Verification of Linear Duration Properties over Continuous Time Markov Chains

Taolue Chen, University of Oxford
Marco Diciolla, University of Oxford
Marta Kwiatkowska, University of Oxford
Alexandru Mereacre, University of Oxford

Stochastic modelling and algorithmic verification techniques have been proved useful in analysing and detecting unusual trends in performance and energy usage of systems such as power management controllers and wireless sensor devices. Many important properties are dependent on the cumulated time that the device spends in certain states, possibly intermittently. We study the problem of verifying *continuous-time Markov chains* (CTMCs) against *linear duration properties* (LDP), i.e. properties stated as conjunctions of linear constraints over the total duration of time spent in states that satisfy a given property. We identify two classes of LDP properties, *eventuality duration properties* (EDP) and *invariance duration properties* (IDP), respectively referring to the reachability of a set of goal states, within a time bound; and the continuous satisfaction of a duration property over an execution path. The central question that we address is how to compute the probability of the set of infinite timed paths of the CTMC that satisfy a given LDP. We present algorithms to approximate these probabilities up to a given precision, stating their complexity and error bounds. The algorithms mainly employ an adaptation of uniformisation and the computation of volumes of multi-dimensional integrals under systems of linear constraints, together with different mechanisms to bound the errors.

Categories and Subject Descriptors: D.2.4 [**Software Engineering**]: Software/Program Verification; G.3 [**Probability and Statistics**]

General Terms: Verification

Additional Key Words and Phrases: Model checking, Continuous time Markov chains, Linear duration properties, Markovian reward models

ACM Reference Format:
T. Chen, M. Diciolla, M.Z. Kwiatkowska, and A. Mereacre. 2013. Verification of Linear Duration Properties over Continuous Time Markov Chains. *ACM Trans. Comput. Logic* V, N, Article A (January YYYY), 36 pages.
ttDOI:http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Stochastic modelling and verification [Kwiatkowska et al. 2007] have become established as a means to analyse properties of system execution paths, for example dependability, performance and energy usage. Tools such as the probabilistic model checker PRISM [Kwiatkowska et al. 2011] have been applied to model and verify many systems, ranging from embedded controllers and nanotechnology designs to wireless sensor devices and cloud computing, in some cases identifying flaws or unusual quanti-

This work is supported by the ERC Advanced Grant VERIWARE and Oxford Martin School.
Author's addresses: T. Chen, M. Diciolla, M.Z. Kwiatkowska, and A. Mereacre, Department of Computer Science, University of Oxford, Wolfson Building, Parks Road, Oxford, OX1 3QD, United Kingdom.

ttDOI:http://dx.doi.org/10.1145/0000000.0000000

tative trends in system performance. The verification proceeds by subjecting a system model to algorithmic analysis against properties, typically expressed in probabilistic temporal logic, such as the probability of the vehicle hitting an obstacle is less than 0.0001, or the probability of an alarm bell ringing within 10 seconds is at least 95%. Many important properties, however, are dependent on the cumulated time that the system spends in certain states, possibly intermittently. Such *duration* properties, following the terminology of Duration Calculus (DC) [Zhou et al. 1991], have been studied in the context of timed automata (TAs) [Alur et al. 1997; Bouajjani et al. 1993; Kesten et al. 1999], but are not currently supported by existing probabilistic model checking tools. They can express, e.g., that the probability of an alarm bell ringing whenever the button has been pressed, possibly intermittently, for at least 2 seconds in total is at least 95%.

In this paper, we consider *Continuous-Time Markov Chain* (CTMC) models and study algorithmic verification for *linear duration properties* (LDP), i.e. properties involving linear constraints over cumulated residence time in certain states. CTMCs are widely used for performance and dependability analysis, aided by recent improvements [Baier et al. 2010]. CTMCs allow the modelling of real-time passage in conjunction with stochastic evolution governed by exponential distributions. They can be thought of as state transition systems, in which the system resides in a state on average for $1/r$ time units, where r is the exit rate, and transitions between the states are determined by a discrete probability distribution. As a concrete example of a system and property studied here, consider the dynamic power management system (DPMS) from [Qiu et al. 2001], analysed in [Norman et al. 2005] against properties such as average power consumption. The DPMS includes a queue of requests, which have an exponentially distributed inter-arrival time, a power management controller and a service provider. The power management controller issues commands to the service provider depending on the power management policy, which involves switching between different power-saving modes. Figure 1 (on page 5) depicts a CTMC model of the service provider for a Fujitsu disk drive. It consists of four states: *Busy*, *Idle*, *Standby* and *Sleep*. In this paper we are interested in computing the probability of, for instance, that *in 10 hours, the energy spent in the Standby state is less than the energy spent in the Sleep state and the energy spent in the Idle state is less than one third of the energy spent in the Busy state*. We remark that the restriction to exponential distributions is not critical, since one can approximate any distribution by phase-type distributions, resulting in series-parallel combinations of exponential distributions [Neuts 1981].

The focus of CTMC model checking has primarily been on algorithms for specifications expressed in stochastic temporal logics, including *branching-time* variants, such as CSL [Aziz et al. 2000; Baier et al. 2003; Zhang et al. 2012], as well as *linear-time* temporal logic (LTL), whose verification reduces to the same problem for *embedded* discrete-time Markov chains (DTMCs) [Courcoubetis and Yannakakis 1995]. Model checking *deterministic* TA (DTA) properties can be achieved by a reduction to computing the reachability probability in a *piecewise-deterministic Markov processes* (PDP, [Davis 1993]), based on the product construction between the CTMC and the DTA [Chen et al. 2009; Chen et al. 2011b; Barbot et al. 2011]. In [Chen et al. 2011a], *time-bounded* verification of properties expressed by *Metric Temporal Logic* (MTL) or general TAs, which allow *nondeterminism*, is formulated. Approximation algorithms are proposed, based on path exploration of the CTMC, constraints generation and reduction to volume computation. There, "time-bounded" refers to the fact that only timed paths over a time interval of fixed, bounded length are considered, e.g. the probability of an alarm bell ringing whenever the button has been pressed for at least 2 seconds continuously. However, as pointed out in [Alur et al. 1997], the expressiveness of DTA/MTL is limited and *cannot* express *duration-bounded* causality properties which constrain the accumulated satisfaction times of state predicates along an execution path, visited possibly intermittently.

Contributions. We consider *linear duration formulas* (LDF) expressed as finite conjunctions of linear constraints on the cumulated time spent in certain states of the CTMC, see Equation (1) (on page 7) for the precise formulation. Since we work with CTMCs, we interpret these formulas over finite and infinite *timed* paths. We distinguish two classes of linear duration properties. The difference lies only in how to interpret LDF over *infinite* timed paths. (Note that the LDF over *finite* timed paths is interpreted in a uniform way.)

— *Eventuality Duration Property (EDP).* Given a set of *goal* states G of the CTMC under consideration, an infinite path is said to satisfy LDF if its prefix until (the first occurrence of) G is reached satisfies EDP. This is similar to [Alur et al. 1997; Kesten et al. 1999]. Here, we also identify two variants, the time-bounded case and the unbounded case;
— *Invariance Duration Property (IDP).* For an infinite path to satisfy LDF, we require that *each* prefix of the infinite path satisfies LDF, again distinguishing the time-bounded case and the unbounded case. This is similar to [Bouajjani et al. 1993]. We remark that, in DC, a stronger requirement is imposed, i.e., any fragment (not only the prefix, but also starting from an arbitrary state) of the infinite path must satisfy LDF. We do not adopt this view, as we work in the traditional setting of temporal logics, rather than an interval temporal logic.

The central questions we consider is how to *compute the probability* of the set of timed paths of the CTMC which satisfy linear-time properties expressed as LDF. To the best of our knowledge, this is the first paper that considers algorithmic verification of duration properties for continuous-time stochastic models like CTMCs.

An extended abstract of the current paper has appeared in [Chen et al. 2012]. In addition to providing full proofs, more explanation and examples which are omitted from [Chen et al. 2012], this paper also includes new results, namely a sharpened error bound (cf. Section 3.2), and an extension to prefix-accumulation assertions in the CTMC setting (cf. Section 5). We now give a brief account of the techniques introduced in this paper. We propose two approaches to verify the time-bounded variant of EDP. First, we define a system of partial differential equations (PDEs) and a system of integral equations whose solutions capture the probability that an EDP is satisfied on a given CTMC. Second, we leverage the uniformisation method [Jensen 1953], which reduces the problem to computing the probability of a set of finite timed paths under a system of linear constraints. This can be solved through the computation of volumes of convex polytopes. In the unbounded case, by exploiting techniques mainly from matrix theory and linear algebra, we show how to approximate the probability by choosing a sufficiently large time-bound. This is of independent interest, and can be used elsewhere, e.g., to improve our previous results [Chen et al. 2011b; Chen et al. 2011a]. To verify an IDP, in the unbounded case we perform a graph analysis of the CTMC according to the LDF, and thus obtain a variant of EDP, which can be solved by extending the approaches developed in the previous case. The time-bounded case can be tackled accordingly and is indeed easier.

We remark that LDPs are closely related to *Markovian Reward Models* (MRM, [Baier et al. 2000]), which are CTMCs augmented with multiple reward structures assigning real-valued rewards to each state in the model. A large variety of performability measures for MRMs can be expressed in continuous stochastic reward logic (CSRL, [Baier et al. 2000]). CSRL model checking for MRMs [Haverkort et al. 2002; Cloth 2006] involves time-bounded and/or reward-bounded reachability problems, which can be formulated in terms of model checking of LDP, over CTMCs, by treating the rewards in the MRM as coefficients of linear duration formulas. (This will be made clearer in Section 2.3.) We emphasise that, in contrast to [Baier et al. 2000; Haverkort et al. 2002; Cloth 2006], we allow the coefficients in LDF to be *negative*, and hence can deal with CSRL in MRMs with arbitrary rewards. The link to MRMs (with arbitrary rewards) is beneficial, as energy constraints [Bouyer et al. 2008; Bouyer et al. 2010] studied in TA can be naturally adapted to stochastic models (such as CTMCs), and can be solved by approaches presented in the current paper.

Related work. Algorithmic verification of duration properties has primarily been studied in the setting of TA, for instance [Alur et al. 1997; Bouajjani et al. 1993; Kesten et al. 1999]. Similarly to our setting, TA also admit the unfolding of the system into timed execution paths, except that we have to calculate the probability of the set of paths satisfying a given property, rather than quantifying over their existence. The "duration bounded reachability" problem of [Alur et al. 1997] can be viewed as a subclass of EDP, in view of the requirement that all coefficients appearing in the linear constraints are nonnegative. Reachability for *integral graphs* [Kesten et al. 1999] can be reduced to verification of EDP for TA, which is solved by mixed linear-integer programming. [Bouajjani et al. 1993] extended the branching real-time logic TCTL with duration constraints and studied response/persistence properties. For DC, which is based on interval temporal logic that differs from our setting, the focus has been on so called *linear durational invariants* (LDI, [Zhou et al. 1994]). Again, TA (and their subclasses or extensions) are considered, and different techniques are proposed, for instance, reduction to linear programming or CTL, and discretisation. We mention, e.g., [Li et al. 1997; Thai and Hung 2004; Zhang et al. 2008], which are specific to TA and cannot be adapted to CTMCs.

There is only scant work addressing probabilistic/stochastic extensions of DC. Simple Probabilistic Duration Calculus, interpreted over (finite-state) continuous semi-Markov processes, is introduced in [Hung and Zhou 1999], together with the associated axiomatic system, and applied to QoS contracts in [Guelev and Hung 2010]. However, algorithmic verification is not addressed. [Hung and Zhang 2007] studied verification problems of (subclasses of) LDI in the setting of probabilistic TA which only involves discrete probabilities. The technique is an adaption of discretisation for TA.

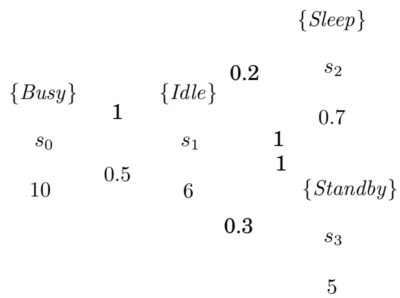
We also mention [Boker et al. 2011], which considers CTL and LTL extended with prefix-accumulation assertions for a quantitative extension of Kripke structures (i.e., weighted Kripke structures). (Un)decidability results are obtained. The prefix-accumulation assertions are similar to our linear constraints modulo the difference between models under consideration (CTMCs are a continuous-time model with randomisation, whereas Kripke structures are a discrete model without randomisation.) For further discussion, we refer the reader to Section 5.

Structure of the paper. This paper is organized as follows. Section 2 introduces basic definitions of CTMCs and duration properties. The relation between the CTMCs with duration property and MRMs is also discussed. Section 3 presents results on verification of EDP, while Section 4 presents results on IDP. Section 5 shows how to tackle extensions to the prefix-accumulation assertions. Section 6 concludes the paper.

2. PRELIMINARIES

2.1. Continuous-time Markov chains

Given a set $\mathcal{H}$, let Pr: $\mathcal{F}(\mathcal{H}) \to [0,1]$ be a *probability measure* on the measurable space $(\mathcal{H}, \mathcal{F}(\mathcal{H}))$, where $\mathcal{F}(\mathcal{H})$ is a σ-algebra over $\mathcal{H}$.



$$\mathbf{Q} = \begin{bmatrix} -10 & 10 & 0 & 0 \\ 3 & -6 & 1.2 & 1.8 \\ 0 & 0.7 & -0.7 & 0 \\ 0 & 5 & 0 & -5 \end{bmatrix}$$

Fig. 1. An example CTMC and its associated infinitesimal generator

Definition 2.1 (*CTMC*). A (labelled) continuous-time Markov chain (CTMC) is a tuple $\mathcal{C} = (S, \mathrm{AP}, L, \alpha, \mathbf{P}, E)$ where :

— S is a finite set of states;
— AP is a finite set of atomic propositions;
— $L : S \to 2^{\mathrm{AP}}$ is the labelling function;
— α is the initial distribution over S;
— $\mathbf{P} : S \times S \to [0, 1]$ is a stochastic matrix; and
— $E : S \to \mathbb{R}_{>0}$ is the exit rate function.

Example 2.2. An example CTMC is illustrated in Figure 1, where $\mathrm{AP} = \{\mathit{Busy}, \mathit{Idle}, \mathit{Sleep}, \mathit{Standby}\}$ and $\alpha(s_0) = 1$ is the initial distribution (in this case, a Dirac distribution). The exit rates are indicated at the states, whereas the transition probabilities are attached to the transitions. The CTMC is a model of the service provider of the DPMS system described in the introduction section of the paper.

