTL state formula φ is a mapping $\llbracket\varphi\rrbracket : S \rightarrow [0, 1]$ from states to utilities. It can be defined by:

1	true	$\llbracket 1 \rrbracket (s) = 1$
p	atomic proposition, $p \in AP$	$\llbracket p \rrbracket (s) = L(s, p)$
$\neg\varphi$	negation	$\llbracket \neg\varphi \rrbracket (s) = 1 - \llbracket \varphi \rrbracket (s)$
$\varphi \wedge \psi$	conjunction	$\llbracket \varphi \wedge \psi \rrbracket (s) = \llbracket \varphi \rrbracket (s) \sqcap \llbracket \psi \rrbracket (s)$
$\varphi \oplus_w \psi$	weighted sum, $w \in [0, 1]$	$\llbracket \varphi \oplus_w \psi \rrbracket (s) = (1 - w) \llbracket \varphi \rrbracket (s) + w \llbracket \psi \rrbracket (s)$

The interpretation of $\exists\Psi$ and $\forall\Psi$ is closely linked to the interpretation of path formulas. Informally, the semantics of $\exists\Psi$ is the expected utility of Ψ under the *best* scheduler and $\forall\Psi$ is its expected utility under the *worst* scheduler. $\diamond_{\alpha}\varphi$ is the maximum of all φ -values along the path, where we apply a discount: if a value appears after time t , it is reduced with the factor $e^{-\alpha t}$, with the result that earlier states can more easily influence the maximum than later ones. $\square_{\alpha}\varphi$ is the minimum of all discounted φ -values along the path, and $\Delta_{\alpha}\varphi$ is the average over all φ -values, where the value after time t now gets relative weight $e^{-\alpha t}$, in accordance with the idea that earlier states are more important than later ones.

There are two possibilities to formalise the semantics of path formulas: either as the *fixpoint* of an operator, or via a look at the *complete path* at once. In the following sections, we will give the detailed definition of the two semantics.

$$\begin{aligned}
\llbracket \exists \diamond_{\alpha} \varphi \rrbracket^f(s) &= \mu u. \llbracket \varphi \rrbracket^f(s) \sqcup \max_{a \in A} \frac{1}{E^a(s) + \alpha} \sum_{s' \in S} R^a(s, s') u(s') \\
\llbracket \forall \diamond_{\alpha} \varphi \rrbracket^f(s) &= \mu u. \llbracket \varphi \rrbracket^f(s) \sqcup \min_{a \in A} \frac{1}{E^a(s) + \alpha} \sum_{s' \in S} R^a(s, s') u(s') \\
\llbracket \exists \square_{\alpha} \varphi \rrbracket^f(s) &= \mu u. \llbracket \varphi \rrbracket^f(s) \sqcap \max_{a \in A} \frac{\alpha}{E^a(s) + \alpha} + \frac{1}{E^a(s) + \alpha} \sum_{s' \in S} R^a(s, s') u(s') \\
\llbracket \forall \square_{\alpha} \varphi \rrbracket^f(s) &= \mu u. \llbracket \varphi \rrbracket^f(s) \sqcap \min_{a \in A} \frac{\alpha}{E^a(s) + \alpha} + \frac{1}{E^a(s) + \alpha} \sum_{s' \in S} R^a(s, s') u(s') \\
\llbracket \exists \triangle_{\alpha} \varphi \rrbracket^f(s) &= \mu u. \max_{a \in A} \frac{\alpha}{E^a(s) + \alpha} \llbracket \varphi \rrbracket^f(s) + \frac{1}{E^a(s) + \alpha} \sum_{s' \in S} R^a(s, s') u(s') \\
\llbracket \forall \triangle_{\alpha} \varphi \rrbracket^f(s) &= \mu u. \min_{a \in A} \frac{\alpha}{E^a(s) + \alpha} \llbracket \varphi \rrbracket^f(s) + \frac{1}{E^a(s) + \alpha} \sum_{s' \in S} R^a(s, s') u(s')
\end{aligned}$$

Fig. 2. Fixpoint semantics for CTMDPs

3.2 Fixpoint Semantics

The fixpoint semantics arises by lifting the classical connection between CTL and the μ -calculus to a quantitative setting. For a transition system, we denote by $\exists Pre(u)$ the set of all states that have a transition into the set u . Then the set $\llbracket \exists \diamond \varphi \rrbracket$ of all states satisfying $\exists \diamond \varphi$ is the smallest fixpoint of the equation $u = \llbracket \varphi \rrbracket \cup \exists Pre(u)$, written as $\mu u. \llbracket \varphi \rrbracket \cup \exists Pre(u)$.

As explained in [1], one can lift these fixpoint equations to a quantitative setting by interpreting \cup as the pointwise maximum and $\exists Pre(u)$ as the expected value of u , achievable in one step, under the best scheduler. Then, the semantics of $\exists \diamond_{\alpha} \varphi$ is the greater of the current utility of φ and the expected utility after one transition. When we formalise this idea, we get the equations in Fig. 2 for the fixpoint semantics. Some remarks on them are in order.

Discounted Maximum Operator. For a CTMC \mathcal{M} , the discounted semantics of $\exists \diamond_{\alpha} \varphi$ is obtained as follows. Let α be a discount rate and $u : S \rightarrow [0, 1]$ be a state utility function. Then $\exists Pre_{\alpha}^{\mathcal{M}}(u) : S \rightarrow [0, 1]$ yields at state s the expected discounted utility immediately after the first jump. That is, if we move to s' at time t , we get $e^{-\alpha t} u(s')$. Thus, $\exists Pre_{\alpha}^{\mathcal{M}}(u)(s)$ is the expected value of the random variable $e^{-\alpha T} u(X)$, where T is the random variable denoting the time of the first jump and X the random variable denoting the state reached by the first jump.

For CTMDPs, we additionally maximize over all actions $a \in A$, i. e. we choose the action with the maximal expected discounted utility. Let X_0 be the current state, and A_0 the action taken in this state. A simple calculation shows that

$$\begin{aligned}
\exists Pre_{\alpha}^{\mathcal{M}}(u)(s) &= \max_{a \in A} \mathbb{E}[e^{-\alpha T} u(X) | X_0 = s, A_0 = a] \\
&= \max_{a \in A} \frac{1}{E^a(s) + \alpha} \sum_{s' \in S} R^a(s, s') u(s').
\end{aligned}$$

Inserting these results in $\llbracket \exists \diamond_{\alpha} \varphi \rrbracket^f = \mu u. \llbracket \varphi \rrbracket^f \sqcup \exists Pre_{\alpha}^{\mathcal{M}}(u)$ and $\llbracket \forall \diamond_{\alpha} \varphi \rrbracket^f = \mu u. \llbracket \varphi \rrbracket^f \sqcup \forall Pre_{\alpha}^{\mathcal{M}}(u)$ yields the equations in Fig. 2.

Discounted Minimum Operator. The operator $\forall \square_{\alpha} \varphi$ should be equivalent to $\neg \exists \diamond_{\alpha} \neg \varphi$. Therefore, we derive the semantics as follows:

$$\llbracket \forall \square_{\alpha} \varphi \rrbracket^f(s) = 1 - \mu u. (1 - \llbracket \varphi \rrbracket^f(s)) \sqcup \max_{a \in A} \mathbb{E}[e^{-\alpha T} u(X) | X_0 = s, A_0 = a]$$

$$= \nu \hat{u}. \llbracket \varphi \rrbracket^f(s) \sqcap \underbrace{\min_{a \in A} \mathbb{E}[1 - e^{-\alpha T}(1 - \hat{u}(X)) | X_0 = s, A_0 = a]}_{=: \widehat{Pre}_\alpha^{\mathcal{M}}(\hat{u})}.$$

Then,

$$\widehat{Pre}_\alpha^{\mathcal{M}}(\hat{u})(s) = \min_{a \in A} \frac{\alpha}{E^a(s) + \alpha} + \frac{1}{E^a(s) + \alpha} \sum_{s' \in S} R^a(s, s') \hat{u}(s').$$

We will see shortly that the fixpoint is unique. Therefore, we can write μ in Fig. 2 instead of ν .

Discounted Average Operator. To calculate $\llbracket \exists \Delta_\alpha \varphi \rrbracket^f$, we have to take an average between $\llbracket \varphi \rrbracket^f$ and the utility after one step, giving more weight to the near future than to the far one. The natural candidate to choose the weight is the pdf of the probability to lose patience, $t \mapsto \alpha e^{-\alpha t}$. This leads to the definition:

$$\begin{aligned} \llbracket \exists \Delta_\alpha \varphi \rrbracket_{\mathcal{M}}^f(s) = & \mu u. \max_{a \in A} \sum_{s' \in S} \int_0^\infty \text{pdf}\left(s \xrightarrow{t} s' \mid a\right) \cdot \\ & \cdot \left[\int_0^t \alpha e^{-\alpha \tau} d\tau \llbracket \varphi \rrbracket_{\mathcal{M}}^f(s) + \int_t^\infty \alpha e^{-\alpha \tau} d\tau u(s') \right] dt \end{aligned}$$

Properties. The fixpoints are unique because the respective functions are contractions:

Lemma 5. *The functions under the fixpoint operators are contractions according to the L^∞ -norm.*

Proof. Let's look at the function

$$\mathcal{D} : u \mapsto \llbracket \varphi \rrbracket_{\mathcal{M}}^f \sqcup \exists Pre_\alpha^{\mathcal{M}}(u) = \llbracket \varphi \rrbracket_{\mathcal{M}}^f(\cdot) \sqcup \max_{a \in A} \frac{1}{E^a + \alpha} \sum_{s' \in S} R^a(\cdot, s') u(s'),$$

used to define $\llbracket \exists \diamond_\alpha \varphi \rrbracket_{\mathcal{M}}^f = \mu u. \mathcal{D}(u)$. We have to show that this function is a contraction, i. e. there exists an $\varepsilon > 0$ such that, given $u, v : S \rightarrow [0, 1]$, we get

$$\|\mathcal{D}(u) - \mathcal{D}(v)\|_\infty \leq (1 - \varepsilon) \|u - v\|_\infty. \quad (0a)$$

We first let s be a state such that $\|\mathcal{D}(u) - \mathcal{D}(v)\|_\infty = |\mathcal{D}(u)(s) - \mathcal{D}(v)(s)|$. We assume w. l. o. g. that $\mathcal{D}(u)(s) \geq \mathcal{D}(v)(s)$.

Then, distinguish on which side of the \sqcup is chosen. Obviously, if $\mathcal{D}(u)(s) = \llbracket \varphi \rrbracket_{\mathcal{M}}^f(s) = \mathcal{D}(v)(s)$, then (0a) holds for any ε . If $\mathcal{D}(u) = \exists Pre_\alpha^{\mathcal{M}}(u) > \mathcal{D}(v) = \llbracket \varphi \rrbracket_{\mathcal{M}}^f(s)$, the difference $|\mathcal{D}(u)(s) - \mathcal{D}(v)(s)|$ is never larger than $|\exists Pre_\alpha^{\mathcal{M}}(u)(s) - \exists Pre_\alpha^{\mathcal{M}}(v)(s)|$, so it is enough to show (0a) for the case that both choose $\exists Pre_\alpha^{\mathcal{M}}$.