In a CTMC $\mathcal{C}$, state residence times are *exponentially* distributed. More precisely, the residence time of the state $s \in S$ is a random variable governed by an exponential distribution with parameter $E(s)$. Hence, the probability to exit state s in t time units (t.u. for short) is given by $\int_0^t E(s) \cdot e^{-E(s)\tau} d\tau$; and the probability to take the transition from s to s' in t t.u. equals $\mathbf{P}(s, s') \cdot \int_0^t E(s) \cdot e^{-E(s)\tau} d\tau$. A state s is *absorbing* if $\mathbf{P}(s, s) = 1$. We also define the *infinitesimal generator* $\mathbf{Q}$ of $\mathcal{C}$ as

$$\mathbf{Q} = \mathbf{E} \cdot \mathbf{P} - \mathbf{E},$$

where $\mathbf{E}$ is the diagonal matrix with exit rates on diagonal. Occasionally we use $X(t)$ to denote the underlying *stochastic process* of $\mathcal{C}$.

We write $\pi(t)$ for the *transient probability distribution*, where, for each $s \in S$,

$$\pi_s(t) = \Pr\left(\{X(t) = s\}\right)$$

is the probability to be in state s at time t. It is well-known that $\pi(t)$ completely depends on the initial distribution α and the infinitesimal generator $\mathbf{Q}$, i.e., it is the solution of the Chapman-Kolmogorov equation

$$\frac{d\pi(t)}{dt} = \pi(t)\mathbf{Q}, \qquad \pi(0) = \alpha \ .$$

Note that efficient algorithms (e.g. the uniformisation approach, cf. Section 3.1.2, Equation (6)) exist to compute $\pi(t)$.

An *infinite timed* path in $\mathcal{C}$ is an infinite sequence

$$\rho = s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} s_2 \cdots \xrightarrow{t_{n-1}} s_n \ldots;$$

and a *finite timed* path is a finite sequence

$$\sigma = s_0 \xrightarrow{t_0} \cdots \xrightarrow{t_{n-1}} s_n.$$

In both cases we assume that $t_i \in \mathbb{R}_{>0}$ for each $i \geq 0$; moreover, we write $\rho[0..n]$ for σ. Below we usually follow the convention to let ρ (resp. σ) range over infinite (resp. finite) timed paths, unless otherwise stated. We define $|\sigma| := n$ to be the length of a finite timed path σ. For a finite or infinite path θ, $\theta[n] := s_n$ is the $(n+1)$-th state of θ and $\theta\langle n\rangle := t_n$ is the time spent in state s_n; let $\theta@t$ be the state occupied in θ at time $t \in \mathbb{R}_{\geq 0}$, i.e., $\theta@t := \theta[n]$, where n is the smallest index such that $\sum_{i=0}^{n} \theta\langle i\rangle > t$. Let $Paths^{\mathcal{C}}$ denote the set of infinite timed paths in $\mathcal{C}$, with abbreviation $Paths$ when $\mathcal{C}$ is clear from the context. Intuitively, a timed path ρ suggests that the CTMC $\mathcal{C}$ starts in state s_0 and stays in this state for t_0 t.u., and then jumps to state s_1, staying there for t_1 t.u., and then jumps to s_2 and so on. An example timed path is $\rho = s_0 \xrightarrow{3} s_1 \xrightarrow{2} s_0 \xrightarrow{1.5} s_1 \xrightarrow{3.4} s_2 \ldots$ with $\rho[2] = s_0$ and $\rho@4 = \rho[1] = s_1$.

Sometimes we refer to *discrete time Markov chains* (DTMCs), denoted by

$$\mathcal{D} = (S, \mathbf{AP}, \alpha, L, \mathbf{P}),$$

where the components of the tuple have the same meaning as those of CTMCs defined in Definition 2.1. In particular, we say such $\mathcal{D}$ is the *embedded DTMC* of the CTMC $\mathcal{C}$. Similarly, a (finite) *discrete* path $\varsigma = s_0 \to s_1 \to \ldots$ is a (finite) sequence of states; $\varsigma[n]$ denotes the state s_i, $\varsigma[0..n]$ denotes the prefix of length n of ς, and $|\varsigma|$ denotes the length of ς (in case that ς is finite). We also define $Paths^{\mathcal{D}}$ to be the set of all infinite paths of the DTMC $\mathcal{D}$. Given a finite discrete path $\varsigma = s_0 \to \cdots \to s_n$ of length n and $x_0, \ldots, x_{n-1} \in \mathbb{R}_{>0}$, we define $\varsigma[x_0, \ldots, x_{n-1}]$ to be the finite *timed* path σ such that $\sigma[i] := s_i$ and $\sigma\langle i\rangle := x_i$ for each $0 \leq i < n$. Let $\Gamma \subseteq \mathbb{R}^n_{>0}$, then

$$\varsigma[\Gamma] = \{\varsigma[x_0, \ldots, x_{n-1}] \mid (x_0, \ldots, x_{n-1}) \in \Gamma\}.$$

The definition of a *Borel space* on timed paths of CTMCs follows [Baier et al. 2003]. A CTMC $\mathcal{C}$ yields a probability measure $\Pr^{\mathcal{C}}_{\alpha}$ on $Paths^{\mathcal{C}}$ as follows. Let $s_0, \ldots, s_k \in S$ with $\mathbf{P}(s_i, s_{i+1}) > 0$ for $0 \leq i < k$ and $I_0, \ldots, I_{k-1}$ be nonempty intervals in $\mathbb{R}_{\geq 0}$. Let $C(s_0, I_0, \ldots, I_{k-1}, s_k)$ denote the *basic cylinder set* consisting of all $\rho \in Paths$ such that $\rho[i] = s_i$ $(0 \leq i \leq k)$ and $\rho\langle i\rangle \in I_i$ $(0 \leq i < k)$. $\mathcal{F}(Paths)$ is the smallest σ-algebra on $Paths$, which contains all sets $C(s_0, I_0, \ldots, I_{k-1}, s_k)$ for all state sequences $(s_0, \ldots, s_k) \in S^{k+1}$ with $\mathbf{P}(s_i, s_{i+1}) > 0$ for $(0 \leq i < k)$ and $I_0, \ldots, I_{k-1}$ ranging over all sequences of nonempty intervals in $\mathbb{R}_{\geq 0}$. The *probability measure* $\Pr^{\mathcal{C}}_{\alpha}$ on $\mathcal{F}(Paths)$ is the unique measure defined by induction on k by $\Pr^{\mathcal{C}}_{\alpha}(C(s_0)) = \alpha(s_0)$ and for $k > 0$:

$$\begin{aligned}&\Pr^{\mathcal{C}}_{\alpha}(C(s_0, I_0, \ldots, I_{k-1}, s_k)) = \\ &\qquad \Pr^{\mathcal{C}}_{\alpha}(C(s_0, I_0, \ldots, I_{k-2}, s_{k-1})) \cdot \int_{I_{k-1}} \mathbf{P}(s_{k-1}, s_k) E(s_{k-1}) \cdot e^{-E(s_{k-1})\tau} d\tau.\end{aligned}$$

Sometimes we write $\Pr$ instead of $\Pr^{\mathcal{C}}_{\alpha}$ when $\mathcal{C}$ and α are clear from the context. Elements of the σ-algebra denote *events* in the probability space. We now define two such events that will be needed later.

Definition 2.3. Given a CTMC $\mathcal{C}$ and $B \subseteq S$, we define:

— $\Diamond^{\leq T} B = \left\{\rho \in Paths^{C} \mid \exists n.\rho[n] \in B \text{ and } \sum_{i=0}^{n-1} \rho\langle i\rangle \leq T\right\}$, i.e., $\Diamond^{\leq T} B$ denotes the set of (infinite) timed paths which reach B in time interval $[0, T]$. Note that $\Pr^{\mathcal{C}}(\Diamond^{\leq T} B)$

can be computed by a reduction to the computation of the transient probability distribution; see [Baier et al. 2003].

— $\Diamond B = \{\rho \in Paths^C \mid \exists n.\rho[n] \in B\}$, i.e., $\Diamond B$ denotes the set of (infinite) timed paths which reach B. (This is the unbounded variant of $\Diamond^{\leq T} B$.) Note that $\Pr^{\mathcal{C}}(\Diamond B)$ is essentially the reachability probability of B in the embedded DTMC of $\mathcal{C}$; see [Baier et al. 2003].

For any two events Ξ_1 and Ξ_2, we write $\Pr(\Xi_1 \mid \Xi_2)$ for the conditional probability of Ξ_1 given Ξ_2, i.e.,

$$\Pr(\Xi_1 \mid \Xi_2) = \frac{\Pr(\Xi_1 \cap \Xi_2)}{\Pr(\Xi_2)}.$$

2.2. Duration Properties

We first introduce a language which includes the propositional calculus augmented with the *duration function* $\int$ and linear inequalities. In the remainder of this section, we assume a CTMC $\mathcal{C} = (S, \mathbf{AP}, L, \alpha, \mathbf{P}, E)$.

State formulas, are defined inductively as

$$\mathrm{sf} ::= ap \mid \neg\mathrm{sf} \mid \mathrm{sf}_1 \wedge \mathrm{sf}_2,$$

where $ap \in$ AP. Given a state formula sf and a state $s \in S$ we say that s satisfies the state formula sf, denoted $s \models \mathrm{sf}$, iff

$$\begin{array}{lcl} s \models ap & \Leftrightarrow & ap \in L(s) \\ s \models \neg\mathrm{sf} & \Leftrightarrow & s \not\models \mathrm{sf} \\ s \models \mathrm{sf}_1 \wedge \mathrm{sf}_2 & \Leftrightarrow & s \models \mathrm{sf}_1 \text{ and } s \models \mathrm{sf}_2 \end{array}$$

The *duration function* $\int$ is interpreted over a *finite* timed path. Let sf be a state formula and $\sigma = s_0 \xrightarrow{t_0} \ldots \xrightarrow{t_{n-1}} s_n$. The value of $\int \mathrm{sf}$ for σ, denoted $[\![\mathrm{sf}]\!]_\sigma$, is defined as $\sum_{\substack{0 \leq i < n, \\ \sigma[i] \models \mathrm{sf}}} t_i$. That is, the value of $\int \mathrm{sf}$ equals the sum of durations spent in states satisfying sf.

A *linear duration formula* (LDF) is of the form:

$$\varphi = \bigwedge_{j \in J} \left(\sum_{k \in K_j} c_{jk} \int \mathrm{sf}_{jk} \leq M_j \right), \tag{1}$$

where $c_{jk}, M_j \in \mathbb{R}$, sf_{jk} are state formulas, and J, K_j for $j \in J$ are finite index sets.

Remark 2.4. We did not introduce the *disjunction* or (more general) Boolean operators in Equation (1) for simplicity. All our results can be generalised to these cases by the *inclusion-exclusion principle*, paying the price of higher complexity.

Definition 2.5. Given a finite timed path $\sigma = s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} \ldots \xrightarrow{t_{n-1}} s_n$ and an LDF φ of the form defined in Equation (1), we write $\sigma \models \varphi$ if for each $j \in J$,

$$\sum_{k \in K_j} c_{jk} \cdot [\![\mathrm{sf}_{jk}]\!]_\sigma \leq M_j.$$

Example 2.6. For the CTMC in Figure 1, the LDF $\varphi = \int Idle - \frac{1}{3} \int Busy \leq 0$ expresses the constraint that during the evolution of the CTMC the accumulated time spent in the $Idle$ state must be less than or equal to one third of the accumulated time spent in the $Busy$ state.

Inspired by the notation of [Zhou et al. 1994], we shall also work on a slight extension of LDF, i.e., formulas of the form[1]:

$$\Phi := \int 1 \leq T \to \varphi,$$

where $T \in \mathbb{R}_{\geq 0} \cup \{\infty\}$. According to Definition 2.5, $\int 1$ denotes the total time spent on a finite timed path σ. Hence $\sigma \models \Phi$ if φ holds whenever the total time of σ is less or equal than T. Note that, if $T = \infty$, Φ simply degenerates to φ.

In general, given a CTMC and a duration property specified by an LDF, we are interested in computing the probability of *infinite timed* paths satisfying the LDF. We now generalise the satisfaction relation on finite paths, as defined in Definition 2.5, to *infinite* paths. Here we have two options, i.e., using the *finitary* and *infinitary* conditions. The former is motivated by standard automata theory, while the latter is natural when one thinks of "globally" (e.g., the $\Box$ operator in LTL).

Definition 2.7. Let $\rho = s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} \ldots$ be an *infinite* timed path and φ (or Φ) be an LDF.

— *Finitary satisfaction condition*. Given a set of *goal* states $G \subseteq S$, we write $\rho \models^G \varphi$ if there exists some $i \in \mathbb{N}$ such that:
 (1) $\rho[i] \in G$ and for any $0 \leq j < i$, $\rho[j] \notin G$; and
 (2) $\rho[0..i] \models \varphi$ (cf. Definition 2.5).
 Furthermore, we write $\rho \models^G_T \varphi$ for a given $T \in \mathbb{R}_{\geq 0}$ if, in addition to (1) and (2), $\sum_{j=0}^{i-1} \rho\langle j\rangle \leq T$ holds.

— *Infinitary satisfaction condition*. We write $\rho \models^\star \varphi$ if, for any $n \geq 0$, $\rho[0..n] \models \varphi$ (cf. Definition 2.5).

Problem statements. Corresponding to Definition 2.7, we focus on algorithmic verification problems for two classes of LDP, i.e., *Eventuality Duration Property* (EDP) and *Invariance Duration Property* (IDP), given below.

— *Verification of EDP*. Formally, given a CTMC $\mathcal{C}$, a set of *goal* states $G \subseteq S$, and an LDF $\Phi = \int 1 \leq T \to \varphi$, compute the probability of the set of infinite timed paths of $\mathcal{C}$ satisfying Φ under the *finitary satisfaction condition*. Depending on T, we distinguish two cases:
 — Time-bounded case: $T < \infty$, for which we denote the desired probability by $\mathrm{Prob}(\mathcal{C} \models^G \Phi)$
 — Unbounded case: $T = \infty$, for which we denote the desired probability by $\mathrm{Prob}(\mathcal{C} \models^G \varphi)$. Note that this is valid as, in this case, Φ is simply equivalent to φ.

 The algorithms for these two cases are given in Section 3.1 and Section 3.2, respectively.

— *Verification of IDP*. Formally, given a CTMC $\mathcal{C}$ and an LDF $\Phi = \int 1 \leq T \to \varphi$, compute the probability of the set of infinite timed paths of $\mathcal{C}$ satisfying Φ under the *infinitary satisfaction condition*. We also have two cases, i.e., the time-bounded case and unbounded case, which we denote by $\mathrm{Prob}(\mathcal{C} \models^\star \Phi)$ and $\mathrm{Prob}(\mathcal{C} \models^\star \varphi)$, respectively. The algorithms for these two cases are given in Section 4.2 and Section 4.1, respectively.

[1]Note that 1 denotes "true", $\to$ denotes "implication" and $\int 1 \leq T \to \varphi$ is a single formula.

2.3. Relationship to MRMs

In this section, we establish a link between the EDP of CTMC and the model of MRM. We start with some definitions.

Definition 2.8 (*MRM*). A (labelled) *Markovian reward model* $\mathcal{M}$ is a pair $(\mathcal{C}, \mathbf{r})$, where $\mathcal{C}$ is CTMC and $\mathbf{r} : S \to \mathbb{R}^d$ is a *reward structure* which assigns to each state $s \in S$ a vector of rewards $(r_1(s), \cdots, r_d(s))$.

Remark 2.9. The MRM defined in Definition 2.8 is more general than the one in [Baier et al. 2000], in the sense that we have *multiple* reward structures, and, more importantly, we allow arbitrary (instead of nonnegative) rewards associated with the states.

As mentioned in Section 1, the logic CSRL is introduced in [Baier et al. 2000]. The fundamental model checking problem for this logic (in particular, a sublogic called CRL) is the following *reward bounded* verification problem (which we extend to the multiple-reward setting, conforming to Definition 2.8): given a set of goal states G and a vector of reward bounds M_j, compute the probability of the paths which reach G and in which the j-th accumulated reward does not exceed M_j for each j. Below we show that this problem is essentially the same as EDP for CTMCs.

On the one hand, for a CTMC $\mathcal{C}$ and LDF φ, we construct an MRM $\mathcal{C}[\varphi]$. For every state $s_i \in S$, we define

$$r_{ji} = \sum_{\substack{t \in K_j, \\ s_i \models \mathrm{sf}_{jt}}} c_{jt}$$

for all $j \in J$. This yields a multiple reward structure $\mathbf{r}$ with $\mathbf{r}(s_i) = (r_{0i}, \cdots, r_{(|J|-1)i})$. Hence $\mathcal{C}[\varphi] = (\mathcal{C}, \mathbf{r})$. It is straightforward to see that the constraint expressed by LDF can be alternatively formulated as the "reward-bounded" constraint for MRMs, since $\sum\limits_{k \in K_j} c_{jk} \int \mathrm{sf}_{jk}$ essentially denotes the accumulated rewards along a finite timed path, and hence each M_j can be regarded as the bound of the reward.

On the other hand, given an MRM and a vector of reward bounds M_j for each reward structure, we construct an LDF φ as

$$\bigwedge_{j \in J} \sum_{s \in S} r_j(s) \int @s \leq M_j,$$

where $@s$ is an atomic proposition which holds exactly at state s. Hence, the reward-bounded verification problem for MRMs can be encoded into verification of linear duration properties in CTMCs.

It is straightforward to see that this correspondence, stated in the (time) unbounded case, can be adapted to the time-bounded case without any difficulties.

3. VERIFICATION OF EDP

In this section, we show how to verify EDP formulas. Throughout this section, we fix a CTMC $\mathcal{C} = (S, \mathbf{AP}, L, \alpha, \mathbf{P}, E)$, a set of goal states $G \subseteq S$, and an LDF

$$\Phi = \int 1 \leq T \to \underbrace{\bigwedge_{j \in J} (\sum_{k \in K_j} c_{jk} \int \mathrm{sf}_{jk} \leq M_j)}_{\varphi}.$$

3.1. Time-bounded Verification of EDP

Our task is to compute $\mathrm{Prob}(\mathcal{C} \models^G \Phi)$. First observe the following.

PROPOSITION 3.1. *Given a* CTMC $\mathcal{C}$ *and an LDF* Φ*, we have:*

$$\mathrm{Prob}(\mathcal{C} \models^G \Phi) = \Pr(\Diamond G) - \Pr(\Diamond^{\leq T} G) + \mathrm{Prob}(\mathcal{C} \models^G_T \varphi).$$

PROOF. We have that

$$\begin{aligned}\mathrm{Prob}(\mathcal{C} \models^G \Phi) &= \Pr(\{\rho \in \mathit{Paths}^{\mathcal{C}} \mid \rho \models^G \Phi\}) \\ &= \Pr\left(\left\{\rho \in \mathit{Paths}^{\mathcal{C}} \mid \rho \models^G \neg \left(\int 1 \leq T\right) \vee \varphi\right\}\right),\end{aligned}$$

where $\varphi = \bigwedge_{j\in J} (\sum_{k\in K_j} c_{jk} \int \mathrm{sf}_{jk} \leq M_j)$. We know that

$$\neg \left(\int 1 \leq T\right) \vee \varphi = \neg \left(\int 1 \leq T\right) \vee \left(\varphi \wedge \int 1 \leq T\right).$$

Therefore, we have

$$\begin{aligned}\mathrm{Prob}(\mathcal{C} \models^G \Phi) &= \Pr\left(\left\{\rho \in \mathit{Paths}^{\mathcal{C}} \middle| \rho \models^G \neg \left(\int 1 \leq T\right) \vee \left(\varphi \wedge \int 1 \leq T\right)\right\}\right) \\ &= \Pr\left(\left\{\rho \in \mathit{Paths}^{\mathcal{C}} \middle| \rho \models^G \neg \left(\int 1 \leq T\right) \vee \left(\rho \models^G \varphi \wedge \int 1 \leq T\right)\right\}\right) \\ &= \Pr\left(\left\{\rho \in \mathit{Paths}^{\mathcal{C}} \middle| \rho \models^G \neg \left(\int 1 \leq T\right)\right\}\right) \\ &\quad + \Pr\left(\left\{\rho \in \mathit{Paths}^{\mathcal{C}} \middle| \rho \models^G \varphi \wedge \int 1 \leq T\right\}\right) \\ &= \Pr(\Diamond G) - \Pr(\Diamond^{\leq T} G) + \mathrm{Prob}(\mathcal{C} \models^G_T \varphi).\end{aligned}$$

This completes the proof. □

Recall that $\Pr(\Diamond G)$ and $\Pr(\Diamond^{\leq T} G)$ can be easily computed (cf. Definition 2.3). Hence, the remainder of this section is devoted to computing:

$$\mathrm{Prob}(\mathcal{C} \models^G_T \varphi) := \Pr(\{\rho \mid \rho \models^G_T \varphi\}),$$

i.e., the probability of the set of paths of the CTMC $\mathcal{C}$ which reach G in time interval $[0, T]$ and satisfy the LDF φ before that happens; see Definition 2.7 (1).

3.1.1. PDE and Integral Equation Formulations. In order to compute $\mathrm{Prob}(\mathcal{C} \models^G_T \varphi)$, we shall use the link to MRMs established in Section 2.3. Recall that $\mathcal{C}[\varphi]$ is the MRM obtained from $\mathcal{C}$ and φ. We need an extra transformation over $\mathcal{C}[\varphi]$, namely, making each state $s \in G$ absorbing and setting $\mathbf{r}(s) = (0, \cdots, 0)$ (i.e., the rewards associated with s are all 0). We denote the resulting MRM $\mathcal{C}[\varphi, G]$. Recall that $X(t)$ is the underlying stochastic process of the CTMC $\mathcal{C}$. We denote by $\mathbf{Y}(T)$ the vector of accumulated rewards in the MRM $\mathcal{C}[\varphi]$ (see Section 2.3) up to time T, i.e.

$$\mathbf{Y}(T) = (Y_0(T), \ldots, Y_{|J|-1}(T)) = \int_0^T \mathbf{r}(X(\tau))d\tau$$

and thus each $Y_j(T)$ ($j \in J$) corresponds to a reward structure in $\mathcal{C}$. The vector of stochastic processes $\mathbf{Y}(T)$ is fully determined by $X(T)$ and the vector of reward structures of the state s_i is $\mathbf{r}(s_i) = (r_{0i}, \ldots, r_{(|J|-1)i})$.

Define $\mathbf{F}(T, \mathbf{y})$ to be the matrix of the joint probability distribution of state and rewards with entries $\mathbf{F}(T, \mathbf{y})[s, s'] = F_s^{s'}(T, \mathbf{y})$ for $s, s' \in S$ and

$$F_s^{s'}(T, \mathbf{y}) = \Pr\left(\left\{X(T) = s', \bigwedge_{j \in J} Y_j(T) \leq y_j \mid X(0) = s\right\}\right),$$

where $\mathbf{y} = (y_0, \cdots, y_{|J|-1})$. Note that we define $\mathbf{F}(T, \mathbf{y})$ over the induced MRM $\mathcal{C}[\varphi, G]$.

THEOREM 3.2. *Given a* CTMC $\mathcal{C}$*, an LDF* φ*, a vector* $\mathbf{M} = (M_0, \ldots, M_{|J|-1})$*, where each* M_j *is defined as in* φ *(cf. Equation* (1)*), and a set of goal states* G*, we obtain the induced* MRM $\mathcal{C}[\varphi, G]$*, and we have:*

$$\mathrm{Prob}(\mathcal{C} \models_T^G \varphi) = \sum_{s \in S} \sum_{s' \in G} \alpha(s) F_s^{s'}(T, \mathbf{M}).$$

PROOF. Let $s' \in G$ be an absorbing state with $\mathbf{r}(s) = (0, \cdots, 0)$. The probability to be in s' at time T is the same as the probability to reach s' before T (see [Baier et al. 2003]). Therefore, we have that:

$$\Pr(\{\rho \in Paths^{\mathcal{C}}(s) \mid \rho \models_T^{\{s'\}} \varphi\}) = \Pr\left(\left\{X(T) = s', \bigwedge_{j \in J} Y_j(T) \leq M_j \mid X(0) = s\right\}\right),$$

which directly follows from the construction in Section 2.3. □

Theorem 3.2 suggests a reduction to $\mathbf{F}(t, \mathbf{y})$, which we now characterise in terms of a system of PDEs.

THEOREM 3.3. *For an* MRM $\mathcal{C}[\varphi, G]$ *the function* $\mathbf{F}(t, \mathbf{y})$ *is given by the following system of PDEs:*

$$\frac{\partial \mathbf{F}(t, \mathbf{y})}{\partial t} + \sum_{j \in J} \mathbf{D}_j \cdot \frac{\partial \mathbf{F}(t, \mathbf{y})}{\partial y_j} = \mathbf{Q} \cdot \mathbf{F}(t, \mathbf{y}), \tag{2}$$

where $\mathbf{D}_j$ *is a diagonal matrix such that* $\mathbf{D}_j(s, s) = r_j(s)$*.*

PROOF. We want to calculate $F_s^{s'}(t, \mathbf{y})$. Assume that we are in state z at time Δt, for some small Δt. We consider three possible scenarios, and calculate the probability of each of them:

— No jumps before Δt;
— One jump before Δt;
— More than one jump before Δt.

No jumps before Δt. The probability of this scenario is:

$$(1 + Q(s, s)\Delta t) \cdot F_s^{s'}(t, \mathbf{y} - \mathbf{r}(s)\Delta t).$$

Here we indicate with $\mathbf{y} - \mathbf{r}(s)\Delta t$ the vector operation resulting in:

$$\mathbf{y} - \mathbf{r}(s)\Delta t = (y_0 - r_0(s)\Delta t, \ldots, y_{|J|-1} - r_{|J|-1}(s)\Delta t).$$

One jump before Δt. We denote the probability of being in state z at time Δt by $g_z(\Delta t)$. In order to derive the probability of this scenario we split it into three different cases:

(1) All rewards positive. Let
$$\mathbf{r}_{max} = (\max_{s\in S}\{r_0(s)\}, \ldots, \max_{s\in S}\{r_{|J|-1}(s)\})$$
and
$$\mathbf{r}_{min} = (\min_{s\in S}\{r_0(s)\}, \ldots, \min_{s\in S}\{r_{|J|-1}(s)\}).$$
The accumulated reward in Δt is at most $\mathbf{r}_{max}\Delta t$ and at least $\mathbf{r}_{min}\Delta t$. It follows that:
$$Q(s,z)\Delta t \cdot F_z^{s'}(t, \mathbf{y} - \mathbf{r}_{max}\Delta t) \leq g_z(\Delta t) \leq Q(s,z)\Delta t \cdot F_z^{s'}(t, \mathbf{y} - \mathbf{r}_{min}\Delta t).$$

(2) All rewards negative. Let
$$\mathbf{r}_{max} = (\max_{s\in S}\{|r_0(s)|\}, \ldots, \max_{s\in S}\{|r_{|J|-1}(s)|\})$$
and
$$\mathbf{r}_{min} = (\min_{s\in S}\{|r_0(s)|\}, \ldots, \min_{s\in S}\{|r_{|J|-1}(s)|\}).$$
It follows that:
$$Q(s,z)\Delta t \cdot F_z^{s'}(t, \mathbf{y} - \mathbf{r}_{max}\Delta t) \leq g_z(\Delta t) \leq Q(s,z)\Delta t \cdot F_z^{s'}(t, \mathbf{y} - \mathbf{r}_{min}\Delta t).$$

(3) Mixed rewards. Let
$$\mathbf{r}_{max} = (\max_{s\in S}\{r_0(s)|r_0(s) \geq 0\}, \ldots, \max_{s\in S}\{r_{|J|-1}(s)|r_{|J|-1}(s) \geq 0\})$$
and
$$\mathbf{r}_{min} = (\min_{s\in S}\{r_0(s)|r_0(s) < 0\}, \ldots, \min_{s\in S}\{r_{|J|-1}(s)|r_{|J|-1}(s) < 0\}).$$
It follows that:
$$Q(s,z)\Delta t \cdot F_z^{s'}(t, \mathbf{y} - \mathbf{r}_{max}\Delta t) \leq g_z(\Delta t) \leq Q(s,z)\Delta t \cdot F_z^{s'}(t, \mathbf{y} - \mathbf{r}_{min}\Delta t).$$

In all three cases above, note that
$$\lim_{\Delta t\to 0} \frac{g_z(\Delta t)}{\Delta t} = Q(s,z)F_z^{s'}(t, \mathbf{y}).$$

More than one jump before Δt. The probability of this scenario is negligible i.e., $o(\Delta t)$. Note that $\lim_{\Delta t\to 0} \frac{o(\Delta t)}{\Delta t} = 0$.