Let $\varepsilon := \min_{a \in A, s \in S} \alpha / (E^a(s) + \alpha)$. Now, for any action $a \in A$, we have

$$\begin{aligned} \left| \frac{1}{E^a(s) + \alpha} \sum_{s' \in S} R^a(s, s') u(s') - \frac{1}{E^a(s) + \alpha} \sum_{s' \in S} R^a(s, s') v(s') \right| & \leq \\ & \leq \frac{1}{E^a(s) + \alpha} \sum_{s' \in S} R^a(s, s') \|u - v\|_\infty = \frac{E^a(s)}{E^a(s) + \alpha} \|u - v\|_\infty = \\ & = \left(1 - \frac{\alpha}{E^a(s) + \alpha} \right) \|u - v\|_\infty \leq (1 - \varepsilon) \|u - v\|_\infty. \quad (0b) \end{aligned}$$

Then let a_u be the action that maximizes $\frac{1}{E^\alpha(s) + \alpha} \sum_{s' \in S} R^\alpha(s, s')u(s')$ and define a_v similarly. If $a_u = a_v$, then (0a) follows from (0b) directly. Otherwise, we know that

$$\begin{aligned} \mathcal{D}(u)(s) &\geq \mathcal{D}(v)(s) = \frac{1}{E^{a_v}(s) + \alpha} \sum_{s' \in S} R^{a_v}(s, s')v(s') \\ &\geq \frac{1}{E^{a_u}(s) + \alpha} \sum_{s' \in S} R^{a_u}(s, s')v(s') \\ &\stackrel{\text{by (0b)}}{\geq} \frac{1}{E^{a_u}(s) + \alpha} \sum_{s' \in S} R^{a_u}(s, s')u(s') - (1 - \varepsilon)\|u - v\|_\infty \\ &= \mathcal{D}(u)(s) - (1 - \varepsilon)\|u - v\|_\infty \end{aligned}$$

and therefore, $\|\mathcal{D}(u) - \mathcal{D}(v)\|_\infty \leq (1 - \varepsilon)\|u - v\|_\infty$ in all cases.

The proofs for other operators are similar. \square

Proposition 6 (Duality Laws). *For all DCTL formulas φ and all CTMDPs \mathcal{M} , we have*

$$\begin{aligned} \llbracket \neg \exists \diamond_\alpha \varphi \rrbracket_{\mathcal{M}}^f &= \llbracket \forall \square_\alpha \neg \varphi \rrbracket_{\mathcal{M}}^f & \llbracket \neg \exists \square_\alpha \varphi \rrbracket_{\mathcal{M}}^f &= \llbracket \forall \diamond_\alpha \neg \varphi \rrbracket_{\mathcal{M}}^f \\ \llbracket \neg \exists \triangle_\alpha \varphi \rrbracket_{\mathcal{M}}^f &= \llbracket \forall \triangle_\alpha \neg \varphi \rrbracket_{\mathcal{M}}^f \end{aligned}$$

Proof. The operators were defined in a way that implies the duality laws. \square

Theorem 7. *Let \mathcal{M} be a CTMDP and $\mathcal{M}_{\text{unif}}$ be its uniformization. Then we have for all DCTL formulas φ that $\llbracket \varphi \rrbracket_{\mathcal{M}}^f = \llbracket \varphi \rrbracket_{\mathcal{M}_{\text{unif}}}^f$.*

Proof. The probability to be in a specific state at any given time, under a given scheduler, is the same for \mathcal{M} and $\mathcal{M}_{\text{unif}}$ (Lemma 4). Only the number of times a state is reentered changes, and so also the number of times that an action can be chosen. However, a positional scheduler has to choose the same action whenever the same state is reentered, so the scheduler cannot make a different choice in $\mathcal{M}_{\text{unif}}$ than in \mathcal{M} . \square

We will see later that so-called late schedulers, that choose the action to take not when entering, but when leaving a state, can deliver better results (higher values for $\exists \diamond_\alpha$, lower values for $\forall \diamond_\alpha$) than positional schedulers. Late schedulers may decide the action not only on the current state, but also on the sojourn time in this state. A scheduler that can distinguish between entering a state for the first time and reentering a state can get an *estimate* of the sojourn time and improve over a positional scheduler based on this estimate. This was a problem for model checking of non-uniform CTMDPs in [3].

3.3 Path Semantics

Another way to define a CTL semantics is to look at the set of (maximal) paths. A path satisfies $\diamond \varphi$ if it contains at least one φ -state; a state s satisfies $\exists \diamond \varphi$ if the set of paths starting in s contains some path satisfying $\diamond \varphi$. If we lift this principle to a quantitative setting, we reinterpret “at least one φ -state” as “the maximum of $\llbracket \varphi \rrbracket$ ”, and we reinterpret “existence of a path” as “supremum over all paths”. This leads to $\llbracket \exists \diamond \varphi \rrbracket = \sup_{\sigma \in \text{Path}} \max_{s \in \sigma} \llbracket \varphi \rrbracket(s)$. In a Markov decision process, one would additionally consider the probabilities of paths.

In the path semantics of DCTL, we give separate semantics to the operators \exists and \diamond_α . Let us first look at \exists and \forall : Assume given a class of schedulers \mathfrak{D} . Then, we define:

$$\begin{aligned} \llbracket \exists \Psi \rrbracket_{\mathcal{M}}^p(s) &= \sup_{D \in \mathfrak{D}} \mathbb{E}_{\sigma \in C(s)}^D \llbracket \Psi \rrbracket_{\mathcal{M}}^p(\sigma) \\ \llbracket \forall \Psi \rrbracket_{\mathcal{M}}^p(s) &= \inf_{D \in \mathfrak{D}} \mathbb{E}_{\sigma \in C(s)}^D \llbracket \Psi \rrbracket_{\mathcal{M}}^p(\sigma) \end{aligned}$$

The expected value \mathbb{E} in these formulas is of course taken over the probability space of paths, under the mentioned scheduler.

The semantics of path formulas is:

$$\llbracket \Diamond_\alpha \varphi \rrbracket_{\mathcal{M}}^P(\sigma) = \max_{t \geq 0} e^{-\alpha t} \llbracket \varphi \rrbracket_{\mathcal{M}}^P(\sigma @ t) \quad (1)$$

$$\llbracket \Box_\alpha \varphi \rrbracket_{\mathcal{M}}^P(\sigma) = \min_{t \geq 0} 1 - e^{-\alpha t} (1 - \llbracket \varphi \rrbracket_{\mathcal{M}}^P(\sigma @ t)) \quad (2)$$

$$\llbracket \Delta_\alpha \varphi \rrbracket_{\mathcal{M}}^P(\sigma) = \int_0^\infty \alpha e^{-\alpha t} \llbracket \varphi \rrbracket_{\mathcal{M}}^P(\sigma @ t) dt$$

Existence of Maximum. In an undiscounted setting and with a finite LTS, $\llbracket \varphi \rrbracket$ can only take a finite number of values on any path, so it is clear that $\llbracket \Diamond \varphi \rrbracket$ is not only a supremum, but really a maximum. However, with continuous discounting, we have to prove existence of the maximum.

Lemma 8. *The discounted maximum and minimum are well-defined, i. e., the maximum (1) and minimum (2) actually do exist.*

Proof. Assume given a path σ . For a fixed state s_0 , the function $t \mapsto e^{-\alpha t} \llbracket \varphi \rrbracket^P(s_0)$ is continuous and strictly falling. The set $T_{s_0} := \{t \in \mathbb{R}_{\geq 0} \mid \sigma @ t = s_0\}$ consists of left-closed intervals. Therefore, $m_{s_0} := \max_{t \in T_{s_0}} e^{-\alpha t} \llbracket \varphi \rrbracket^P(s_0)$ is well-defined. Note that $\bigcup_{s \in S} T_s = \mathbb{R}_{\geq 0}$. Now,

$$\llbracket \Diamond_\alpha \varphi \rrbracket^P(\sigma) = \sup_{t \geq 0} e^{-\alpha t} \llbracket \varphi \rrbracket^P(\sigma @ t) = \sup_{s \in S} \sup_{t \in T_s} e^{-\alpha t} \llbracket \varphi \rrbracket^P(s) = \sup_{s \in S} m_s$$

is the supremum of a finite set, and therefore its maximum. \square

Example 9. Let us calculate $\llbracket \exists \Diamond_2 \text{black} \rrbracket^P(s)$ in the Markov chain of Fig. 1. As it does not contain nondeterminism, there is only the trivial scheduler. There are two time-abstract maximal paths in this Markov chain: $s \rightarrow u$ and $s \rightarrow t \rightarrow u$. We calculate $\mathbb{E}_{\sigma \in C(s)} \llbracket \Diamond_2 \text{black} \rrbracket^P(\sigma)$ by summing over these two paths separately. For the path $s \rightarrow u$,

$$\begin{aligned} \mathbb{E}_{\sigma=(s \rightarrow u)} \llbracket \Diamond_2 \text{black} \rrbracket^P(\sigma) &= \int_0^\infty \llbracket \Diamond_2 \text{black} \rrbracket^P\left(s \xrightarrow{t} u\right) \text{pdf}\left(s \xrightarrow{t} u\right) dt \\ &= \int_0^\infty \max_{\tau \geq 0} e^{-2\tau} \llbracket \text{black} \rrbracket^P\left(\left[s \xrightarrow{t} u\right] @ \tau\right) R(s, u) e^{-E(s)t} dt \\ &= \int_0^\infty \max\left\{e^{-2t} \frac{9}{10}, \frac{1}{10}\right\} e^{-4t} dt \\ &\stackrel{*}{=} \int_0^{\ln 3} e^{-2t} \frac{9}{10} e^{-4t} dt + \int_{\ln 3}^\infty \frac{1}{10} e^{-4t} dt \\ &= \frac{9}{10} \frac{1 - e^{-6 \ln 3}}{6} + \frac{1}{10} \frac{e^{-4 \ln 3}}{4} = \frac{9(1 - 3^{-6})}{10 \cdot 6} + \frac{1 \cdot 3^{-4}}{10 \cdot 4} \\ &= \frac{1459}{9720} \approx 0.15 \end{aligned}$$

The equality $\stackrel{*}{=}$ holds because

$$\frac{9}{10} e^{-2t} \geq \frac{1}{10} \quad \Leftrightarrow \quad e^{-2t} \geq \frac{1}{9} \quad \Leftrightarrow \quad -2t \geq \ln \frac{1}{9} = -\ln 9 \quad \Leftrightarrow \quad t \leq \frac{1}{2} \ln 9 = \ln 3$$

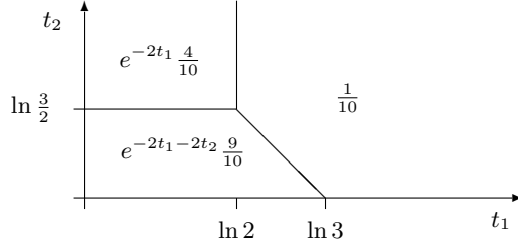


Fig. 2a. Value of $\max \left\{ \frac{1}{10}, e^{-2t_1} \frac{4}{10}, e^{-2t_1-2t_2} \frac{9}{10} \right\}$. Note that $\ln \frac{3}{2} = \ln 3 - \ln 2$.