The joint distribution is given by
$$F_s^{s'}(t + \Delta t, \mathbf{y}) = (1 + Q(s,s)\Delta t) \cdot F_s^{s'}(t, \mathbf{y} - \mathbf{r}(s)\Delta t) + \sum_{z\neq s} g_z(\Delta t) + o(\Delta t).$$

From here on we derive the equations for $F_s^{s'}(\cdot)$ only for nonzero rewards. It can be extended to the general case. Let $|\mathbf{y}| = |J|$ be the cardinality of $\mathbf{y}$. We rewrite $F_s^{s'}(t, \mathbf{y})$ as $F_s^{s'}(t, y_0, \ldots, y_{|J|-1})$ to ease the notation and proofs. Given the above notation we can add and subtract terms from the joint distribution of $X(t)$ and $\mathbf{Y}(t)$ as follows:

$$
\begin{aligned}
F_s^{s'}(t+\Delta t,\mathbf{y}) &= F_s^{s'}(t,\mathbf{y}-\mathbf{r}(s)\Delta t) + Q(s,s)\Delta t \cdot F_s^{s'}(t,\mathbf{y}-\mathbf{r}(s)\Delta t) + \sum_{z\neq s} g_z(\Delta t) + o(\Delta t) \\
&= \left(F_s^{s'}(t,\mathbf{y}) - F_s^{s'}(t,\mathbf{y})\right) + F_s^{s'}(t,\mathbf{y}-\mathbf{r}(s)\Delta t) + Q(s,s)\Delta t \cdot F_s^{s'}(t,\mathbf{y}-\mathbf{r}(s)\Delta t) \\
&\quad + \sum_{z\neq s} g_z(\Delta t) + o(\Delta t)
\end{aligned}
$$

Let $\widehat{\mathbf{D}}(s)$ be a diagonal matrix such that $\widehat{\mathbf{D}}(s)[i,i] = r_i(s)$, for all $i \leq |J|-1$ such that $r_i(s) \neq 0$. Note that $\widehat{\mathbf{D}}(s)$ is invertible. We observe that

$$
\begin{aligned}
& F_s^{s'}(t+\Delta t,\mathbf{y}) - F_s^{s'}(t,\mathbf{y}) \\
&= F_s^{s'}(t,\mathbf{y}-\mathbf{r}(s)\Delta t) - F_s^{s'}(t,\mathbf{y}) + Q(s,s)\Delta t \cdot F_s^{s'}(t,\mathbf{y}-\mathbf{r}(s)\Delta t) \\
&\quad + \sum_{z\neq s} g_z(\Delta t) + o(\Delta t) \\
&= \widehat{\mathbf{D}}(s)^{-1} \cdot \widehat{\mathbf{D}}(s) \left(F_s^{s'}(t,\mathbf{y}-\mathbf{r}(s)\Delta t) - F_s^{s'}(t,\mathbf{y})\right) + Q(s,s)\Delta t \cdot F_s^{s'}(t,\mathbf{y}-\mathbf{r}(s)\Delta t) \\
&\quad + \sum_{z\neq s} g_z(\Delta t) + o(\Delta t),
\end{aligned}
$$

and

$$
\begin{aligned}
& \frac{F_s^{s'}(t+\Delta t,\mathbf{y}) - F_s^{s'}(t,\mathbf{y})}{\Delta t} \\
&= \widehat{\mathbf{D}}(s)^{-1} \cdot \widehat{\mathbf{D}}(s) \left(\frac{F_s^{s'}(t,\mathbf{y}-\mathbf{r}(s)\Delta t) - F_s^{s'}(t,\mathbf{y})}{\Delta t}\right) + Q(s,s) \cdot F_s^{s'}(t,\mathbf{y}-\mathbf{r}(s)\Delta t) \\
&\quad + \sum_{z\neq s} \frac{g_z(\Delta t)}{\Delta t} + o(\Delta t).
\end{aligned}
$$

Notice that all the three cases result in the same outcome. Taking the limit $\lim\limits_{\Delta t\to 0}$ and renaming the variables we obtain that

$$
\frac{\partial F_s^{s'}(t,\mathbf{y})}{\partial t} + \sum_{j\in J} r_j(s) \frac{\partial F_s^{s'}(t,\mathbf{y})}{\partial y_j} = \sum_{z\in S} Q(s,z) F_z^{s'}(t,\mathbf{y}).
$$

In matrix notation, one has

$$
\frac{\partial \mathbf{F}(t,\mathbf{y})}{\partial t} + \sum_{j\in J} \mathbf{D}_j \cdot \frac{\partial \mathbf{F}(t,\mathbf{y})}{\partial y_j} = \mathbf{Q} \cdot \mathbf{F}(t,\mathbf{y}),
$$

which completes the proof. □

Remark 3.4. The system of PDEs from Theorem 3.3 is a special case of the system of PDEs given in [Horton et al. 1998; Gribaudo and Telek 2007], which is presented for stochastic Petri nets.

Example 3.5. For the CTMC depicted in Figure 1, with $r(s_0) = 1$ and $r(s_1) = -1$, we can derive the following system of PDEs:

$$\begin{aligned}\frac{\partial F^{s_1}_{s_0}(t,y)}{\partial t} + \frac{\partial F^{s_1}_{s_0}(t,y)}{\partial y} &= 10F^{s_1}_{s_1}(t,y) - 10F^{s_1}_{s_0}(t,y),\\ \frac{\partial F^{s_0}_{s_1}(t,y)}{\partial t} - \frac{\partial F^{s_0}_{s_1}(t,y)}{\partial y} &= -6F^{s_0}_{s_1}(t,y) + 3F^{s_0}_{s_0}(t,y)\\ &\quad +1.2F^{s_0}_{s_2}(t,y) + 1.8F^{s_0}_{s_3}(t,y).\end{aligned}$$

Note that trivial equations like $0 = 0$ are simply omitted.

Next we provide an alternative characterisation of the joint probability distribution in terms of a system of integral equations, as follows.

THEOREM 3.6. *The solution of the system of PDEs in Equation* (2) *is the least fixpoint of the following system of integral equations:*

$$F^{s'}_s(t,\mathbf{y}) = e^{\mathbf{Q}(s,s)t}F^{s'}_s(0,\mathbf{y}-\mathbf{r}(s)t) + \int_0^t \sum_{z\neq s} e^{\mathbf{Q}(s,s)x}\mathbf{Q}(s,z)F^{s'}_z(t-x,\mathbf{y}-\mathbf{r}(s)x)dx.$$

PROOF. One possible solution for the hyperbolic system of PDEs obtained is the method of characteristics proposed in [Pattipati et al. 1993]. The method consists in finding the characteristic curves $\mathbf{y}(t)$ on which PDEs reduce to ODEs. Let $y(t)$ be an arbitrary curve and consider the derivative of $F^{s'}_s(t,\mathbf{y}(t))$ in t. More specifically,

$$\frac{dF^{s'}_s(t,\mathbf{y}(t))}{dt} = \frac{\partial F^{s'}_s(t,\mathbf{y}(t))}{\partial t}\frac{dt}{dt} + \frac{\partial F^{s'}_s(t,\mathbf{y}(t))}{\partial y}\frac{d\mathbf{y}(t)}{dt}.$$

Note that $\frac{dt}{dt} = 1$, then considering those curves $\mathbf{y}(t)$ such that $\frac{d\mathbf{y}(t)}{dt} = \mathbf{r}(s)$ yields

$$\frac{dF^{s'}_s(t,\mathbf{y}(t))}{dt} = \frac{\partial F^{s'}_s(t,\mathbf{y}(t))}{\partial t} + \sum_{j\in J}\frac{\partial F^{s'}_s(t,\mathbf{y}(t))}{\partial y_j}r_j(s) \tag{3}$$

Note here that the right-hand side of Equation (3) is the left-hand side of Equation (2), which implies that:

$$\frac{dF^{s'}_s(t,\mathbf{y}(t))}{dt} = \sum_{z\in S} Q(s,z)F^{s'}_z(t,\mathbf{y}(t)). \tag{4}$$

Equation (4) defines a system of ordinary differential equations that can be solved if we fix an initial value for $F^{s'}_s(0,\mathbf{y}(0))$. The solution is given by:

$$F^{s'}_s(t,\mathbf{y}(t)) = e^{Q(s,s)t}\left[\int_0^t e^{-Q(s,s)x}\sum_{z\neq s} Q(s,z)F^{s'}_z(x,\mathbf{y}(x))dx + F^{s'}_s(0,\mathbf{y}(0))\right]. \tag{5}$$

The curve $\mathbf{y}(t)$ defined by the ODE $\frac{d\mathbf{y}(t)}{dt} = \mathbf{r}(s)$ has as solution:

$$\mathbf{y}(t) = \mathbf{r}(s)t + \mathbf{C}.$$

We can calculate the value of $\mathbf{C}$, given a time t^* and the value $\mathbf{y}^*$ of the accumulated reward, by

$$\mathbf{C} = \mathbf{y}^* - \mathbf{r}(s)t^*.$$

In order to find the solution for the PDE in Equation (2) at a given t^* and $\mathbf{y}^*$, we solve the ODE in Equation (4) on the curve given by

$$\mathbf{y}(t) = \mathbf{r}(s)t + \mathbf{y}^* - \mathbf{r}(s)t^* = \mathbf{y}^* - \mathbf{r}(s)(t^* - t),$$

and more specifically, by substituting $x = t^* - x$:

$$F_s^{s'}(t^*, \mathbf{y}^*) = e^{Q(s,s)t^*} F_s^{s'}(0, \mathbf{y}^* - \mathbf{r}(s)t^*) + \int_0^{t^*} \sum_{z \neq s} e^{Q(s,s)x} Q(s,z) F_z^{s'}(t^* - x, \mathbf{y}^* - \mathbf{r}(s)x) dx.$$

This completes the proof. □

Remark 3.7. For readers who are familiar with PDP, Equation (2) can also be obtained as follows. For every state s of the CTMC we assign the system of differential equations: for each $j \in J$,

$$\frac{dx_j(t)}{dt} = r_j(s), \quad x_j(t) \in \mathbb{R}.$$

Note that $x_j(t)$ will denote the total accumulated reward at time t for the reward structure j. This results in a PDP with the state space $S \times \mathbb{R}^{|J|}$. The function $F_s^{s'}(t, \mathbf{y})$ represents the probability to reach the set of states $\{s'\} \times (-\infty, y_0] \times \cdots \times (-\infty, y_{|J|-1}]$ at time t.

Theorem 3.3 and Theorem 3.6 imply that, to solve the bounded-time EDP verification problem, we need to solve (first-order) PDEs or integral equations. However, this is usually costly and numerically unstable [Higham 2002]. We present solutions in the next section, based on uniformisation.

3.1.2. Uniformisation. In this section we present a uniformisation-based algorithm to compute $F_s^{s'}(t, \mathbf{y})$. The *uniformisation* method [Jensen 1953] involves transforming the CTMC $\mathcal{C}$ into a behaviorally equivalent DTMC $\mathcal{D}$. (NB this is *not* the embedded DTMC of $\mathcal{C}$.) The state space and initial distribution of $\mathcal{D}$ are the same as for $\mathcal{C}$. The probability matrix $\widehat{\mathbf{P}}$ of $\mathcal{D}$ is constructed by $\widehat{\mathbf{P}} = \mathbf{I} + \frac{1}{\Lambda}\mathbf{Q}$, where Λ is the maximal exit rate of $\mathcal{C}$. We obtain:

$$\pi(t) = e^{(\widehat{\mathbf{P}} - \mathbf{I})\Lambda t} = \sum_{n=0}^{\infty} \widehat{\mathbf{P}}^n \frac{(\Lambda t)^n}{n!} e^{-\Lambda t}. \tag{6}$$

We now apply the uniformisation technique to efficiently compute $F_s^{s'}(t, \mathbf{y})$. First, we note that the infinite sum in Equation (6) is equal to the probability $\frac{(\Lambda t)^n}{n!} e^{-\Lambda t}$ that exactly n Poisson arrivals occur in an interval of time $[0, t]$ multiplied with the probability $\widehat{\mathbf{P}}^n$ to take the state transitions corresponding to the arrivals. Then using Equation (6) we obtain:

$$F_s^{s'}(t, \mathbf{y}) = \sum_{n=0}^{\infty} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!} \cdot \left(\sum_{|\varsigma| = n} \Pr(\{\varsigma \mid X(0) = s\}) \cdot \Pr\left(\{X(n) = s', \mathbf{Y}(t) \leq \mathbf{y} \mid \varsigma\}\right) \right),$$

where for a given path $\varsigma = s \to s_1 \to \cdots \to s_{n-1} \to s_n$,

$$\mathrm{Prob}(\varsigma) := \Pr\left(\{\varsigma \mid X(0) = s\}\right) = \widehat{\mathbf{P}}(s, s_1) \times \cdots \times \widehat{\mathbf{P}}(s_{n-1}, s_n).$$

If $|\varsigma| = 0$ then $\mathrm{Prob}(\varsigma) := 1$. $\Pr(\{X(n) = s', \mathbf{Y}(t) \leq \mathbf{y} \mid \varsigma\})$ denotes the *conditional probability* that given the path ς at step n the state is s' and the total accumulated reward until time t is less than $\mathbf{y}$. The above equation can also be written as:

$$F_s^{s'}(t, \mathbf{y}) = \sum_{n=0}^{\infty} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!} \sum_{\substack{|\varsigma|=n,\\ \varsigma[0]=s,\\ \varsigma[n]=s'}} \mathrm{Prob}(\varsigma) \cdot \Pr(\{\mathbf{Y}(t) \leq \mathbf{y} \mid \varsigma\}). \tag{7}$$

Note that

$$\mathrm{Prob}(\varsigma) \cdot \Pr(\{\mathbf{Y}(t) \leq \mathbf{y} \mid \varsigma\}) = \Pr(\{\mathbf{Y}(t) \leq \mathbf{y} \wedge \varsigma\}). \tag{8}$$

Now the task is to compute $\Pr(\{\mathbf{Y}(t) \leq \mathbf{y} \wedge \varsigma\})$. for which we reduce to the computation of integration over a convex polytope. The basic idea is to generate timed constraints over variables determining the residence time of each state along ς to make $\mathbf{Y}(t) \leq \mathbf{y}$ hold. The desired probability can thus be formulated as a multidimensional integral, which can be computed by the efficient algorithm given in [Lasserre and Zeron 2001].