Similarly, for the path $s \rightarrow t \rightarrow u$, we have that

$$\begin{aligned}
\mathbb{E}_{\sigma=(s \rightarrow t \rightarrow u)} [\diamond_2 \text{black}]^P(\sigma) &= \int_0^\infty \int_0^\infty [\diamond_2 \text{black}]^P \left(s \xrightarrow{t_1} t \xrightarrow{t_2} u \right) \text{pdf} \left(s \xrightarrow{t_1} t \xrightarrow{t_2} u \right) dt_1 dt_2 \\
&= \int_0^\infty \int_0^\infty \max_{\tau \geq 0} e^{-2\tau} [\text{black}]^P \left(\left[s \xrightarrow{t_1} t \xrightarrow{t_2} u \right] @\tau \right) \frac{3}{4} 4e^{-4t_1} 3e^{-3t_2} dt_1 dt_2 = \\
&= \int_0^\infty \int_0^\infty \max \left\{ \frac{1}{10}, e^{-2t_1} \frac{4}{10}, e^{-2(t_1+t_2)} \frac{9}{10} \right\} R(s, t) e^{-E(s)t_1} R(t, u) e^{-E(t)t_2} dt_1 dt_2
\end{aligned}$$

Which of the terms is the maximum, depending on t_1 and t_2 , is illustrated in figure 2a.

$$\begin{aligned}
&= \int_0^{\ln \frac{3}{2}} \int_0^{\ln 3 - t_2} e^{-2t_1-2t_2} \frac{9}{10} 9e^{-4t_1-3t_2} dt_1 dt_2 + \int_0^{\ln \frac{3}{2}} \int_{(\ln 3)-t_2}^\infty \frac{1}{10} 9e^{-4t_1-3t_2} dt_1 dt_2 + \\
&\quad + \int_{\ln \frac{3}{2}}^\infty \int_0^{\ln 2} e^{-2t_1} \frac{4}{10} 9e^{-4t_1-3t_2} dt_1 dt_2 + \int_{\ln \frac{3}{2}}^\infty \int_{\ln 2}^\infty \frac{1}{10} 9e^{-4t_1-3t_2} dt_1 dt_2 = \\
&= \frac{81}{10} \int_0^{\ln \frac{3}{2}} e^{-5t_2} \frac{1}{6} (1 - e^{6t_2-6\ln 3}) dt_2 + \frac{9}{10} \int_0^{\ln \frac{3}{2}} e^{-3t_2} \frac{1}{4} e^{4t_2-4\ln 3} dt_2 + \\
&\quad + \frac{36}{10} \int_{\ln \frac{3}{2}}^\infty e^{-3t_2} \frac{1}{6} (1 - e^{-6\ln 2}) dt_2 + \frac{9}{10} \int_{\ln \frac{3}{2}}^\infty e^{-3t_2} \frac{1}{4} e^{-4\ln 2} dt_2 = \\
&= \frac{27}{20} \left[\frac{1}{5} (1 - e^{-5\ln \frac{3}{2}}) - 3^{-6} (e^{\ln \frac{3}{2}} - 1) \right] + \frac{9}{40} 3^{-4} (e^{\ln \frac{3}{2}} - 1) + \\
&\quad + \frac{6}{10} (1 - 2^{-6}) \frac{1}{3} e^{-3\ln \frac{3}{2}} + \frac{9}{40} 2^{-4} \frac{1}{3} e^{-3\ln \frac{3}{2}} = \\
&= \frac{27}{100} - \frac{8}{225} - \frac{1}{1080} + \frac{1}{720} + \frac{16}{270} - \frac{1}{1080} + \frac{1}{720} = \frac{1591}{5400} \approx 0.29
\end{aligned}$$

So, in the end,

$$[\exists \diamond_2 \text{black}]^P(s) = \frac{1459}{9720} + \frac{1591}{5400} = \frac{10807}{24300} \approx 0.44.$$

Lemma 10.

$$[\diamond_\alpha \varphi]_{\mathcal{M}}^P(\sigma) = \max_{t \geq 0} [0 \oplus_{e^{-\alpha t}} \varphi]_{\mathcal{M}}^P(\sigma @ t)$$

$$[\square_\alpha \varphi]_{\mathcal{M}}^P(\sigma) = \min_{t \geq 0} [1 \oplus_{e^{-\alpha t}} \varphi]_{\mathcal{M}}^P(\sigma @ t)$$

Proposition 11. *The following dualities between \diamond_α and \square_α hold. Δ_α is its own dual.*

$$\begin{aligned} \llbracket \exists \diamond_\alpha \neg \varphi \rrbracket^P &= \llbracket \neg \forall \square_\alpha \varphi \rrbracket^P & \llbracket \forall \diamond_\alpha \neg \varphi \rrbracket^P &= \llbracket \neg \exists \square_\alpha \varphi \rrbracket^P \\ \llbracket \exists \Delta_\alpha \neg \varphi \rrbracket^P &= \llbracket \neg \forall \Delta_\alpha \varphi \rrbracket^P \end{aligned}$$

Proposition 12. *The fixpoint and path semantics of Δ_α coincide.*

Proof. The proof of this property exactly follows the lines of [1, Thm. 2]. □

4 Model Checking DCTL

The model checking problem for DCTL consists in computing the value of $\llbracket \varphi \rrbracket_{\mathcal{M}}^f(s)$, given a formula φ and a state s in a CTMDP \mathcal{M} . Based on the equations found in the previous section, we propose the following algorithms to solve this problem.

Finding the utility of the formulas 1 and p (for $p \in AP$) is trivial. Checking the formulas $\neg \varphi$, $\varphi \wedge \psi$, and $\varphi \oplus_w \psi$ is easy and straightforward, if we have already found the truth values of the subformulas. We will not treat them in the rest of the article, but concentrate on the operators $\exists \diamond_\alpha$, $\exists \square_\alpha$ and $\exists \Delta_\alpha$. The duality laws (Props. 6 and 11) can be used to check the operators $\forall \diamond_\alpha$, $\forall \square_\alpha$ and $\forall \Delta_\alpha$.

4.1 Model Checking the Fixpoint Semantics

Reduction to DCTL Semantics for MDPs. The fixpoint equations in Fig. 2, expressing the semantics for DCTL over CTMDPs, are very similar to the fixpoint equations for discrete-time MDPs. For example, in the discrete-time case, we have

$$\llbracket \exists \square_\rho \varphi \rrbracket_{\mathcal{P}}^f(s) = \mu u. \llbracket \varphi \rrbracket_{\mathcal{P}}^f(s) \sqcap \left[(1 - \rho) + \rho \max_{a \in A} \sum_{s' \in S} P^a(s, s') u(s') \right]$$

In fact, they are so analogous that the DCTL model checking problem for CTMDPs reduces to the DCTL model checking problem for MDPs: We can model check a DCTL formula φ over a uniform CTMDP \mathcal{M} by model checking a formula φ' over the embedded Markov decision process. Here φ' arises from φ by changing the discount factors \diamond_α etc. to \diamond_ρ in φ , where $\rho = E_{\mathcal{M}} / (E_{\mathcal{M}} + \alpha)$. Note that, as the fixpoint semantics only uses positional schedulers, it is possible to analyse other CTMDPs after uniformising them.

Theorem 13. *Let \mathcal{M} be a CTMDP, let \mathcal{P} be its embedded discrete-time MDP (possibly after uniformisation) and let $\rho = E_{\mathcal{M}} / (E_{\mathcal{M}} + \alpha)$. Then we have for all DCTL formulas φ that*

$$\begin{aligned} \llbracket \exists \diamond_\alpha \varphi \rrbracket_{\mathcal{M}}^f &= \llbracket \exists \diamond_\rho \varphi' \rrbracket_{\mathcal{P}}^f & \llbracket \exists \square_\alpha \varphi \rrbracket_{\mathcal{M}}^f &= \llbracket \exists \square_\rho \varphi' \rrbracket_{\mathcal{P}}^f & \llbracket \exists \Delta_\alpha \varphi \rrbracket_{\mathcal{M}}^f &= \llbracket \exists \Delta_\rho \varphi' \rrbracket_{\mathcal{P}}^f \\ \llbracket \forall \diamond_\alpha \varphi \rrbracket_{\mathcal{M}}^f &= \llbracket \forall \diamond_\rho \varphi' \rrbracket_{\mathcal{P}}^f & \llbracket \forall \square_\alpha \varphi \rrbracket_{\mathcal{M}}^f &= \llbracket \forall \square_\rho \varphi' \rrbracket_{\mathcal{P}}^f & \llbracket \forall \Delta_\alpha \varphi \rrbracket_{\mathcal{M}}^f &= \llbracket \forall \Delta_\rho \varphi' \rrbracket_{\mathcal{P}}^f \end{aligned}$$

Since the reduction above is clearly linear in the size of \mathcal{M} , the complexity of model checking the DCTL fixpoint semantics is the same for CTMDPs and discrete-time MDPs. Thus, we obtain the following corollary from [1, Thm. 7].

Corollary 14. *The problem of model checking $\llbracket \varphi \rrbracket_{\mathcal{M}}^f$ can be solved in nondeterministic polynomial time in the size of \mathcal{M} and exponential in $|\varphi|$.*

4.2 Model Checking the Path Semantics

The article [1] proposes a model checking procedure for discrete-time Markov chains of $\exists \diamond_\alpha \varphi$ that can be adapted to continuous-time Markov chains. (Note that we assume there is no non-determinism here.) For any path σ , the value $\llbracket \diamond_\alpha \varphi \rrbracket^P(\sigma)$ is reached at a time when σ enters a state that is better than $\sigma @ 0$ (i. e., $e^{-\alpha t} \llbracket \varphi \rrbracket^P(\sigma @ t) > \llbracket \varphi \rrbracket^P(\sigma @ 0)$ for some t). We therefore order the states according to their value $\llbracket \varphi \rrbracket^P$:

$$\llbracket \varphi \rrbracket^P(s_1) \geq \llbracket \varphi \rrbracket^P(s_2) \geq \dots \geq \llbracket \varphi \rrbracket^P(s_n)$$

In state s_1 , no other state can improve $\llbracket \varphi \rrbracket^P$, so we know that $\llbracket \exists \diamond_\alpha \varphi \rrbracket^P(s_1) = \llbracket \varphi \rrbracket^P(s_1)$.

In state s_2 , the only possible improvement is to reach s_1 at some time t with $\llbracket \varphi \rrbracket^P(s_2) < \llbracket \varphi \rrbracket^P(s_1)e^{-\alpha t} = \llbracket \exists \diamond_\alpha \varphi \rrbracket^P(s_1)e^{-\alpha t}$. We calculate the probability that s_1 is reached early enough and the resulting discounted value of $\llbracket \exists \diamond_\alpha \varphi \rrbracket^P(s_2)$ via a variant of time-bounded reachability.

In general, if a path starts in s_i , improvements occur if some state s_j is reached at a time t such that $\llbracket \varphi \rrbracket^P(s_i) < \llbracket \exists \diamond_\alpha \varphi \rrbracket^P(s_j)e^{-\alpha t}$. We can base our algorithm on the observation that *after the first such state s_j , no more improvement is possible*: any further improvement is already taken into account by $\llbracket \exists \diamond_\alpha \varphi \rrbracket^P(s_j)$. The second idea used in the algorithm is that of *cutoff time*: After some time, the discount factor $e^{-\alpha t}$ is so small that reaching s_{i-1} will no longer improve over s_i . From that moment on, we could as well presume that any state s_j (for $j = i, \dots, n$) had $\llbracket \varphi \rrbracket^P(s_j) = \llbracket \varphi \rrbracket^P(s_{i-1})$. Therefore, it is enough to regard the current path until the cutoff time, after which we can continue analysis in a simplified Markov chain.

This allows to use a sequence of CTMCs $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_n$ where \mathcal{M}_i has the same structure as \mathcal{M} but change $\llbracket \varphi \rrbracket^P$ to:

$$\llbracket \varphi \rrbracket_{\mathcal{M}_i}^P(s_j) := \llbracket \varphi \rrbracket_{\mathcal{M}}^P(s_{\min\{i,j\}}).$$

Note that one can easily prove that $\llbracket \exists \diamond_\alpha \varphi \rrbracket_{\mathcal{M}_i}^P(s_j) = \llbracket \exists \diamond_\alpha \varphi \rrbracket_{\mathcal{M}}^P(s_j)$ if $j \leq i$. We now calculate $\llbracket \exists \diamond_\alpha \varphi \rrbracket_{\mathcal{M}_i}^P$ recursively:

Base case. Trivial: $\llbracket \exists \diamond_\alpha \varphi \rrbracket_{\mathcal{M}_1}^P(s_i) = \llbracket \varphi \rrbracket_{\mathcal{M}}^P(s_1)$ for all $1 \leq i \leq n$.

Iteration step. Assume that $\llbracket \exists \diamond_\alpha \varphi \rrbracket_{\mathcal{M}_i}^P(s_j)$ has been calculated. We know that it is equal to $\llbracket \exists \diamond_\alpha \varphi \rrbracket_{\mathcal{M}}^P(s_j)$ if $j \leq i$.

One can calculate of $\llbracket \exists \diamond_\alpha \varphi \rrbracket_{\mathcal{M}_{i+1}}^P$ by solving a reachability problem in a (slightly modified) CTMC \mathcal{M}'_{i+1} . First, select some path σ through \mathcal{M}'_{i+1} and at a suitable moment t_{iter} take the utility $\llbracket \exists \diamond_\alpha \varphi \rrbracket_{\mathcal{M}_i}^P(\sigma @ t_{\text{iter}})$. If σ starts in s_1, \dots, s_i , we can pick $t_{\text{iter}} = 0$. To simplify the choice, these states are absorbing in \mathcal{M}'_{i+1} , so that any value for t_{iter} is equivalent to picking $t_{\text{iter}} = 0$.

Otherwise, we follow the transitions of σ until it reaches some state in s_1, \dots, s_i ; then, t_{iter} is the time of reaching such a state for the first time. To take into account the discount factor, we add an (absorbing) state s_\perp with $\llbracket \varphi \rrbracket_{\mathcal{M}'_{i+1}}^P(s_\perp) := 0$, and from each state in s_{i+1}, \dots, s_n , we add a transition with rate α to s_\perp . However, we never wait longer than $t_{\text{cut}} := \lceil \ln \llbracket \varphi \rrbracket_{\mathcal{M}}^P(s_i) - \ln \llbracket \varphi \rrbracket_{\mathcal{M}}^P(s_{i+1}) \rceil / \alpha$: If no such state is reached before t_{cut} , we pick $t_{\text{iter}} = t_{\text{cut}}$.

Overall, \mathcal{M}'_{i+1} has state space $S_{\mathcal{M}} \cup \{s_\perp\}$, rate matrix

$$R_{\mathcal{M}'_{i+1}} = \left(\begin{array}{ccc|c} & & & 0 \\ & \mathbf{0} & & \vdots \\ & & & 0 \\ \hline & & & \alpha \\ & (R_{\mathcal{M}})_{i+1 \dots n} & & \vdots \\ & & & \alpha \\ \hline 0 & \dots & 0 & 0 \end{array} \right) \left. \begin{array}{l} \} i \text{ zero rows} \\ \\ \} n - i \text{ rows from } R_{\mathcal{M}} = R_{\mathcal{M}_{i+1}} \\ \\ \} \text{zero row for } s_\perp \end{array} \right\} \quad (3)$$

and the other parts of \mathcal{M}'_{i+1} coincide with \mathcal{M}_{i+1} . This rate matrix has the additional property that we can wait until t_{cut} in all cases: one only chooses an earlier moment for t_{iter} if one has reached s_1, \dots, s_i , and these states are absorbing. This leads to the following lemma:

Lemma 15. *With the sequence of \mathcal{M}_i introduced above, and $Q = R_{\mathcal{M}} - E_{\mathcal{M}} \cdot I$ the infinitesimal generator of \mathcal{M} ,*

$$\llbracket \exists \diamond_{\alpha} \varphi \rrbracket_{\mathcal{M}_{i+1}}^{\text{P}} = \exp((Q - \alpha I)|_{i+1 \dots n} t_{\text{cut}}) \llbracket \exists \diamond_{\alpha} \varphi \rrbracket_{\mathcal{M}_i}^{\text{P}}$$

Proof. We split the set of paths into two subsets, depending on whether $\{s_1, \dots, s_i\}$ is entered before t_{cut} or not. Then, we calculate the contribution to $\llbracket \exists \diamond_{\alpha} \varphi \rrbracket_{\mathcal{M}_{i+1}}^{\text{P}}(s_j)$ of both subsets separately; the total utility is the sum of the two contributions.

- Suppose σ enters some state $s_0 \in \{s_1, \dots, s_i\}$ at time $t < t_{\text{cut}}$ (after being in s_{i+1}, \dots, s_n); in that case, $\llbracket \diamond_{\alpha} \varphi \rrbracket_{\mathcal{M}_{i+1}}^{\text{P}}(\sigma) = e^{-\alpha t} \llbracket \exists \diamond_{\alpha} \varphi \rrbracket_{\mathcal{M}}^{\text{P}}(s_0)$, which is already known.

The probability density to enter such a state can be written as

$$\text{pdf}(s_{i+1} \vee \dots \vee s_n \mathcal{U}^=t s_0) = e^{Q|_{i+1 \dots n} t} Q|_{i+1 \dots n} \mathbf{1}|_{s_0}$$

where Q is the infinitesimal generator matrix of the Markov chain, $Q|_{i+1 \dots n}$ is the variant of Q where the states s_1, \dots, s_i are made absorbing, and $\mathbf{1}|_{s_0}$ is the vector containing a 1 in position s_0 and zeroes elsewhere. The contribution to $\llbracket \exists \diamond_{\alpha} \varphi \rrbracket_{\mathcal{M}_{i+1}}^{\text{P}}$ of these paths is therefore

$$\begin{aligned} \llbracket \exists \diamond_{\alpha} \varphi \rrbracket_{\mathcal{M}_{i+1}}^{\text{P}} \Big|_{\rightsquigarrow s_0} &= \int_0^{t_{\text{cut}}} e^{Q|_{i+1 \dots n} t} Q|_{i+1 \dots n} e^{-\alpha t} dt \llbracket \exists \diamond_{\alpha} \varphi \rrbracket_{\mathcal{M}_i}^{\text{P}} \Big|_{1 \dots i} \\ &\stackrel{*}{=} \int_0^{t_{\text{cut}}} e^{(Q-\alpha I)|_{i+1 \dots n} t} Q|_{i+1 \dots n} dt \llbracket \exists \diamond_{\alpha} \varphi \rrbracket_{\mathcal{M}_i}^{\text{P}} \Big|_{1 \dots i} \\ &\stackrel{\dagger}{=} \int_0^{t_{\text{cut}}} e^{(Q-\alpha I)|_{i+1 \dots n} t} (Q - \alpha I)|_{i+1 \dots n} dt \llbracket \exists \diamond_{\alpha} \varphi \rrbracket_{\mathcal{M}_i}^{\text{P}} \Big|_{1 \dots i} \\ &= (e^{(Q-\alpha I)|_{i+1 \dots n} t_{\text{cut}}} - I) \llbracket \exists \diamond_{\alpha} \varphi \rrbracket_{\mathcal{M}_i}^{\text{P}} \Big|_{1 \dots i} \end{aligned}$$

In the equality marked with $\stackrel{*}{=}$, we have used

$$e^{Q|_{i+1 \dots n} t} e^{-\alpha t} Q|_{i+1 \dots n} = e^{(Q|_{i+1 \dots n} - \alpha I)t} Q|_{i+1 \dots n} = e^{(Q-\alpha I)|_{i+1 \dots n} t} Q|_{i+1 \dots n}$$

and in the equality marked with $\stackrel{\dagger}{=}$:

$$Q|_{i+1 \dots n} \llbracket \exists \diamond_{\alpha} \varphi \rrbracket_{\mathcal{M}_i}^{\text{P}} \Big|_{1 \dots i} = (Q - \alpha I)|_{i+1 \dots n} \llbracket \exists \diamond_{\alpha} \varphi \rrbracket_{\mathcal{M}_i}^{\text{P}} \Big|_{1 \dots i}.$$

In both cases, the columns that do change in the left factor (the first i and the last $n - i$ columns, respectively) correspond to rows that are 0 in the right factor (the first i matrix rows and the last $n - i$ vector entries, respectively).