Given a *discrete* finite path ς of length k, an LDF φ, and a time-bound T, we define the set of linear constraints $\mathcal{S}$ generated in Algorithm 1. In Algorithm 1, line 3 generates the set of constraints from each conjunct in formula φ. In line 5 we add one more constraint to ensure that in the interval of time $[0, T]$ we will reach the last state of ς.

ALGORITHM 1: Generate a set of linear constraints $\mathcal{S}$ induced by φ, ς and T

Input: LDF φ, a path ς of length k and a time-bound T
Output: A set of linear constraints $\mathcal{S}$

1 $\mathcal{S} = \varnothing$;
2 **for** $j \in J$ **do**
3 $\quad \mathcal{S} = \mathcal{S} \cup \left\{ \sum_{i \in K_j} c_{ji} \cdot \sum_{\substack{0 \leq \ell \leq k,\\ \varsigma[\ell] \models \mathrm{sf}_{ji}}} x_\ell \leq M_j \right\}$;
4 **end**
5 $\mathcal{S} = \mathcal{S} \cup \left\{ \sum_{i=0}^{k-1} x_i \leq T \right\} \cup \left\{ \sum_{i=0}^{k} x_i \geq T \right\}$;
6 $\mathcal{S} = \mathcal{S} \cup \{x_i > 0\}$ for all x_i;
7 **return** $\mathcal{S}$;

Example 3.8. Assume the LDF $\varphi = \int \mathit{Idle} - \frac{1}{3} \int \mathit{Busy} \leq 0 \wedge \int \mathit{Idle} - \frac{1}{4} \int \mathit{Sleep} \leq 0$, the discrete path $\varsigma = s_0 \to s_1 \to s_2 \to s_1 \to s_3$ and the time-bound $T = 6$. The set of linear constraints $\mathcal{S}$ generated by Algorithm 1 induced by ς, φ and T is:

$$\mathcal{S} = \begin{cases} -\frac{1}{3} \cdot x_0 + x_1 + 0 \cdot x_2 + x_3 \leq 0 \\ 0 \cdot x_0 + x_1 - \frac{1}{4} \cdot x_2 + x_3 \leq 0 \\ x_0 + x_1 + x_2 + x_3 \leq 6 \\ x_0, x_1, x_2, x_3 > 0 \end{cases}$$

LEMMA 3.9. *Assume a discrete path ς of the* CTMC $\mathcal{C}$, *an LDF φ and a time-bound T. Let $\mathcal{S}$ be the set of linear constraints obtained by Algorithm 1. Then*

$$\varsigma[x_0, \ldots, x_{k-1}] \models \left(\varphi \wedge \int 1 \leq T\right) \quad \textit{iff} \quad (x_0, \ldots, x_{k-1}) \textit{ satisfies the constraints in } \mathcal{S}.$$

PROOF. Let φ_j be the j-th conjunct of φ. It is easy to see that:

$$\varsigma[x_0,\ldots,x_{k-1}] \models \varphi_j \quad \text{iff} \quad (x_0,\ldots,x_{k-1}) \text{ satisfies the constraints in } \mathcal{S},$$

which follows from the definition of $\models$ (see Definition 2.5). Note that $\varsigma[x_0,\ldots,x_{k-1}] \models \int 1 \le t$ iff $\sum_{i=0}^{k-1} x_i \le t$ (see Definition 2.7), which proves the lemma. □

We define

$$\mathrm{Prob}(\varsigma[\mathcal{S}]) := \Pr\nolimits^{\mathcal{C}(\varsigma[0])}(\{\varsigma[x_0,\ldots,x_{k-1}] \mid (x_0,\ldots,x_{k-1}) \text{ satisfies the constraints in } \mathcal{S}\}).$$

For future use, declare the function $Volume\_int(\alpha,\varsigma,\mathcal{S})$ which, given an initial distribution α, a finite discrete path $\varsigma = s_0 \to \cdots \to s_k$ of length k and a set of linear constraints $\mathcal{S}$ over $x_0,\cdots,x_{k-1}$, returns

$$\alpha(s_0)\cdot\prod_{i=0}^{k-1} E(s_i)\cdot P(s_i,s_{i+1})\cdot \underbrace{\int\cdots\int}_{k}{}_{\mathcal{S}} \prod_{i=0}^{k-1} e^{-E(s_i)\tau_i}d\tau_i. \tag{9}$$

Evidently $\mathrm{Prob}(\varsigma[\mathcal{S}])$ is equal to $Volume\_int(\alpha,\varsigma,\mathcal{S})$. In [Lasserre and Zeron 2001] an algorithm based on the Laplace transform is proposed to compute certain multidimensional integrals over polytopes. In Equation 9, the integration is over $\mathcal{S}$, which is the intersection of hyperplanes (in terms of linear inequalities). Hence, the algorithm of [Lasserre and Zeron 2001] can be applied directly. The time complexity of solving the multidimensional integral is $\mathcal{O}(|J|^k)$. Recall that $|J|$ is the number of constraints and k is the number of variables. Note that we omit the simple constraints from Alg.1, line 5 and 6, when computing the complexity of the algorithm. The simple constraints denote a constant term in the overall complexity.

The following theorem concludes this section, showing that, in order to compute $\Pr(\{\mathbf{Y}(t) \le \mathbf{y} \wedge \varsigma\})$, one only needs to compute $\mathrm{Prob}(\varsigma[\mathcal{S}])$, where $\mathcal{S}$ is generated from Algorithm 1.

THEOREM 3.10. *Let ς be a discrete path of the* CTMC $\mathcal{C}$ *ending in G, $\mathcal{C}[\varphi,G]$ be the* MRM *induced by $\mathcal{C}$ and LDF φ, and $\mathcal{S}$ the set of linear constraints generated by ς, φ and time-bound t. We have that:*

$$\Pr\nolimits^{\mathcal{C}(s)[\varphi,G]}(\{\mathbf{Y}(t) \le \mathbf{y} \wedge \varsigma\}) = \mathrm{Prob}(\varsigma[\mathcal{S}]),$$

where $\mathbf{y} = \mathbf{M} = (M_0,\ldots,M_{|J|-1})$.

PROOF. Let $\mathcal{C}(s)$ be the CTMC $\mathcal{C}$ such that, for a given state $s \in S$, $\alpha(s) = 1$. From Theorem 3.2 we know that:

$$\Pr\nolimits^{\mathcal{C}(s)[\varphi,G]}(\{\mathbf{Y}(t) \le \mathbf{y}\}) = \Pr\nolimits^{\mathcal{C}(s)}(\{\rho \in Paths^{\mathcal{C}}(s) \mid \rho \models_t^G \varphi\}).$$

Let ς be a discrete path of length k such that $\varsigma[0] = s$. We have that:

$$\begin{aligned}
&\Pr\nolimits^{\mathcal{C}(s)[\varphi,G]}(\{\mathbf{Y}(t) \le \mathbf{y} \wedge \varsigma\}) = \\
&\Pr\nolimits^{\mathcal{C}(s)[\varphi,G]}\Bigg(\{X(t) = \varsigma[k], \mathbf{Y}(t) \le \mathbf{y} \wedge \\
&\qquad\qquad \exists z_0,\ldots,z_{k-1}, 0 \le z_0 < z_1 < \cdots < z_{k-1} < t, X(0) = s, \bigwedge_{i=0}^{k-1} X(z_i) = \varsigma[i]\}\Bigg) = \\
&\Pr\nolimits^{\mathcal{C}(s)[\varphi,G]}\left(\{\rho \in Paths^{\mathcal{C}}(s) \mid \rho \models_t^{\varsigma[k]} \varphi, \bigwedge_{i=0}^{k-1} \rho[i] = \varsigma[i]\}\right).
\end{aligned}$$

From Lemma 3.9 we obtain:

$$\mathrm{Prob}(\varsigma[\mathcal{S}]) = \mathrm{Pr}^{\mathcal{C}(\varsigma[0])}\left(\left\{\rho \in \mathit{Paths}^{\mathcal{C}} \mid \varsigma[\rho\langle 0\rangle, \ldots, \rho\langle k-1\rangle] \models \varphi \wedge \int 1 \leq t\right\}\right).$$

One can easily see that:

$$\mathrm{Pr}^{\mathcal{C}(s)[\varphi,G]}\left(\left\{\rho \in \mathit{Paths}^{\mathcal{C}}(s) \mid \rho \models_t^{\varsigma[k]} \varphi, \bigwedge_{i=0}^{k-1} \rho[i] = \varsigma[i]\right\}\right) =$$
$$\mathrm{Pr}^{\mathcal{C}(\varsigma[0])}\left(\left\{\rho \in \mathit{Paths}^{\mathcal{C}} \mid \varsigma[\rho\langle 0\rangle, \ldots, \rho\langle k-1\rangle] \models \varphi \wedge \int 1 \leq t\right\}\right).$$

This completes the proof. $\square$

3.1.3. Algorithm. In order to compute $F_s^{s'}(t, \mathbf{y})$ we must pick a finite set $\mathcal{P}$ of paths from $\mathit{Paths}^{\mathcal{D}}$. Following [Qureshi and Sanders 1994], we introduce a threshold $w \in (0, 1)$ such that only if $\mathrm{Prob}(\varsigma) > w$ then $\varsigma \in \mathcal{P}$. This is mainly for efficiency considerations. We also fix a maximum length N for the paths in $\mathcal{P}$. Now we define

$$\mathcal{P}(s, s', w, N) := \{\varsigma \in \mathit{Paths}^{\mathcal{D}} \mid |\varsigma| = N, \varsigma[0] = s, \varsigma[N] = s', \mathrm{Prob}(\varsigma) > w\}.$$

We can approximate $F_s^{s'}(t, \mathbf{y})$ as

$$\widetilde{F_{N}^{w}}_{s}^{s'}(t, \mathbf{y}) = \sum_{n=0}^{N} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!} \sum_{\varsigma \in \mathcal{P}(s,s',w,n)} \mathrm{Prob}(\varsigma) \, \mathrm{Pr}\left(\{\mathbf{Y}(t) \leq \mathbf{y} \mid \varsigma\}\right),$$

where w and N are chosen as stated in Theorem 3.12. The approximation algorithm to compute $\mathit{Prob} := F_s^{s'}(t, \mathbf{y})$ is given in Algorithm 2.

ALGORITHM 2: Compute $\widetilde{F_{N}^{w}}_{s}^{s'}(t, \mathbf{y})$

$\mathit{Prob} = 0$;
$\mathit{Paths} = \{s\}$;
while $\mathit{Paths} \neq \varnothing$ **do**
  choose $\varsigma \in \mathit{Paths}$;
  $\mathit{Paths} = \mathit{Paths} \setminus \{\varsigma\}$;
  if $\mathrm{Prob}(\varsigma) > w$ *and* $|\varsigma| \leq N$ **then**
    if $\varsigma[|\varsigma|] = s'$ **then**
      $\mathit{Prob} += e^{-\Lambda t} \frac{(\Lambda t)^{|\varsigma|}}{|\varsigma|!} \mathrm{Prob}(\varsigma) \, \mathrm{Pr}\{\mathbf{Y}(t) \leq \mathbf{y} \mid \varsigma\}$;
    else
      for $s'' \in S$ **do**
        insert $(\varsigma \circ s'')$ into Paths;
      end
    end
  end
end
end
return Prob;

Note that $\circ$ represents the concatenation operator; $\varsigma[|\varsigma|]$ is the last state of ς.

Error bound. We give a bound for the truncation of the infinite sum to a finite one, considering only the discrete paths whose probability is greater than w. We start with the following technical lemma.

LEMMA 3.11. *Let $\varepsilon \in \mathbb{R}_{>0}$ and $T \in \mathbb{R}_{\geq 0}$. For any $N > \Lambda T e^2 + \ln(\frac{1}{\varepsilon})$, we have that*

$$\sum_{i=N+1}^{\infty} \frac{e^{-\Lambda T}(\Lambda T)^i}{i!} \leq \varepsilon.$$

PROOF. We have that

$$\begin{aligned}
&\sum_{i=N+1}^{\infty} \frac{e^{-\Lambda T}(\Lambda T)^i}{i!} = e^{-\Lambda T} \cdot \left(\sum_{i=N+1}^{\infty} \frac{(\Lambda T)^i}{i!} \right) \\
&\leq e^{-\Lambda T} \cdot e^{\Lambda T} \cdot \frac{(\Lambda T)^N}{N!} && \text{(Taylor expansion)} \\
&\leq \frac{(\Lambda T)^N}{(N/e)^N} = \left(\frac{\Lambda T e}{N} \right)^N && \text{(Stirling's approximation)} \\
&\leq \left(\frac{1}{e} \right)^N && (N > \Lambda T e^2) \\
&\leq \left(\frac{1}{e} \right)^{\ln(1/\varepsilon)} = \varepsilon. && (N > \ln(\frac{1}{\varepsilon}))
\end{aligned}$$

□

The following theorem states the error bound, which also suggests how to choose N and w for Algorithm 2 for a given ε.

THEOREM 3.12. *Given $\varepsilon > 0$, for $N > \Lambda t e^2 + \ln\left(\frac{1}{\varepsilon}\right)$, and $w < \frac{\varepsilon}{\sum_{n=0}^{N} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!}}$, we have that*

$$\left| F_s^{s'}(t, \mathbf{y}) - \widetilde{F^w_N}_s^{s'}(t, \mathbf{y}) \right| \leq 2\varepsilon.$$

PROOF.

$$\begin{aligned}
&\left| F_s^{s'}(t, y) - \widetilde{F^w_N}_s^{s'}(t, \mathbf{y}) \right| \\
&= \left| \sum_{n=0}^{\infty} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!} \cdot \sum_{\substack{|\varsigma|=n, \\ \varsigma[0]=s, \\ \varsigma[n]=s'}} \Pr(\{\varsigma\}) \cdot \Pr(\{\mathbf{Y}(t) \leq \mathbf{y} \mid \varsigma\}) - \right. \\
&\qquad \left. \sum_{n=0}^{N} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!} \cdot \sum_{\varsigma \in \mathcal{P}(s, s', w, n)} \Pr(\{\varsigma\}) \Pr(\{Y(t) \leq \mathbf{y} \mid \varsigma\}) \right|
\end{aligned}$$

$$= \Bigg| \underbrace{\sum_{n=N+1}^{\infty} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!} \cdot \sum_{\substack{|\varsigma|=n,\\ \varsigma[0]=s,\\ \varsigma[n]=s'}} \Pr(\{\varsigma\}) \cdot \Pr\left(\{\mathbf{Y}(t) \leq \mathbf{y} \mid \varsigma\}\right)}_{(\star)}$$

$$+ \underbrace{\sum_{n=0}^{N} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!} \cdot \sum_{\substack{|\varsigma|=n,\\ \varsigma[0]=s,\\ \varsigma[n]=s'}} \Pr(\{\varsigma\}) \cdot \Pr\left(\{\mathbf{Y}(t) \leq \mathbf{y} \mid \varsigma\}\right) -}_{(\star\star)}$$

$$\underbrace{\sum_{n=0}^{N} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!} \cdot \sum_{\varsigma \in \mathcal{P}(s,s',w,n)} \Pr(\{\varsigma\}) \Pr\left(\{Y(t) \leq \mathbf{y} \mid \varsigma\}\right)}_{(\star\star)} \Bigg|$$

We bound term $(\star)$ and term $(\star\star)$ separately.

— First, for $N > \Lambda t e^2 + ln\left(\frac{1}{\epsilon}\right)$ and by Lemma 3.11:

$$(\star) \leq \sum_{n=N+1}^{\infty} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!} \leq \epsilon$$

— Second:

$$\begin{aligned}
(\star\star) &= \sum_{n=0}^{N} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!} \cdot \sum_{\varsigma \notin \mathcal{P}(s,s',w,n)} \Pr(\{\varsigma\}) \Pr\left(\{Y(t) \leq \mathbf{y} \mid \varsigma\}\right) \\
&\leq \sum_{n=0}^{N} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!} \cdot w \cdot \sum_{\varsigma \notin \mathcal{P}(s,s',w,n)} \Pr\left(\{Y(t) \leq \mathbf{y} \mid \varsigma\}\right) \\
&\leq w \cdot \sum_{n=0}^{N} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!}.
\end{aligned}$$

It follows that:

$$\left| F_s^{s'}(t, y) - \widetilde{F_{N}^{w}}_s^{s'}(t, \mathbf{y}) \right| \leq \left| \epsilon + w \cdot \sum_{n=0}^{N} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!} \right|.$$

Taking $w \leq \frac{\epsilon}{\sum_{n=0}^{N} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!}}$, we obtain:

$$\left| F_s^{s'}(t, y) - \widetilde{F_{N}^{w}}_s^{s'}(t, \mathbf{y}) \right| \leq 2\epsilon.$$

This completes the proof. $\square$

Complexity. We analyse the complexity of Algorithm 2. Recall that $|S|$ is the number of states of $\mathcal{C}$. Algorithm 2 is composed of two main steps: (1) find all paths of length at most N; and (2) for each of those paths, ς, compute $\Pr(\{\mathbf{Y}(t) \leq \mathbf{y} \mid \varsigma\})$.

THEOREM 3.13. *The complexity of Algorithm 2 is* $\mathcal{O}(|S|^N \cdot (|J| + |J|^N))$.

PROOF. The number of paths of length less than $N-1$, from standard graph theory, is at most $|S|^N$ (in case of fully connected CTMCs). Subsequently, for each of those $|S|^N$ paths, say ς, we have to compute $\Pr(\{\mathbf{Y}(t) \leq \mathbf{y} \mid \varsigma\})$. Using the approach that generates the set of linear constraints we have that the complexity to compute the volume of convex polytopes defined over N variables is $|J|^N$ (see [Lasserre and Zeron 2001]), whereas the complexity to generate the set of linear constraints is linear in the cardinality of J. Therefore, the total complexity of Algorithm 2 is $\mathcal{O}(|S|^N \cdot (|J| + |J|^N))$. $\square$

3.2. Unbounded Verification of EDP

In this section we show how to compute $\mathrm{Prob}(\mathcal{C} \models^G \varphi)$. The main idea is that we approximate $\mathrm{Prob}(\mathcal{C} \models^G \varphi)$ by $\mathrm{Prob}(\mathcal{C} \models^G_T \varphi)$ for a sufficiently large $T \in \mathbb{R}_{\geq 0}$. Hence, we reduce the problem to time-bounded verification of EDP, which has been solved in Section 3.1.

For this purpose, we first introduce some background from linear algebra and matrix theory. We write $\mathbf{A}$ for a square matrix, with $a_{ij} \in \mathbb{R}$ the element of the i'th row and j'th column of $\mathbf{A}$. $\mathbf{A}$ is a *nonnegative matrix* if for any i, j, $a_{ij} \geq 0$. We write $eig(\mathbf{A})$ to be the set of all eigenvalues of matrix $\mathbf{A}$, and $\rho(\mathbf{A}) = \max\{|\lambda| \mid \lambda \in eig(\mathbf{A})\}$ be the *spectral radius* of $\mathbf{A}$, i.e., the maximum module of the eigenvalues of $\mathbf{A}$.

Definition 3.14. Let $\mathbf{A}$ be a square matrix. The *logarithmic norm* of $\mathbf{A}$, denoted by $\mu(\mathbf{A})$, is defined as

$$\mu(\mathbf{A}) = \max\left\{\lambda \middle| \lambda \in eig\left(\frac{\mathbf{A} + \mathbf{A}^\top}{2}\right)\right\}.$$

Note that this is well defined; as $\frac{\mathbf{A}+\mathbf{A}^\top}{2}$ is a symmetric matrix, all the eigenvalues are reals.

Note that $\mu(\mathbf{A}) \leq \rho\left(\frac{\mathbf{A}+\mathbf{A}^\top}{2}\right)$ and $\rho(\mathbf{A}) = \rho(\mathbf{A}^\top)$.

Definition 3.15. Let $\mathbf{A}$ be a square matrix of dimension m. We call the graph $\mathcal{G}_\mathbf{A}$ of $\mathbf{A}$ the dependency graph where:

— the nodes of the graph are $\{1, \cdots, m\}$, and
— there is an edge from i to j iff $a_{ij} > 0$.

Let $\mathcal{G}_\mathbf{A}$ be a dependency graph. $\mathcal{G}_\mathbf{A}$ is called *strongly connected* if there is a path from each vertex in $\mathcal{G}_\mathbf{A}$ to every other vertex. The *strongly connected components* (SCCs) of $\mathcal{G}_\mathbf{A}$ are its maximal strongly connected subgraphs. Moreover, a matrix $\mathbf{A}$ is *irreducible* iff $\mathcal{G}_\mathbf{A}$ is strongly connected.

PROPOSITION 3.16 ([DAHLQUIST 1958]). *Let* $||\cdot||$ *be the spectral matrix norm,* α *be a vector with its associated Euclidean vector norm, and* $T \geq 0$*. It holds that:*

$$||\alpha \cdot e^{\mathbf{Q}T}|| \leq ||\alpha|| \cdot e^{\mu(\mathbf{Q})T}.$$

PROPOSITION 3.17 ([HORN AND JOHNSON 1990]). *Let* $\mathbf{A}$ *be an arbitrary matrix and* $\epsilon > 0$*, then there exists some induced matrix norm* $||\cdot||$ *such that:*

$$||\mathbf{A}|| \leq \rho(\mathbf{A}) + \epsilon.$$

Definition 3.18. An $m \times m$ *substochastic matrix* $\mathbf{A}$ is a nonnegative matrix with the following properties:

— for any $0 \leq i \leq m$, $\sum_{1 \leq j \leq m} a_{ij} \leq 1$; and
— there exists some $0 \leq i \leq m$, such that $\sum_{1 \leq j \leq m} a_{ij} < 1$.

LEMMA 3.19. *Let $\mathbf{A}$ be an $m \times m$ irreducible substochastic matrix. It holds that* $\rho(\mathbf{A}) < 1$.

PROOF. For any $1 \leq i \leq m$ let $r_i^{(n)} = \sum_{k=1}^{m} \mathbf{A}_{ik}^n$ be the i-th row sum of $\mathbf{A}^n$. For $n = 1$ we write r_i instead of $r_i^{(1)}$. Since $\mathbf{A}$ is substochastic we have that $0 \leq r_i \leq 1$ for any $1 \leq i \leq m$ and $r_j < 1$ for at least one $1 \leq j \leq m$. Note that for $n \geq 1$:

$$r_j^{(n)} = \sum_{k=1}^{m} \mathbf{A}_{jk}^n = \sum_{k=1}^{m} \mathbf{A}_{jk} r_k^{(n-1)} \leq \sum_{k=1}^{m} \mathbf{A}_{jk} = r_j < 1.$$

By irreducibility of $\mathbf{A}$, for any i there is l such that $\mathbf{A}_{ij}^l > 0$. In fact, given that $\mathbf{A}$ is a $m \times m$ matrix and $i \neq j$ then we can assume $l < m$. Thus, we have that:

$$r_i^{(m)} = \sum_{k=1}^{m} \mathbf{A}_{ik}^l r_k^{(m-l)} < r_i^{(l)} \leq 1.$$

By the Gershgorin circle theorem [Horn and Johnson 1990], we have that $\rho(A^m) < 1$. Hence $\rho(A) < 1$. $\square$

LEMMA 3.20. *Suppose that $\rho(\mathbf{A}) < 1$, then $\mu(\mathbf{A}) < 1$.*

PROOF. We know that $\mu(\mathbf{A}) \leq \rho\left(\frac{\mathbf{A}+\mathbf{A}^\top}{2}\right)$. For *any* induced matrix norm $||\cdot||$, it holds that:

$$\rho\left(\frac{\mathbf{A}+\mathbf{A}^\top}{2}\right) \leq \frac{1}{2}(||\mathbf{A}+\mathbf{A}^\top||) \leq \frac{1}{2}||\mathbf{A}|| + \frac{1}{2}||\mathbf{A}^\top||.$$

Let $\epsilon > 0$ then from Proposition 3.17 it holds that for some matrix norm $||\cdot||$:

$$\begin{aligned}
\mu(\mathbf{A}) &\leq \rho\left(\frac{\mathbf{A}+\mathbf{A}^\top}{2}\right) \\
&\leq \frac{1}{2}||\mathbf{A}|| + \frac{1}{2}||\mathbf{A}^\top|| \\
&\leq \frac{1}{2}\rho(\mathbf{A}) + \frac{1}{2}\epsilon + \frac{1}{2}\rho(\mathbf{A}^\top) + \frac{1}{2}\epsilon \\
&= \rho(\mathbf{A}) + \epsilon.
\end{aligned}$$

From Lemma 3.19 we know that $\rho(\mathbf{A}) < 1$ and so we can pick an ϵ such that $\rho(\mathbf{A})+\epsilon < 1$. It follows that $\mu(\mathbf{A}) < 1$. $\square$

Now fix the CTMC $\mathcal{C}$ and the set of goal states $G \subseteq S$ with $|G| = m$. Recall that $\mathbf{Q}$ is the infinitesimal generator of $\mathcal{C}$. As the first step, we identify the set of states $S_{>0} \subseteq S$ starting from which there is positive probability to reach G. This can be done through a graph analysis in a standard way, see [Baier and Katoen 2008, Ch. 10]. We still write

$\mathbf{Q}_{>0}$ for the principal submatrix of the infinitesimal generator $\mathbf{Q}$ corresponding to $S_{>0}$. We partition $\mathbf{Q}_{>0}$ as follows

$$\mathbf{Q}_{>0} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \tag{10}$$

where $\mathbf{Q}_1$ is the square matrix of size $(|S_{>0}| - m) \times (|S_{>0}| - m)$ denoting transitions between the set of transient states $s \in S_{>0} \setminus G$, $\mathbf{Q}_2$ is the matrix of size $(|S_{>0}| - m) \times m$ denoting transitions from the transient states to the set of goal states G and $\mathbf{0}$ is a matrix composed of zeros. The reader should note that, given any infinitesimal generator $\mathbf{Q}$, it is always possible to express $\mathbf{Q} = \Lambda(\mathbf{P} - \mathbf{I})$, where Λ is the maximal exit rate of $\mathcal{C}$, $\mathbf{I}$ is the identity matrix and $\mathbf{P} = (\mathbf{I} + \frac{\mathbf{Q}}{\lambda})$ is a stochastic matrix. In the sequel we indicate with $\mathbf{P}_1$ the principal submatrix of $\mathbf{P}$ corresponding to $\mathbf{Q}_1$. Abusing notation we indicate with $\mathbf{I}_1$ the identity matrix of the same size as $\mathbf{P}_1$.

We define a random variable $T_G : Paths^{\mathcal{C}} \to \mathbb{R}_{\geq 0}$ that will denote the first entrance time of G. More specifically, given a path ρ:

$$T_G(\rho) = \begin{cases} \infty & \forall j \in \mathbb{N}.\ \rho[j] \notin G \\ \sum\limits_{j=0}^{k-1} \rho\langle j\rangle & \text{o/w, where } k = \min\{l \mid \rho[l] \in G\} \end{cases}.$$

The following proposition states a helpful property of the "transient part" of the infinitesimal generator of $\mathcal{C}$, relying on Lemma 3.19 and Lemma 3.20. Note that [Etessami et al. 2012] contains a similar argument showing essentially the same result, although in a different context.

PROPOSITION 3.21.

$$\mu(\mathbf{Q}_1) < 0.$$

PROOF. We first focus our attention on $\mathbf{P}_1$, which is a substochastic matrix. Let $\mathcal{G}_{\mathbf{P_1}}$ be the dependency graph of $\mathbf{P}_1$. We consider the SCC-decomposition of $\mathcal{G}_{\mathbf{P_1}}$, and assume a topological ordering among SCCs $\{B_1, \cdots, B_k\}$ such that, for $i \in B_m$ and $i' \in B_{m'}$, the existence of an edge from i to i' implies that $m < m'$. By Lemma 3.19, we have the following property: for any $\ell \in \{1, \cdots, k\}$ and the principal submatrix corresponding to B_ℓ, written as $\mathbf{P}_1[B_\ell]$,

$$\rho(\mathbf{P}_1[B_\ell]) < 1. \tag{11}$$

Since $\mathbf{P}_1$ is a nonnegative matrix, we have that there exists a nonnegative eigenvector v associated with $\rho(\mathbf{P}_1)$, i.e.,

$$\mathbf{P}_1 v = \rho(\mathbf{P}_1)v$$

We observe that, for any index $1 \leq i \leq n$, if $v_i > 0$ then, for any j such that there is an edge from j to i, we have that:

$$\begin{aligned}
(\mathbf{P}_1 v)_j &= \sum_{1 \leq k \leq n} p_{jk} v_k \\
&= \sum_{\substack{1 \leq k \leq n, \\ k \neq i}} p_{jk} v_k + p_{ji} v_i \\
&\geq p_{ji} v_i \\
&> 0.
\end{aligned}$$

Since $(\mathbf{P}_1 v)_j = \rho(\mathbf{P}_1)v_j$, we obtain that $v_j > 0$. Repeating the same argument, we have that, for each SCC, if for *some* index i we have $v_i > 0$, then for *any* index i in this SCC, $v_i > 0$.