- Now suppose σ stays in states $\{s_{i+1}, \dots, s_n\}$ until t_{cut} ; in that case, we can as well continue (after t_{cut}) the evaluation of the path in \mathcal{M}_i , because the discount at that time (which is $e^{-\alpha t_{\text{cut}}} = \llbracket \varphi \rrbracket_{\mathcal{M}}^{\text{P}}(s_{i+1}) / \llbracket \varphi \rrbracket_{\mathcal{M}}^{\text{P}}(s_i)$) ensures that the difference between \mathcal{M}_{i+1} and \mathcal{M}_i will not lead to a higher value of $\llbracket \diamond_{\alpha} \varphi \rrbracket^{\text{P}}(\sigma)$. Therefore, we can conclude that $\llbracket \diamond_{\alpha} \varphi \rrbracket_{\mathcal{M}_{i+1}}^{\text{P}}(\sigma) = e^{-\alpha t_{\text{cut}}} \llbracket \diamond_{\alpha} \varphi \rrbracket_{\mathcal{M}_i}^{\text{P}}(\sigma @ t_{\text{cut}} \dots) = \llbracket \varphi \rrbracket_{\mathcal{M}}^{\text{P}}(s_{i+1}) / \llbracket \varphi \rrbracket_{\mathcal{M}}^{\text{P}}(s_i) \cdot \llbracket \diamond_{\alpha} \varphi \rrbracket_{\mathcal{M}_i}^{\text{P}}(\sigma @ t_{\text{cut}} \dots)$, which is easy to calculate.

In this case, the discount $e^{-\alpha t_{\text{cut}}}$ does not depend on t , so we can use the cdf directly instead of the pdf. The distribution over the states at t_{cut} can be described by the matrix

$$\Pr(s_{i+1} \vee \dots \vee s_n \mathcal{U}^=t_{\text{cut}} \cdot) = e^{Q|_{i+1 \dots n} t_{\text{cut}}}.$$

Therefore, the contribution to $\llbracket \exists \diamond_{\alpha} \varphi \rrbracket_{\mathcal{M}_{i+1}}^{\text{P}}$ of these paths is

$$\begin{aligned} \llbracket \exists \diamond_{\alpha} \varphi \rrbracket_{\mathcal{M}_{i+1}}^{\text{P}} \Big|_{\not\rightsquigarrow s_{1 \dots i}} &= e^{Q|_{i+1 \dots n} t_{\text{cut}}} e^{-\alpha t_{\text{cut}}} \llbracket \exists \diamond_{\alpha} \varphi \rrbracket_{\mathcal{M}_i}^{\text{P}} \Big|_{i+1 \dots n} \\ &= e^{(Q|_{i+1 \dots n} - \alpha I)t_{\text{cut}}} \llbracket \exists \diamond_{\alpha} \varphi \rrbracket_{\mathcal{M}_i}^{\text{P}} \Big|_{i+1 \dots n} \\ &\stackrel{*}{=} e^{(Q-\alpha I)|_{i+1 \dots n} t_{\text{cut}}} \llbracket \exists \diamond_{\alpha} \varphi \rrbracket_{\mathcal{M}_i}^{\text{P}} \Big|_{i+1 \dots n} \end{aligned}$$

The equality marked with $*$ holds for similar reasons as in the first case.

If we sum the two parts together, and also add the probability for the paths starting in $\{s_1, \dots, s_i\}$, we get:

$$\begin{aligned} \llbracket \exists \diamond_{\alpha} \varphi \rrbracket_{\mathcal{M}_{i+1}}^P &= (e^{(Q-\alpha I)|_{i+1\dots n} t_{\text{cut}}} - I) \llbracket \exists \diamond_{\alpha} \varphi \rrbracket_{\mathcal{M}_i}^P \Big|_{1\dots i} \\ &\quad + e^{(Q-\alpha I)|_{i+1\dots n} t_{\text{cut}}} \llbracket \exists \diamond_{\alpha} \varphi \rrbracket_{\mathcal{M}_i}^P \Big|_{i+1\dots n} \\ &\quad + \llbracket \exists \diamond_{\alpha} \varphi \rrbracket_{\mathcal{M}_i}^P \Big|_{1\dots i} \\ &= e^{(Q-\alpha I)|_{i+1\dots n} t_{\text{cut}}} \llbracket \exists \diamond_{\alpha} \varphi \rrbracket_{\mathcal{M}_i}^P \end{aligned}$$

□

Time Complexity. Checking one $\exists \diamond_{\alpha}$ operator requires to solve a sequence of related CTMC reachability problems. Using Jensen's method (similar to checking CSL properties of CTMCs, see [3, 9]), we know that it takes $O((E_{\mathcal{M}} + \alpha)t|R| \cdot |S|)$ arithmetic operations, where t is the maximal time bound. This is comparable to [1, Lemma 19]: The time bound for checking the corresponding formula in a DTMC is in $O(|S|^3 \cdot K)$, where K is the maximal step bound. The time bound t is, in principle, the total of all cutoff times; however, if it is very large, the discount $e^{-\alpha t}$ becomes so small that one can abort the calculation before all cutoff times have passed. For an error bound ε , we have that $e^{-\alpha t} < \varepsilon$ iff $t > \frac{1}{\alpha} \ln \frac{1}{\varepsilon}$.

Proposition 16. *The problem of model checking $\llbracket \varphi \rrbracket_{\mathcal{M}}^P$ (not containing Δ_{α} operators) can be solved in time $O(|\varphi| \cdot |R| \cdot |S|^{\frac{E+\alpha}{\alpha}} \ln \frac{1}{\varepsilon})$.*

5 Other Scheduler Classes

The article [1] uses the best and worst *history-dependent* scheduler. In the above definitions, we looked for the best and worst positional scheduler. In this section, we illustrate how the choice of scheduler class influences the semantics. To make the scheduler class more clear, we will add a subscript \mathfrak{D} to \exists , the main operator that chooses from the set of schedulers \mathfrak{D} . In the following, we will give some examples for the $\exists_{\mathfrak{D}}$ operator for several classes of schedulers. The treatment of the $\forall_{\mathfrak{D}}$ operator always follows the same lines and does not enlighten more.

We designate the (time-abstract) positional schedulers used above by \mathfrak{P} , and we will introduce the classes $\mathfrak{T}\mathfrak{P}$ of total-time positional schedulers and $\mathfrak{L}\mathfrak{P}$ of late time-abstract positional schedulers.

5.1 Late Schedulers ($\mathfrak{L}\mathfrak{P}$)

A CTMDP is *locally uniform* if in every state, the exit rate does not depend on the action chosen (i. e., $\forall s \in S : \forall a, b \in A : E^a(s) = E^b(s)$). In a locally uniform CTMDP, it is possible to delay the choice for one action or another until the moment that the sojourn time in a state has passed. So-called *late* schedulers, defined in [6], are functions from the current state and the sojourn time in this state to an action that is enabled: $D : S \times \mathbb{R}_{\geq 0} \rightarrow A$. As in time-bounded reachability [6], a late scheduler can improve on the value of a formula.

Example 17. Consider the CTMDP in Fig. 3, with initial state s_4 . The best positional scheduler for $\llbracket \exists \diamond_1 \text{black} \rrbracket^P(s_4)$ always chooses action b , as the expected utility is higher than when choosing a . This can be seen by reusing (3a) and (3b) in Example 18 and letting $t_3 = 0$:

$$\begin{aligned} \llbracket \exists_{\{D_a\}} \diamond_1 \varphi \rrbracket^P(s_4) &= \frac{1}{384} e^{2 \cdot 0} + \frac{1}{3} e^{-0} + \frac{1}{8} = \frac{59}{128} = 0.4609375 \\ \llbracket \exists_{\{D_b\}} \diamond_1 \varphi \rrbracket^P(s_4) &= \frac{1}{108} e^{2 \cdot 0} + \frac{1}{2} e^{-0} = \frac{55}{108} \approx 0.50926 \end{aligned}$$

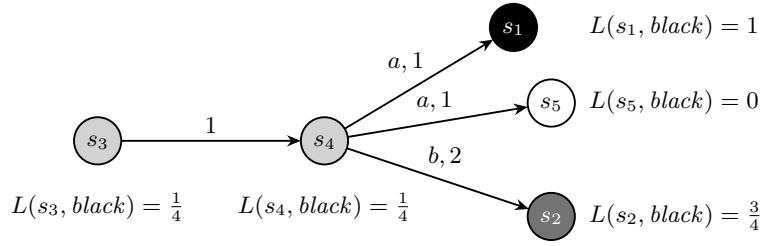


Fig. 3. Timed schedulers are better than positional ones.

However, a late scheduler can base its choice also on t_4 , the sojourn time in state s_4 . For times $t_4 \in [\ln 3, \ln 4)$, choosing b will not improve $\llbracket \exists \diamond_1 \text{black} \rrbracket^P(s_4)$ because $e^{-t_4} \llbracket \text{black} \rrbracket^P(s_2) \leq \llbracket \text{black} \rrbracket^P(s_4)$, but by choosing a , it improves with probability of $\frac{1}{2}$, because $e^{-t_4} \llbracket \text{black} \rrbracket^P(s_1) > \llbracket \text{black} \rrbracket^P(s_4)$. For very short sojourn times $t_4 \in [0, S)$, for some $S \leq \ln 3$, it is more advantageous to choose b . To find S , one solves the integral:

$$\begin{aligned}
\llbracket \exists \diamond_1 \text{black} \rrbracket^P(s_4) &= \int_0^{\infty} \text{pdf}(s_4 \xrightarrow{t_4}) \left(\underbrace{\left(\frac{1}{2} \left(\frac{1}{4} \sqcup e^{-t_4} \right) + \frac{1}{2} \cdot \frac{1}{4} \right)}_{\text{action } a \text{ is chosen}} \sqcup \underbrace{\frac{3}{4} e^{-t_4}}_{\text{action } b} \right) dt_4 \\
&= \int_0^{\infty} \text{pdf}(s_4 \xrightarrow{t_4}) \left(\frac{1}{4} \sqcup \left(\frac{1}{8} + \frac{1}{2} e^{-t_4} \right) \sqcup \frac{3}{4} e^{-t_4} \right) dt_4 \\
&= \int_0^{\ln 2} 2e^{-2t_4} \frac{3}{4} e^{-t_4} dt_4 + \int_{\ln 2}^{\ln 4} 2e^{-2t_4} \left(\frac{1}{8} + \frac{1}{2} e^{-t_4} \right) dt_4 + \int_{\ln 4}^{\infty} 2e^{-2t_4} \frac{1}{4} dt_4 \\
&= \frac{1}{2} (1 - e^{-3 \ln 2}) + \frac{1}{8} (e^{-2 \ln 2} - e^{-2 \ln 4}) + \frac{1}{3} (e^{-3 \ln 2} - e^{-3 \ln 4}) + \frac{1}{4} e^{-2 \ln 4} \\
&= \frac{197}{384} \approx 0.51302 > 0.50926
\end{aligned}$$

The exact switching point is therefore $S = \ln 2$.

5.2 Total-Time Positional Schedulers (\mathfrak{TTP})

A total-time positional scheduler bases its choice not only on the current state, but also on the total time that the path has taken until now. Formally, such a scheduler is a function $D : S \times \mathbb{R}_{\geq 0} \rightarrow A$ but the real parameter now indicates the total time spent before entering the current state.

Example 18. Consider the CTMDP in Fig. 3 again. We now regard s_3 as the initial state. Similar to Example 17, the best positional scheduler would always choose b in s_4 , but a \mathfrak{TTP} -scheduler can improve on this value because it bases its decision also on t_3 , the sojourn time in state s_3 . A similar integral as above shows that an optimal scheduler chooses b for times $t_3 \in [0, R)$, where $R \approx 0.41903$, and chooses a for $t_3 \in (R, \ln 4)$. Note that this scheduler has to estimate how long the sojourn time in s_4 will be when it makes its decision; therefore, it is less exact than the late scheduler of Example 17.

To see this, let D_a be the positional scheduler that always chooses a in s_4 , and let D_b be the scheduler that always chooses b . We then have:

$$\llbracket \exists_{\{D_a\}} \diamond_1 \varphi \rrbracket^P(s_3) =$$

$$\begin{aligned}
&= \int_0^{\ln 4} e^{-t} \left[\overbrace{\int_0^{\ln 4-t} \frac{1}{2} 2e^{-2u} e^{-(t+u)} du}^{C(s_3, [0, \ln 4], s_4) \text{ and reach } s_1 \text{ early}} + \overbrace{\int_{\ln 4-t}^{\infty} \frac{1}{2} 2e^{-2u} \frac{1}{4} du}^{C(s_3, [0, \ln 4], s_4) \text{ and reach } s_1 \text{ late}} + \overbrace{\int_0^{\infty} \frac{1}{2} 2e^{-2u} \frac{1}{4} du}^{C(s_3, [0, \ln 4], s_4, \mathbb{R}_{\geq 0}, s_5)} \right] dt + \overbrace{\int_{\ln 4}^{\infty} e^{-t} \frac{1}{4} dt}^{C(s_3, [\ln 4, \infty), s_4)} \\
&= \int_0^{\ln 4} e^{-t} \left[-\frac{1}{3} e^{-3u} \Big|_0^{\ln 4-t} \cdot e^{-t} - \frac{1}{2} e^{-2u} \Big|_{\ln 4-t}^{\infty} \cdot \frac{1}{4} - \frac{1}{2} e^{-2u} \Big|_0^{\infty} \cdot \frac{1}{4} \right] dt - e^{-t} \Big|_{\ln 4}^{\infty} \cdot \frac{1}{4} \\
&= \int_0^{\ln 4} e^{-t} \left[\frac{1}{3} (1 - e^{-3(\ln 4-t)}) e^{-t} + \frac{1}{2} e^{-2(\ln 4-t)} \frac{1}{4} + \frac{1}{2} \cdot \frac{1}{4} \right] dt + e^{-\ln 4} \frac{1}{4} \\
&= \int_0^{\ln 4} e^{-t} \left[\frac{1}{3} e^{-t} - \frac{1}{3} 4^{-3} e^{2t} + \frac{1}{8} 4^{-2} e^{2t} + \frac{1}{8} \right] dt + 4^{-1} \frac{1}{4} \\
&= \int_0^{\ln 4} \left[\frac{1}{3} e^{-2t} + \left(-\frac{1}{192} + \frac{1}{128} \right) e^t + \frac{1}{8} e^{-t} \right] dt + \frac{1}{16} \\
&= \left[-\frac{1}{3} \cdot \frac{1}{2} e^{-2t} + \frac{1}{384} e^t - \frac{1}{8} e^{-t} \right]_0^{\ln 4} + \frac{1}{16} \\
&= \frac{1}{6} (1 - e^{-2 \ln 4}) + \frac{1}{384} (e^{\ln 4} - 1) + \frac{1}{8} (1 - e^{-\ln 4}) + \frac{1}{16} \\
&= \frac{123}{384} = \frac{41}{128} = 0.3203125
\end{aligned}$$

and similarly

$$\begin{aligned}
&[[\exists_{\{D_b\}} \diamond_1 \varphi]]^P(s_3) = \\
&= \int_0^{\ln 3} e^{-t} \left[\overbrace{\int_0^{\ln 3-t} 2e^{-2u} \frac{3}{4} e^{-(t+u)} du}^{C(s_3, [0, \ln 3], s_4) \text{ and reach } s_2 \text{ early}} + \overbrace{\int_{\ln 3-t}^{\infty} 2e^{-2u} \frac{1}{4} du}^{C(s_3, [0, \ln 3], s_4) \text{ and reach } s_2 \text{ late}} \right] dt + \overbrace{\int_{\ln 3}^{\infty} e^{-t} \frac{1}{4} dt}^{C(s_3, [\ln 3, \infty), s_4)} \\
&= \int_0^{\ln 3} e^{-t} \left[\frac{2}{4} e^{-t} - \frac{2}{4} 3^{-3} e^{2t} + 3^{-2} \frac{1}{4} e^{2t} \right] dt + 3^{-1} \frac{1}{4} \\
&= \int_0^{\ln 3} \left[\frac{1}{2} e^{-2t} + \left(-\frac{1}{54} + \frac{1}{36} \right) e^t \right] dt + \frac{1}{12} \\
&= \frac{35}{108} \approx 0.32407
\end{aligned}$$

So, $[[\exists_{\mathfrak{P}} \diamond_1 \varphi]]^P(s_3) = \frac{35}{108}$. For the \mathfrak{ITP} -scheduler, we calculate:

$$\begin{aligned}
[[\exists_{\mathfrak{ITP}} \diamond_1 \varphi]]^P(s_3) &= \int_0^{\infty} \text{pdf}(s_3 \xrightarrow{t_3} s_4) \\
&\quad \left[\int_0^{\infty} \text{pdf}(s_4 \xrightarrow{t_2} s_1) \left(\frac{1}{4} \sqcup 1 e^{-(t_3+t_2)} \right) + \text{pdf}(s_4 \xrightarrow{t_2} s_5) \frac{1}{4} dt_2 \right. \\
&\quad \left. \sqcup \int_0^{\infty} \text{pdf}(s_4 \xrightarrow{t_2} s_2) \left(\frac{1}{4} \sqcup \frac{3}{4} e^{-(t_3+t_2)} \right) dt_2 \right] dt_3 \\
&= \int_0^{\ln 3} e^{-t_3} \left[\left(\frac{1}{384} e^{2t_3} + \frac{1}{3} e^{-t_3} + \frac{1}{8} \right) \sqcup \left(\frac{1}{108} e^{2t_3} + \frac{1}{2} e^{-t_3} \right) \right] dt_3
\end{aligned}$$

$$\begin{aligned}
& + \int_{\ln 3}^{\ln 4} e^{-t_3} \left[\left(\frac{1}{384} e^{2t_3} + \frac{1}{3} e^{-t_3} + \frac{1}{8} \right) \sqcup \frac{1}{4} \right] dt_3 \\
& + \int_{\ln 4}^{\infty} e^{-t_3} \left[\frac{1}{4} \sqcup \frac{1}{4} \right] dt_3
\end{aligned}$$

For the integral $\int_0^{\ln 3}$, the inequation $\frac{1}{384} e^{2t_3} + \frac{1}{3} e^{-t_3} + \frac{1}{8} \geq \frac{1}{108} e^{2t_3} + \frac{1}{2} e^{-t_3}$ is equivalent to $0 \geq (e^{-t_3})^3 - \frac{3}{4} (e^{-t_3})^2 + \frac{23}{576}$. The relevant real root is at $e^{-t_3} = R := \frac{1}{24} \sqrt[3]{-60 + 12\sqrt{299}i} + \frac{1}{24} \sqrt[3]{-60 - 12\sqrt{299}i} + \frac{1}{4}$, i.e. the inequation is equivalent to $t_3 \leq -\ln R \approx 0.41903$. – For $\int_{\ln 3}^{\ln 4}$, the inequation $\frac{1}{384} e^{2t_3} + \frac{1}{3} e^{-t_3} + \frac{1}{8} \geq \frac{1}{4}$ is equivalent to $(e^{-t_3} + \frac{1}{8})(e^{-t_3} - \frac{1}{4})^2 \geq 0$. This holds for $e^{-t_3} \geq -\frac{1}{8}$, and therefore for all $t_3 \in \mathbb{R}$. We also aggregate $\int_{-\ln R}^{\ln 3}$ with $\int_{\ln 3}^{\ln 4}$:

$$\begin{aligned}
& \llbracket \exists_{\mathbb{R}^3} \diamond_1 \varphi \rrbracket^{\mathbb{P}}(s_3) = \\
& = \int_0^{-\ln R} \frac{1}{108} e^{t_3} + \frac{1}{2} e^{-2t_3} dt_3 + \int_{-\ln R}^{\ln 4} \frac{1}{384} e^{t_3} + \frac{1}{3} e^{-2t_3} + \frac{1}{8} e^{-t_3} dt_3 + \int_{\ln 4}^{\infty} e^{-t_3} \frac{1}{4} dt_3 \\
& = \frac{e^{-\ln R} - 1}{108} + \frac{1 - e^{2\ln R}}{2 \cdot 2} + \frac{e^{\ln 4} - e^{-\ln R}}{384} + \frac{e^{2\ln R} - e^{-2\ln 4}}{3 \cdot 2} + \frac{e^{\ln R} - e^{-\ln 4}}{8} + e^{-\ln 4} \frac{1}{4} \\
& = \frac{1}{108} R^{-1} - \frac{1}{108} + \frac{1}{4} - \frac{1}{4} R^2 + \frac{1}{384} 4 - \frac{1}{384} R^{-1} + \frac{1}{6} R^2 - \frac{1}{6} 4^{-2} + \frac{1}{8} R - \frac{1}{8} 4^{-1} + \frac{1}{16} \\
& = -\frac{1}{12} R^2 + \frac{1}{8} R + \frac{235}{864} + \frac{23}{3456} R^{-1} \approx 0.32827 > 0.32407
\end{aligned}$$

In the above calculations, we have used the following auxiliary equalities. If $t_3 \leq \ln 4$, we have:

$$\begin{aligned}
& \int_0^{\infty} \text{pdf}\left(s_4 \xrightarrow{\frac{t_2}{a}} s_1\right) \left(\frac{1}{4} \sqcup 1e^{-(t_3+t_2)}\right) + \text{pdf}\left(s_4 \xrightarrow{\frac{t_2}{a}} s_5\right) \frac{1}{4} dt_2 \\
& = \int_0^{\ln 4 - t_3} e^{-2t_2} e^{-(t_3+t_2)} + e^{-2t_2} \frac{1}{4} dt_2 + \int_{\ln 4 - t_3}^{\infty} e^{-2t_2} \frac{1}{4} + e^{-2t_2} \frac{1}{4} dt_2 \\
& = \frac{1}{3} (1 - e^{-3(\ln 4 - t_3)}) e^{-t_3} + \frac{1}{2} (1 - e^{-2(\ln 4 - t_3)}) \frac{1}{4} + e^{-2(\ln 4 - t_3)} \frac{1}{4} = \frac{1}{384} e^{2t_3} + \frac{1}{3} e^{-t_3} + \frac{1}{8} \quad (3a)
\end{aligned}$$