It follows that there must exist some SCC such that, for any index i in this SCC, we have $v_i > 0$. Let $\hbar$ be the maximum index for such an SCC. Consider the principal submatrix corresponding to $B_\hbar$. For each index $i \in B_\hbar$, we have that:

$$
\begin{aligned}
(\mathbf{P}_1 v)_i &= \sum_{1 \le j \le n} p_{ij} v_j \\
&= \sum_{\substack{1 \le j \le n, \\ j \in B_\hbar}} p_{ij} v_j + \sum_{\substack{1 \le j \le n, \\ j \notin B_\hbar}} p_{ij} v_j \\
&= \sum_{\substack{1 \le j \le n, \\ j \in B_\hbar}} p_{ij} v_j \\
&= \rho(\mathbf{P}_1) v_i .
\end{aligned}
$$

It follows that $\rho(\mathbf{P}_1[B_\hbar]) \ge \rho(\mathbf{P}_1)$. However, we also have $\rho(\mathbf{P}_1[B_\hbar]) \le \rho(\mathbf{P}_1)$ as $\mathbf{P}_1[B_\hbar]$ is a principal submatrix. Hence $\rho(\mathbf{P}_1[B_\hbar]) = \rho(\mathbf{P}_1)$. Therefore, $\rho(\mathbf{P}_1) < 1$ by Equation (11).

Now note that by Lemma 3.20 if $\rho(\mathbf{P}_1) < 1$ then $\mu(\mathbf{P}_1) < 1$. Moreover, $\mu(\mathbf{Q}_1) = \mu(\Lambda(\mathbf{P}_1 - \mathbf{I}_1))$ which in turn yields that $\mu(\mathbf{Q}_1) \le \Lambda(\mu(\mathbf{P}_1) - 1)$ since $\mu(\mathbf{I}_1) = 1$. Thus, $\mu(\mathbf{P}_1) < 1$ implies that $\mu(\mathbf{Q}_1) < 0$, which concludes the proof. $\square$

PROPOSITION 3.22. *For any $T \in \mathbb{R}_{\ge 0}$ it holds that:*

$$\Pr^{\mathcal{C}}(\{\rho \in \mathit{Paths}^{\mathcal{C}} \mid \rho \models \Diamond G \wedge T_G(\rho) > T\}) = \hat{\alpha} \cdot e^{\mathbf{Q}_1 T} \mathbf{e},$$

where $\hat{\alpha} = \alpha[1, \ldots . |S_{>0}| - m]$ and $\mathbf{e}$ is a vector assigning 1's to the goal states and 0's to all the other states.

PROOF. Proof in [Nielsen et al. 2010]. $\square$

Now we are in a position to state the main result of this section.

THEOREM 3.23. *Given $0 < \varepsilon < 1$ and $T > \frac{\ln(\varepsilon/\sqrt{|G|})}{\mu(\mathbf{Q}_1)}$:*

$$\mathrm{Prob}(\mathcal{C} \models^G \varphi) - \mathrm{Prob}(\mathcal{C} \models^G_T \varphi) \le \varepsilon.$$

PROOF. We have

$$
\begin{aligned}
&\mathrm{Prob}(\mathcal{C} \models^G \varphi) - \mathrm{Prob}(\mathcal{C} \models^G_T \varphi) \\
\le &\Pr^{\mathcal{C}}(\{\rho \in \mathit{Paths}^{\mathcal{C}} \mid \rho \models \Diamond G \wedge T_G(\rho) \ge T\}) \\
= &\hat{\alpha} \cdot e^{\mathbf{Q}_1 T} \mathbf{e} = ||\hat{\alpha} \cdot e^{\mathbf{Q}_1 T} \mathbf{e}|| && \text{(by Prop. 3.22)} \\
\le &||\hat{\alpha}|| \cdot e^{\mu(\mathbf{Q}_1) T} \cdot ||\mathbf{e}|| && \text{(by Prop. 3.16)} \\
\le &\varepsilon
\end{aligned}
$$

The correctness of the bound is guaranteed by Proposition 3.21. $\square$

Due to this theorem, given an error bound ε and a set of goal states G, we can pick a time bound T such that $T \geq \frac{\ln(\varepsilon/\sqrt{|G|})}{\mu(\mathbf{Q}_1)}$ and compute $\mathrm{Prob}(\mathcal{C} \models^G_T \varphi)$. For computing $\mu(\mathbf{Q}_1)$, we note that it only requires computing eigenvalues of the symmetrisation of $\mathbf{Q}_1$ for which efficient numerical algorithms exist.

Remark 3.24. This significantly improves a bound obtained in [Chen et al. 2012, Theorem 7, page 272] through the Markov inequality, i.e., $\sum\limits_{s\in S} \alpha(s)\frac{\mathbb{E}_s[T_G]}{\varepsilon}$. For sufficiently small ε, this is an exponential improvement.

4. VERIFICATION OF IDP

In this section, we tackle the problem of verification of IDP. Again, we fix a CTMC $\mathcal{C} = (S, \mathbf{AP}, L, \alpha, \mathbf{P}, E)$ and an LDF

$$\Phi = \int 1 \leq T \to \underbrace{\bigwedge_{j\in J} \left(\sum_{k\in K_j} c_{jk} \int \mathrm{sf}_{jk} \leq M_j \right)}_{\varphi}.$$

As highlighted in Section 2, we shall distinguish two cases according to whether T is finite or infinite. We firstly give some definitions and algorithms that are common to both cases.

Given φ, a *discrete* finite path ς of length k and a time-bound $T < \infty$, we define the set of linear constraints $\mathcal{S}$ as generated in Algorithm 3. Note that this is different from the constraints obtained from Algorithm 1 in the previous section.

ALGORITHM 3: Generate a set of linear constraints $\mathcal{S}$ induced by φ, ς and T

Input: LDF φ, a path ς of length k and a time-bound T
Output: A set of linear constraints $\mathcal{S}$

$\mathcal{S} = \varnothing$;
for $z = 0;\ \ z < k;\ \ z{+}{+}$ **do**
  for $j \in J$ **do**
    $\mathcal{S} = \mathcal{S} \cup \left\{ \sum\limits_{i\in K_j} c_{ji} \cdot \sum\limits_{\substack{0\leq \ell \leq z,\\ \varsigma[\ell]\models \mathrm{sf}_{ji}}} x_\ell \leq M_j \right\}$;
  end
end
$\mathcal{S} = \mathcal{S} \cup \left\{ \sum\limits_{i=0}^{k-1} x_i \leq T \right\} \cup \left\{ \sum\limits_{i=0}^{k} x_i \geq T \right\}$
; $\mathcal{S} = \mathcal{S} \cup \{x_i > 0\}$ for all x_i;
return $\mathcal{S}$;

Example 4.1. Let $\varphi = \int \mathit{Busy} - \int \mathit{Idle} \leq 0$ be an LDF and $\varsigma = s_0 \to s_1 \to s_0 \to s_1 \to s_3$. The set of linear constraints $\mathcal{S}$ induced by ς and φ is:

$$\mathcal{S} = \begin{cases} x_{00} \leq 0 \\ x_{00} - x_{01} \leq 0 \\ x_{00} - x_{01} + x_{02} \leq 0 \\ x_{00} - x_{01} + x_{02} - x_{03} \leq 0 \\ x_{00}, x_{01}, x_{02}, x_{03} > 0 \end{cases}$$

LEMMA 4.2. *Let ς be a finite path of the* CTMC $\mathcal{C}$*, φ be an LDF and T be a time-bound. Moreover, let $\mathcal{S}$ be the set of linear constraints obtained by Algorithm 3. Then*

$$\varsigma[x_0, \ldots, x_{n-1}] \models^\star \left(\varphi \wedge \int 1 \leq T\right) \quad \textit{iff} \quad (x_0, \ldots, x_{n-1}) \in \mathcal{S}.$$

PROOF. Let φ_j be the j-th conjunct of φ. From Definition 2.7 we have that

$$\varsigma[x_0, \ldots, x_{n-1}] \models^\star \varphi_j \quad \text{iff} \quad (x_0, \ldots, x_{n-1}) \in \mathcal{S} = \bigcup_{z=0}^{n-1} \left\{ \sum_{i \in K_j} c_{ji} \cdot \sum_{\substack{0 \leq \ell \leq z, \\ \varsigma[\ell] \models \mathrm{sf}_{ji}}} x_\ell \leq M_j \right\}.$$

Note that $\varsigma[x_0, \ldots, x_{n-1}] \models \int 1 \leq t$ iff $\sum_{i=0}^{n-1} x_i \leq t$ (see Definition 2.7), which proves the lemma. $\square$

We define $\mathrm{Prob}^\star(\varsigma[\mathcal{S}])$ to be

$$\Pr^{\mathcal{C}}(\{\rho \in \mathit{Paths}^{\mathcal{C}} \mid \exists\, (x_0, \ldots, x_{n-1}) \in \mathcal{S}.\ \rho[0..n] \in \varsigma[x_0, \ldots, x_{n-1}] \wedge \rho[0..n] \models^\star \varphi\}),$$

which can be computed by the function $Volume\_int(\alpha, \varsigma, \mathcal{S})$ (cf. Equation (9)), where $\mathcal{S}$ is the set of constraints generated from Algorithm 3. We now introduce an auxiliary definition for paths of CTMCs.

Definition 4.3. Given an infinite timed path ρ, an *absorbing* set of states G of the CTMC $\mathcal{C}$, and a time bound $T < \infty$, we write $\rho \models^\star_{G,T} \varphi$ if there exists some $n \in \mathbb{N}$ such that:

— $\rho[n] \in G$ and $\sum_{i=0}^{n} \rho\langle i \rangle \leq T$, and
— for each $0 \leq i \leq n$, $\rho[0..i] \models \varphi$.

Remark 4.4. Note that, as we assume that G is absorbing, the only difference between $\rho \models^\star_{G,T} \varphi$ and $\rho \models^G_T \varphi$ given in Definition 2.7 lies in that, here, we require that, for each $0 \leq i \leq n$, $\rho[0..i] \models \varphi$, whereas in Definition 2.7 we require that $\rho[0..n] \models \varphi$. This reflects the distinction between EDP and IDP.

Our task now is to approximate the probability $\mathrm{Prob}(\mathcal{C} \models^\star_{G,T} \varphi)$. For this purpose, we present Algorithm 4 which computes an approximation $\widetilde{\mathrm{Prob}}_N(\mathcal{C} \models^\star_{G,T} \varphi)$ of $\mathrm{Prob}(\mathcal{C} \models^\star_{G,T} \varphi)$ for a given N.

ALGORITHM 4: Compute $\widetilde{\mathrm{Prob}}_N(\mathcal{C} \models^\star_{G,T} \varphi)$

Input: A CTMC $\mathcal{C}$, an LDF formula φ, set of goal states G, time-bound T, and N

```
Prob = 0;
for ς ∈ Paths^D s.t. ∃i. ς[i] ∈ G and |ς| ≤ N do
    Generate S from φ, ς and T, by Algorithm 3;
    Prob+ = Volume_int(α, ς, S);
end
return Prob;
```

4.1. Verification of unbounded IDP

This section is devoted to computing $\mathrm{Prob}(\mathcal{C} \models^\star \varphi)$. For this purpose, we need to perform graph analysis of $\mathcal{C}$. We start with some standard definitions. Note that some of the notions on graphs are essentially the same as in Section 3.2; for readability we present them here in terms of CTMCs.

Definition 4.5 (*BSCC*). Assume a CTMC $\mathcal{C}$. A set of states $B \subseteq S$ is a *strongly connected component* (SCC) of $\mathcal{C}$ if, for any two states $s, s' \in B$, there exists a discrete path $\varsigma = s_0 \to s_1 \to \ldots \to s_n$ such that $s_i \in B$ for $0 \leq i \leq n$, $s_0 = s$ and $s_n = s'$. An SCC B is a *bottom strongly connected component* (BSCC) if no state outside B is reachable from any state in B.

Definition 4.6. Given a BSCC B of the CTMC $\mathcal{C}$ and an LDF φ, we say

— B is *bad* w.r.t. the j-th conjunct in φ, φ_j, if

$$\exists s \in B.\ \exists i \in K_j.\ s \models \mathrm{sf}_{ji} \wedge c_{ji} > 0$$

and otherwise B is *good* w.r.t. φ_j.

— B is *good* w.r.t. φ (written $B \models \varphi$) if B is good for each conjunct of φ; otherwise B is *bad* (written $B \not\models \varphi$).

LEMMA 4.7. *Given a* CTMC $\mathcal{C} = (S, \mathbf{AP}, L, \alpha, \mathbf{P}, E)$*, an LDF* φ *and a BSCC* B*, we have that, if* B *is bad, then* $\mathrm{Pr}^{\mathcal{C}}\left(\{\rho \mid \rho \models^\star \varphi\} \mid \Diamond B\right) = 0$.

PROOF. We have the following basic facts, which follow from ergodicity theorems related to stochastic processes (see [Meyn and Tweedie 1996]):

(1) Given a BSCC B, every state $s \in B$ is visited infinitely often with probability 1.
(2) Any path $\rho \in Paths^{\mathcal{C}}$ eventually reaches one of the BSCCs of $\mathcal{C}$.