If $t_3 \geq \ln 4$, we have:

$$\int_0^{\infty} \text{pdf}\left(s_4 \xrightarrow{\frac{t_2}{a}} s_1\right) \left(\frac{1}{4} \sqcup 1e^{-(t_3+t_2)}\right) + \text{pdf}\left(s_4 \xrightarrow{\frac{t_2}{a}} s_5\right) \frac{1}{4} dt_2 = \int_0^{\infty} e^{-2t_2} \frac{1}{4} + e^{-2t_2} \frac{1}{4} dt_2 = \frac{1}{4}$$

If $t_3 \leq \ln 3$, we have:

$$\begin{aligned}
& \int_0^{\infty} \text{pdf}\left(s_4 \xrightarrow{\frac{t_2}{b}} s_2\right) \left(\frac{1}{4} \sqcup \frac{3}{4} e^{-(t_3+t_2)}\right) dt_2 = \int_0^{\ln 3 - t_3} 2e^{-2t_2} \frac{3}{4} e^{-(t_3+t_2)} dt_2 + \int_{\ln 3 - t_3}^{\infty} 2e^{-2t_2} \frac{1}{4} dt_2 \\
& = \frac{2}{3} (1 - e^{-3(\ln 3 - t_3)}) \frac{3}{4} e^{-t_3} + e^{-2(\ln 3 - t_3)} \frac{1}{4} = \frac{1}{108} e^{2t_3} + \frac{1}{2} e^{-t_3} \quad (3b)
\end{aligned}$$

If $t_3 \geq \ln 3$, we have:

$$\int_0^{\infty} \text{pdf}\left(s_4 \xrightarrow{\frac{t_2}{b}} s_2\right) \left(\frac{1}{4} \sqcup \frac{3}{4} e^{-(t_3+t_2)}\right) dt_2 = \int_0^{\infty} 2e^{-2t_2} \frac{1}{4} dt_2 = \frac{1}{4}$$

5.3 History-Dependent Schedulers (\mathfrak{H})

A history-dependent scheduler, in principle, has access to the full history, i. e. to the complete path to make a decision. In practice, it is enough to use the highest utility u_{\max} achieved until now to decide which actions still could improve on the utility of $\llbracket \exists_{\mathfrak{H}} \diamond_{\alpha} \varphi \rrbracket^{\mathcal{P}}$. Additionally, similar in Example 18, the decision should take into account the discount incurred between the time t_{\max} when that utility was achieved and the current time t . Therefore, a history-dependent scheduler has enough information if it knows the value $u_{\max} e^{\alpha(t-t_{\max})}$.

This is basically the idea used by [1] to find a model checking algorithm for the path semantics of MDPs with history-dependent schedulers. Why can't we use this idea for CTMDPs? The problem is that in discrete-time systems, discount factors are powers of a constant $\rho \in (0, 1)$. As a consequence, only a finite number of the powers $u_{\max}, \rho^{-1}u_{\max}, \rho^{-2}u_{\max}, \dots$ is relevant to find $\llbracket \exists_{\mathfrak{H}} \diamond_{\rho} \varphi \rrbracket^{\mathcal{P}}$. This allows to define a linear program with finitely many variables to calculate the utilities. However, in the continuous-time case, the corresponding linear program would have infinitely many variables, so one cannot prove termination of the algorithm that solves the linear program.

To argue in more detail, I follow the construction given in [1]. To simplify notation, I assume that the CTMDP is uniform. (If it isn't, one can uniformise it following the uniformisation procedure proposed in [3], by adding states that do not allow to change the decision. The form of our schedulers does not allow to exploit the additional information provided by visiting these additional states, because our schedulers do not count the number of transitions taken.) Let

$$h^{\exists}(s, x) = \sup_{D \in \mathfrak{H}} \mathbb{E}_s^D \left[x \sqcup \max_{t \geq 0} e^{-\alpha t} \llbracket \varphi \rrbracket^{\mathcal{P}} (\sigma @ t) \right].$$

Obviously, $\llbracket \exists_{\mathfrak{H}} \diamond_{\alpha} \varphi \rrbracket^{\mathcal{P}} (s) = h^{\exists}(s, 0)$. We now define an operator, which is a contraction with fixpoint h^{\exists} , by:

$$H^{\exists}(h)(s, x) = \begin{cases} x & \text{if } x \geq 1 \\ \max_{a \in A} \sum_{s' \in S} \int_0^{\infty} \text{pdf} \left(s \xrightarrow{t} s' \mid a \right) e^{-\alpha t} h(s', e^{\alpha t} [x \sqcup \llbracket \varphi \rrbracket^{\mathcal{P}} (s)]) dt & \text{otherwise} \end{cases}$$

$$= \begin{cases} x & \text{if } x \geq 1 \\ E \max_{a \in A} \sum_{s' \in S} P^a(s, s') \int_0^{\infty} e^{-(E+\alpha)t} h(s', e^{\alpha t} x) dt & \text{if } \llbracket \varphi \rrbracket^{\mathcal{P}} (s) < x < 1 \\ E \max_{a \in A} \sum_{s' \in S} P^a(s, s') \int_0^{\infty} e^{-(E+\alpha)t} h(s', e^{\alpha t} \llbracket \varphi \rrbracket^{\mathcal{P}} (s)) dt & \text{otherwise} \end{cases}$$

Substitute $u = e^{\alpha t} x$ (so $t = \ln(u/x)/\alpha$ and $dt = \frac{1}{\alpha u} du$) in the second case, and $u = e^{\alpha t} \llbracket \varphi \rrbracket^{\mathcal{P}} (s)$ in the third case:

$$H^{\exists}(h)(s, x) = \begin{cases} x & \text{if } x \geq 1 \\ E \max_{a \in A} \sum_{s' \in S} P^a(s, s') \int_x^{\infty} \left(\frac{u}{x} \right)^{-(E+\alpha)/\alpha} h(s', u) \frac{1}{\alpha u} du & \text{if } \llbracket \varphi \rrbracket^{\mathcal{P}} (s) < x < 1 \\ E \max_{a \in A} \sum_{s' \in S} P^a(s, s') \int_{\llbracket \varphi \rrbracket^{\mathcal{P}} (s)}^{\infty} \left(\frac{u}{\llbracket \varphi \rrbracket^{\mathcal{P}} (s)} \right)^{-(E+\alpha)/\alpha} h(s', u) \frac{1}{\alpha u} du & \text{otherwise} \end{cases}$$

$$= \begin{cases} x & \text{if } x \geq 1 \\ \frac{E}{\alpha} x^{1+E/\alpha} \max_{a \in A} \sum_{s' \in S} P^a(s, s') \int_x^{\infty} u^{-2-E/\alpha} h(s', u) du & \text{if } \llbracket \varphi \rrbracket^{\mathcal{P}} (s) < x < 1 \\ \frac{E}{\alpha} \llbracket \varphi \rrbracket^{\mathcal{P}} (s)^{1+E/\alpha} \max_{a \in A} \sum_{s' \in S} P^a(s, s') \int_{\llbracket \varphi \rrbracket^{\mathcal{P}} (s)}^{\infty} u^{-2-E/\alpha} h(s', u) du & \text{otherwise} \end{cases}$$

Now assume that $h(s, x) \equiv x$ if $x \geq 1$. We then can evaluate the integral for $u \geq 1$. This leads to $\frac{E}{\alpha} x^{1+E/\alpha} \max_{a \in A} \sum_{s' \in S} P^a(s, s') \int_1^\infty u^{-2-E/\alpha} h(s', u) du = \frac{E}{\alpha} x^{1+E/\alpha} \int_1^\infty u^{-1-E/\alpha} du = \frac{E}{\alpha} x^{1+E/\alpha} \cdot \frac{1}{-E/\alpha} u^{-E/\alpha} \Big|_1^\infty = 0 - -x^{1+E/\alpha}$, and similarly for $u \llbracket \varphi \rrbracket^P(s) \geq 1$.

$$H^\exists(h)(s, x) = \begin{cases} x & \text{if } x \geq 1 \\ x^{1+E/\alpha} + \frac{E}{\alpha} x^{1+E/\alpha} \max_{a \in A} \sum_{s' \in S} P^a(s, s') \int_x^1 u^{-2-E/\alpha} h(s', u) du & \text{if } \llbracket \varphi \rrbracket^P(s) < x < 1 \\ \llbracket \varphi \rrbracket^P(s)^{1+E/\alpha} & \text{otherwise} \\ \frac{E}{\alpha} \llbracket \varphi \rrbracket^P(s)^{1+E/\alpha} \max_{a \in A} \sum_{s' \in S} P^a(s, s') \int_{\llbracket \varphi \rrbracket^P(s)}^1 u^{-2-E/\alpha} h(s', u) du & \text{otherwise} \end{cases}$$

Now one can always choose the exit rate E such that

$$F := E/\alpha \in \mathbb{N}.$$

Then $u^{-2-E/\alpha}$ has an integer exponent, and it almost looks as if $h(s, x)$, for $\llbracket \varphi \rrbracket^P(s) < x < 1$, is a polynomial in x : Let

$$\begin{aligned} h^0(s, x) &:= x \sqcup \llbracket \varphi \rrbracket^P(s) \\ h^{i+1}(s, x) &:= H^\exists(h^i). \end{aligned} \tag{3c}$$

However, $h^1(s, x)$ (for $\llbracket \varphi \rrbracket^P(s) < x < 1$) may include a term x^{1+F} , and therefore when calculating h^2 , we have to integrate over $u^{-2-F}[u^{1+F} + \dots] = u^{-1} + \dots$. Therefore, h^2 may include a term $\ln x$. Further integrals show that there may also be powers of $\ln x$ in h^3 and higher approximations.

Proposition 18a. *The $h^i(s, x)$ (for $i \geq 1$) defined in (3c) have the form*

$$h^i(s, x) = \begin{cases} x & \text{if } x \geq 1 \\ \max_{a \in A} \sum_{j=0}^{1+F} \sum_{k=0}^{i-1} c_{s'jk}^a \cdot x^j (\ln x)^k & \text{if } \llbracket \varphi \rrbracket^P(s) < x < 1 \\ C_s & \text{otherwise} \end{cases} \tag{3d}$$

where $c_{s'jk}^a$ and C_s are real numbers.

Proof. By induction on i .

Base case. h^1 is defined to be $H^\exists(h^0)$. From the definition of H^\exists , it is easy to see that $h^1(s, x) = x$ if $x \geq 1$ and that for each $s \in S$, the restriction of $h^1(s, x)$ to $x \leq \llbracket \varphi \rrbracket^P(s)$ is constant, as required. Now assume that $\llbracket \varphi \rrbracket^P(s) < x < 1$. Then,

$$h^1(s, x) = x^{1+F} + Fx^{1+F} \max_{a \in A} \sum_{s' \in S} P^a(s, s') \int_x^1 u^{-2-F} \underbrace{(u \sqcup \llbracket \varphi \rrbracket^P(s'))}_{h^0(s', u)} du$$

We now concentrate on calculating the integral $I(x) := \int_x^1 u^{-2-F} (u \sqcup \llbracket \varphi \rrbracket^P(s')) du$. Note that if the integral is a Laurent polynomial whose nonzero coefficients belong to $x^0, x^{-1}, x^{-2}, \dots, x^{-1-F}$, then h^1 has the required form. Depending on whether $x < \llbracket \varphi \rrbracket^P(s')$ or not, one has to split the integral.

1. $x \geq \llbracket \varphi \rrbracket^P(s')$: There is no need to split the integral. Then,

$$I(x) = \int_x^1 u^{-2-F} u du = \left[\frac{u^{-F}}{-F} \right]_x^1 = \frac{x^{-F} - 1}{F}$$

2. $x < \llbracket \varphi \rrbracket^{\text{P}}(s')$: We split the integral into $\int_x^{\llbracket \varphi \rrbracket^{\text{P}}(s')} \dots du + \int_{\llbracket \varphi \rrbracket^{\text{P}}(s')}^1 \dots du$. Then,

$$\begin{aligned}
I(x) &= \int_x^{\llbracket \varphi \rrbracket^{\text{P}}(s')} u^{-2-F} \llbracket \varphi \rrbracket^{\text{P}}(s') du + \int_{\llbracket \varphi \rrbracket^{\text{P}}(s')}^1 u^{-2-F} u du \\
&= \llbracket \varphi \rrbracket^{\text{P}}(s') \left[\frac{u^{-1-F}}{-1-F} \right]_x^{\llbracket \varphi \rrbracket^{\text{P}}(s')} + \left[\frac{u^{-F}}{-F} \right]_{\llbracket \varphi \rrbracket^{\text{P}}(s')}^1 \\
&= \llbracket \varphi \rrbracket^{\text{P}}(s') \frac{x^{-1-F} - \llbracket \varphi \rrbracket^{\text{P}}(s')^{-1-F}}{1+F} + \frac{\llbracket \varphi \rrbracket^{\text{P}}(s')^{-F} - 1}{F} \\
&= \frac{\llbracket \varphi \rrbracket^{\text{P}}(s')}{1+F} x^{-1-F} + \frac{\llbracket \varphi \rrbracket^{\text{P}}(s')^{-F}}{F(1+F)} - \frac{1}{F}
\end{aligned}$$

In both cases, we get Laurent polynomial of the required form, so h^1 fits into the scheme prescribed by (3d).

Iteration step. Assume that h^i has the form described above. We are going to calculate $h^{i+1} = H^{\exists}(h^i)$. Obviously, if $x \geq 1$ or $x \leq \llbracket \varphi \rrbracket^{\text{P}}(s)$, then h^{i+1} has the required form. In the remaining case, we have to solve the integral $I(x) := \int_x^1 u^{-2-F} h^i(s', u) du$. Again, if this is a kind of Laurent polynomial (now perhaps including coefficients for $\ln x$), then h^{i+1} has the required form.

If $x \leq \llbracket \varphi \rrbracket^{\text{P}}(s')$, we get a calculation that is very similar to the second part of the base case above: one gets that $I(x) = Ax^{-1-F} + B$, for some constants $A, B \in \mathbb{R}$. Therefore, let us assume that $\llbracket \varphi \rrbracket^{\text{P}}(s') < x < 1$.

$$I(x) = \int_x^1 u^{-2-F} \max_{a \in A} \sum_{j=0}^{1+F} \sum_{k=0}^{i-1} c_{s'jk}^a u^j (\ln u)^k du$$

We want to split this integral into pieces, such that only one action $a \in A$ has to be considered for every piece. For every piece, one calculates the related integral

$$J^a(x, y) := \int_x^y u^{-2-F} \sum_{j=0}^{1+F} \sum_{k=0}^{i-1} c_{s'jk}^a u^j (\ln u)^k du$$

Then, $I(x)$ is a sum of the form $J^{a_1}(x, x_1) + J^{a_2}(x_1, x_2) + \dots + J^{a_n}(x_{n-1}, 1)$. Note that only $J^{a_1}(x, x_1)$ depends on x directly; the other terms should be real constants. If J^a has a suitable Laurent polynomial form (in the first variable), then h^{i+1} satisfies (3d). To calculate J^a , we use the known integrals (for $k \geq 0$):

$$\begin{aligned}
\int u^n (\ln u)^k du &= \frac{u^{n+1} (\ln u)^k}{n+1} - \frac{k}{n+1} \int u^n (\ln u)^{k-1} du \quad \text{if } n \neq -1 \\
&= \sum_{l=0}^k \frac{(-1)^l k!}{(k-l)!} \cdot \frac{u^{n+1} (\ln u)^{k-l}}{(n+1)^{l+1}} \\
\int u^{-1} (\ln u)^k du &= \frac{(\ln u)^{k+1}}{k+1}
\end{aligned}$$

and continue, by distinguishing the case $j = 1 + F$:

$$\begin{aligned}
J^a(x, y) &= \left(\sum_{j=0}^F \sum_{k=0}^{i-1} c_{s'jk}^a \int_x^y u^{j-2-F} (\ln u)^k du \right) + \sum_{k=0}^{i-1} c_{s'(1+F)k}^a \int_x^y u^{-1} (\ln u)^k du \\
&= \sum_{j=0}^F \sum_{k=0}^{i-1} c_{s'jk}^a \left[\sum_{l=0}^k \frac{(-1)^l k!}{(k-l)!} \frac{u^{j-1-F} (\ln u)^{k-l}}{(j-1-F)^{l+1}} \right]_x^y + \sum_{k=0}^{i-1} c_{s'(1+F)k}^a \left[\frac{(\ln u)^{k+1}}{k+1} \right]_x^y \\
&= \sum_{j=0}^F \sum_{k=0}^{i-1} c_{s'jk}^a \sum_{l=0}^k k! \frac{y^{j-1-F} (\ln y)^{k-l} - x^{j-1-F} (\ln x)^{k-l}}{(-1)^l (k-l)! (j-1-F)^{l+1}} + \sum_{k=0}^{i-1} c_{s'(1+F)k}^a \frac{(\ln y)^{k+1} - (\ln x)^{k+1}}{k+1}
\end{aligned}$$

As $J^a(x, y)$ is multiplied with x^{1+F} , it follows that the result is a polynomial of the required form. \square

Proposition 18a shows that – at least in principle –, there is a finite description of the approximations h^i of h^\exists . Therefore, it is possible to devise an algorithm that approximates h^\exists ; however, the description and the calculations needed (as can be seen from the proof of Prop. 18a) are very complex. Therefore, I consider calculating h^\exists for history-dependent schedulers too difficult to be used.

6 Conclusion

It is possible to extend the DCTL semantics with time, to continuous-time Markov chains and MDPs. No big surprises have happened; the extension is pretty much straightforward. The distinction between the fixpoint and the path semantics known from the discrete-time case also applies to the continuous-time case. Properties interpreted under the fixpoint semantics can be easily checked using the same algorithms as in the discrete-time case; because fixpoint semantics do not look further ahead than one step, a variation on the expected time to take a transition (the reciprocal of the exit rate) is the discount per step. Properties interpreted under the path semantics are more difficult to verify; this article provides an algorithm for CTMCs (without nondeterminism).

To simplify the analysis, I have chosen to fix a single discount rate per property. This corresponds to an exponential decay of utility over time. One can without much work extend the current presentation and give each state its own discount rate (e.g. by adapting the matrix (3)).

Another way to make analysis of such temporal properties possible is taking a finite horizon. Everything before the time limit is equally relevant, and everything after is equally irrelevant. For example, time-bounded until formulas in CSL [3, 9] use a finite horizon. Depending on the situation, one or the other way of giving more weight to the near future is better-suited. If one does not want to predict for how long exactly a computer system will be used, it seems advantageous to use an analysis method that does not completely ignore a possible breakdown shortly after some deadline. Let me illustrate this point with an example: When a space agency writes its year report, it has to decide which missions have finished successfully; to make this decision possible, selecting a fixed deadline is unavoidable. However, one often sees that satellites continue working longer than this deadline. It would be a pity if the software was the first part to break down, so a system that lasts longer is preferred. How much longer? – Here, one cannot give a strict deadline, so a discounted property seems more appropriate.

Rewards in Markov chains can also be used to express quantitative properties [2]. DCTL still allows to combine multiple aspects of utility in ways that are not normally offered by analysis methods for chains with (multiple) rewards.

Acknowledgement. This work has been partially supported by DFG/NWO Bilateral Research Programme ROCKS. I thank Axel Legay (INRIA/IRISA Rennes, France) and Mariëlle Stoelinga (Universiteit Twente, The Netherlands) for helpful discussions, partly during a visit to INRIA/IRISA Rennes. I have used Maple [5] to check the example calculations.

References

1. de Alfaro, L., Faella, M., Henzinger, T. A., Majumdar, R., Stoelinga, M.: Model checking discounted temporal properties. *Theoretical Computer Science* 345(1), 139–170 (2005)
2. Baier, C., Cloth, L., Haverkort, B. R., Hermanns, H., Katoen, J.-P.: Performability assessment by model checking of Markov reward models. *Formal Methods in System Design* 36(1), 1–36 (2010)
3. Baier, C., Hermanns, H., Katoen, J.-P., Haverkort, B. R.: Efficient computation of time-bounded reachability probabilities in uniform continuous-time Markov decision processes. *Theoretical Computer Science* 345(1), 2–26 (2005)
- 3a. Baier, C., Katoen, J.-P., Hermanns, H., Wolf, V.: Comparative branching-time semantics for Markov chains. *Information and computation* 200(2), 149–214 (2005)

4. Jansen, D. N.: More or less true: DCTL for continuous-time MDPs. In: Braberman, V., Fribourg, L. (eds.) FORMATS 2013. LNCS, vol. 8053, pp. 137–151. Springer, Heidelberg (2013)
5. Maplesoft: technical computing software for engineers, mathematicians, scientists, instructors and students. <http://www.maplesoft.com/>
6. Neuhäüßer, M. R., Stoelinga, M., Katoen, J.-P.: Delayed nondeterminism in continuous-time Markov decision processes. In: de Alfaro, L. (ed.) FOSSACS 2009. LNCS, vol. 5504, pp. 364–379. Springer, Berlin (2009)
7. Wolovick, N., Johr, S.: A characterization of meaningful schedulers for continuous-time Markov decision processes. In: Asarin, E., Bouyer, P. (eds.) FORMATS 2006. LNCS, vol. 4202, pp. 352–367. Springer, Berlin (2006)
8. Zadeh, L. A.: Fuzzy sets, fuzzy logic, and fuzzy systems. World Scientific, Singapore (1996)
9. Zhang, L., Jansen, D. N., Nielson, F., Hermanns, H.: Efficient CSL model checking using stratification. Logical Methods in Computer Science 8(2) (2012), paper 17