Given the second fact we only need to prove that for a bad BSCC B it holds that $\mathrm{Pr}^{\mathcal{C}}\left(\{\rho \mid \rho \models^\star \varphi\} \mid \Diamond B\right) = 0$. We note that:

$$\mathrm{Pr}^{\mathcal{C}}(\{\rho \mid \rho \models^\star \varphi\} \mid \Diamond B) = \frac{\mathrm{Pr}^{\mathcal{C}}(\{\rho \mid \rho \models^\star \varphi\} \cap \Diamond B)}{\mathrm{Pr}^{\mathcal{C}}(\Diamond B)}.$$

Therefore, in order to prove that $\mathrm{Pr}^{\mathcal{C}}(\{\rho \mid \rho \models^\star \varphi\} \mid \Diamond B) = 0$, it is enough to show that $\{\rho \mid \rho \models^\star \varphi\} \cap \Diamond B = \varnothing$. We prove it by contradiction. First, observe that

$$\{\rho \mid \rho \models^\star \varphi\} \cap \Diamond B = \bigcap_{j \in J} \left(\{\rho \mid \rho \models^\star \varphi_j\} \cap \Diamond B\right),$$

where φ_j is the j-th conjunct of φ. Therefore, we will only show that $\{\rho \mid \rho \models^\star \varphi_j\} \cap \Diamond B = \varnothing$ for some $j \in J$. Let $\rho \in \{\rho \mid \rho \models^\star \varphi_j\} \cap \Diamond B$. Then $\rho \in \Diamond B$. Given that B is bad it holds that $\exists s \in B.\ \exists i \in K_j.\ \mathrm{sf}_{ji} \in L(s) \wedge c_{ji} > 0$. From the first fact we know that there exist infinitely many n such that $\rho[n] = s$. Therefore, we have that $c_{ji} \int \mathrm{sf}_{ji} \to \infty$. We also know that $\rho \models^\star \varphi_j$ iff $\forall n. \rho[0 \ldots n] \models \varphi$ or

$$\forall n. \sum_{k \in K_j} c_{jk} \sum_{\substack{0 \leq i' < n, \\ \rho[0 \ldots n] \models \mathrm{sf}_{jk}}} \rho[0 \ldots n]\langle i' \rangle \leq M_j. \tag{12}$$

Given that $i \in K_j$ and $c_{ji} \int \mathrm{sf}_{ji} \to \infty$, Equation (12) does not hold. Therefore, we have that $\rho \not\models^\star \varphi_j$, which is a contradiction. $\square$

Let $\mathbf{BSCC}$ be the set of all BSCCs in $\mathcal{C}$ and $\widetilde{\mathrm{BSCC}}$ be the set of all good BSCCs.

Definition 4.8. Given a CTMC $\mathcal{C} = (S, \mathbf{AP}, L, \alpha, \mathbf{P}, E)$ and an LDF φ, we define a new CTMC $\mathcal{C}^a = (S, \mathbf{AP}^a, L^a, \alpha, \mathbf{P}^a, E)$ as follows:

— $\mathbf{AP}^a = \mathbf{AP} \cup \{\bot\}$, where $\bot$ is fresh;
— for all $s \in B$ and $B \in \widetilde{\mathrm{BSCC}}$ make s absorbing and let $L^a(s) = L(s) \cup \{\bot\}$; and
— for all other states $s \notin B$, $B \in \widetilde{\mathrm{BSCC}}$ and $s' \in S$, let $\mathbf{P}^a(s, s') = \mathbf{P}(s, s')$, $L^a(s) = L(s)$.

Example 4.9. As an example consider the left CTMC $\mathcal{C}$ from Figure 2, in which there are two BSCCs $B_1 = \{s_4, s_5\}$ and $B_2 = \{s_1, s_2, s_3\}$. Moreover, assume that $B_1 \not\models \varphi$ and $B_2 \models \varphi$ for a given LDF φ. After applying Definition 4.8 to $\mathcal{C}$ we get $\mathcal{C}^a$ shown on the right, where the labels of the states s_1, s_2 and s_3 are augmented with the label $\{\bot\}$ and all the other labels are left unchanged.

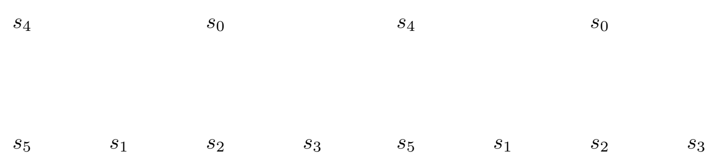


Fig. 2. Example of BSCC decomposition to demonstrate CTMC conversion in Definition 4.8

We now introduce an auxiliary definition, which, roughly, is the counterpart of (the unbounded version of) Definition 4.3.

Definition 4.10. Given an infinite timed path ρ and $G \subset S$, we write $\rho \models^\star_G \varphi$ if there exists some $n \in \mathbb{N}$ such that:

— $\rho[n] \in G$, and
— for each $0 \leq i \leq n$, $\rho[0..i] \models \varphi$.

The following proposition states that, in order to compute $\mathrm{Prob}(\mathcal{C} \models^\star \varphi)$, one can first make good BSCCs absorbing while removing bad BSCCs, and then reduce to computing $\mathrm{Prob}(\mathcal{C} \models^\star_G \varphi)$ for a suitable G, which, in turn, uses Algorithm 4.

PROPOSITION 4.11. *Given a* CTMC $\mathcal{C} = (S, \mathbf{AP}, L, \alpha, \mathbf{P}, E)$ *and an LDF* φ*, we have that*

$$\mathrm{Prob}(\mathcal{C} \models^\star \varphi) = \mathrm{Pr}^{\mathcal{C}^a}(\{\rho \mid \rho \models^\star_G \varphi\}),$$

where $G = \{s \in S \mid \bot \in L(s)\}$.

PROOF. Applying the law of total probability we have that

$$\begin{aligned}
& \mathrm{Pr}^{\mathcal{C}}(\{\rho \mid \rho \models^\star \varphi\}) \\
= & \sum_{B \in \mathbf{BSCC}} \mathrm{Pr}^{\mathcal{C}}\left(\{\rho \mid \rho \models^\star \varphi\} \mid \Diamond B\right) \cdot \mathrm{Pr}^{\mathcal{C}}(\Diamond B) \\
= & \sum_{B \in \widetilde{\mathrm{BSCC}}} \mathrm{Pr}^{\mathcal{C}}\left(\{\rho \mid \rho \models^\star \varphi\} \mid \Diamond B\right) \cdot \mathrm{Pr}^{\mathcal{C}}(\Diamond B) \quad \text{(by Lemma 4.7)} \\
= & \sum_{B \in \widetilde{\mathrm{BSCC}}} \mathrm{Pr}^{\mathcal{C}}\left(\{\rho \mid \rho \models^\star \varphi \wedge (\{\rho[0 \ldots n-1] \not\models \varphi\} \cup \{\rho[0 \ldots n-1] \models \varphi\})\} \mid \Diamond B\right) \cdot \mathrm{Pr}^{\mathcal{C}}(\Diamond B),
\end{aligned}$$

where for all $i < n$, $\rho[i] \notin B$. We have

$$\begin{aligned}
&\Pr^{\mathcal{C}}(\{\rho \mid \rho \models^{\star} \varphi\}) \\
=& \sum_{B \in \widetilde{\mathrm{BSCC}}} \Pr^{\mathcal{C}}\left(\{\rho \mid \rho \models^{\star} \varphi \wedge \rho[0 \ldots n-1] \not\models \varphi\} \mid \Diamond B\right) \cdot \Pr^{\mathcal{C}}(\Diamond B) \\
&+ \sum_{B \in \widetilde{\mathrm{BSCC}}} \Pr^{\mathcal{C}}\left(\{\rho \mid \rho \models^{\star} \varphi \wedge \rho[0 \ldots n-1] \models \varphi\} \mid \Diamond B\right) \cdot \Pr^{\mathcal{C}}(\Diamond B).
\end{aligned}$$

By definition of $\models^{*}$, $\Pr^{\mathcal{C}}\left(\{\rho \mid \rho \models^{\star} \varphi \wedge \rho[0 \ldots n-1] \not\models \varphi\} \mid \Diamond B\right) = 0$. Using similar reasoning as in Lemma 4.7, one can show that $\Pr^{\mathcal{C}}\left(\{\rho \mid \rho \models^{\star} \varphi \wedge \rho[0 \ldots n-1] \models \varphi\} \mid \Diamond B\right) = 1$, for any $B \in \widetilde{\mathrm{BSCC}}$. Therefore, we obtain that

$$\begin{aligned}
&\Pr^{\mathcal{C}}(\{\rho \mid \rho \models^{\star} \varphi\}) \\
=& \sum_{B \in \widetilde{\mathrm{BSCC}}} \Pr^{\mathcal{C}}(\Diamond B) = \sum_{B \in \widetilde{\mathrm{BSCC}}} \Pr^{\mathcal{C}}(\{\rho \mid \rho \models^{\star}_{B} \varphi\}) \\
=& \Pr^{\mathcal{C}}\left(\bigcup_{B \in \widetilde{\mathrm{BSCC}}} \{\rho \mid \rho \models^{\star}_{B} \varphi\}\right) \\
=& \Pr^{\mathcal{C}^a}(\{\rho \mid \rho \models^{\star}_{G} \varphi\}),
\end{aligned}$$

where $G = \bigcup_{B \in \widetilde{\mathrm{BSCC}}} \{s \in B\} = \{s \in S \mid \perp \in L(s)\}$ by Definition 4.8. □

ALGORITHM 5: Compute $\widetilde{\mathrm{Prob}}(\mathcal{C} \models^{\star} \varphi)$

Input: A CTMC $\mathcal{C}$, an LDF formula φ, ε_1 and ε_2

1 Identify all BSCCs B in $\mathcal{C}$;
2 $G = \varnothing$;
3 $\mathrm{Prob} = 0$;
4 **for** ***each*** *BSCC* B **do**
5   **if** $B \models \varphi$ **then**
6   Make every state in B absorbing;
7         $G = G \cup B$;
8   **end**
9 **end**
10 Choose $T \geq \frac{\ln(\varepsilon_1)}{\mu(\mathbf{Q}_1)}$ and $N \geq \Lambda T e^2 + \ln\left(\frac{1}{\varepsilon_2}\right)$;
11 $\mathrm{Prob} = \widetilde{\mathrm{Prob}}_N(\mathcal{C} \models^{\star}_{G,T} \varphi)$;
12 **return** Prob;
13 Recall that $\mu(\mathbf{Q}_1)$ denotes the logarithmic norm of $\mathbf{Q}_1$ (cf. Definition 3.14).

4.1.1. Algorithm. Algorithm 5 computes $\widetilde{\mathrm{Prob}}(\mathcal{C} \models^{\star} \varphi)$ which is an approximation of $\mathrm{Prob}(\mathcal{C} \models^{\star} \varphi)$. Lines 4-9 obtain $\mathcal{C}^a$ and the goal states G, according to Definition 4.8, and then the algorithm calls the function $\widetilde{\mathrm{Prob}}_N(\mathcal{C} \models^{\star}_{G,T} \varphi)$, given by Algorithm 4 (on page 26), by choosing T and N according to the specified error bounds ε_1 and ε_2 respectively.

Error bound. Intuitively there are two factors that contribute to the error introduced by Algorithm 5:

— the error introduced by approximating $\Pr^{\mathcal{C}^a}(\{\rho \mid \rho \models^{\star}_{G} \varphi\})$ with $\mathrm{Prob}(\mathcal{C}^a \models^{\star}_{G,T} \varphi)$, which can be obtained in a similar way as for Theorem 3.23, denoted by ε_1; and
— the error introduced by approximating $\mathrm{Prob}(\mathcal{C}^a \models^{\star}_{G,T} \varphi)$ with $\widetilde{\mathrm{Prob}}_N(\mathcal{C}^a \models^{\star}_{G,T} \varphi)$, denoted by ε_2.

THEOREM 4.12. *Given ε_1 and ε_2, we have that:*

$$\mathrm{Prob}(\mathcal{C} \models^{\star} \varphi) - \widetilde{\mathrm{Prob}}(\mathcal{C} \models^{\star} \varphi) \leq \varepsilon_1 + \varepsilon_2.$$

where $\widetilde{\mathrm{Prob}}(\mathcal{C} \models^{\star} \varphi)$ can be computed by Algorithm 5.

PROOF. The claim follows from Theorems 3.12, 3.23, and Proposition 4.11. □

Remark 4.13. Given ε *a priori*, one possibility is to let $\varepsilon_1 = \varepsilon_2 = \frac{\varepsilon}{2\sqrt{|G|}}$, and hence $T = \frac{\ln\left(\frac{\varepsilon}{2\sqrt{|G|}}\right)}{\mu(\mathbf{Q}_1)}$ and $N = \frac{\Lambda e^2 \ln\left(\frac{\epsilon}{2\sqrt{|G|}}\right)}{\mu(\mathbf{Q}_1)} + \ln(\frac{2\sqrt{|G|}}{\varepsilon})$ suffice.

4.2. Verification of Time-bounded IDP

In this section we show how to deal with the time-bounded variant of IDP. A well-known fact regarding CTMCs is that the set of Zeno paths is of probability 0, i.e.,

LEMMA 4.14. *Given a CTMC $\mathcal{C}$ and a time bound $T < \infty$, we have that:*

$$\Pr^{\mathcal{C}}\left(\left\{\rho \mid \rho \models^{\star} \int 1 \leq T\right\}\right) = 0.$$

We refer the readers to [Baier et al. 2003] for a proof.

For a CTMC $\mathcal{C}$, we write $\mathcal{C}[s]$ for the CTMC obtained from $\mathcal{C}$ by making the state s absorbing. The following theorem plays a pivotal role.

THEOREM 4.15. *Given a* CTMC $\mathcal{C}$ *and an LDF* Φ *it holds that:*

$$\mathrm{Prob}(\mathcal{C} \models^{\star} \Phi) = \sum_{s \in S} \mathrm{Prob}(\mathcal{C}[s] \models^{\star}_{\{s\},T} \varphi).$$

PROOF. By the law of total probability we have that:

$$\Pr^{\mathcal{C}}(\{\rho \mid \rho \models^{\star} \Phi\}) = \sum_{s \in S} \Pr^{\mathcal{C}}(\{\rho \mid \rho \models^{\star} \Phi\} \mid \{\rho \mid \rho @T = s\}) \cdot \Pr^{\mathcal{C}}(\{\rho \mid \rho @T = s\}),$$

since $\sum_{s \in S} \Pr^{\mathcal{C}}(\{\rho \mid \rho @T = s\}) = 1$. Observe that:

$$\begin{aligned}
& \Pr^{\mathcal{C}}(\{\rho \mid \rho \models^{\star} \Phi\} \mid \{\rho \mid \rho @T = s\}) \\
= & \frac{\Pr^{\mathcal{C}}(\{\rho \mid \rho \models^{\star} \Phi\} \bigcap \{\rho \mid \rho @T = s\})}{\Pr^{\mathcal{C}}(\{\rho \mid \rho @T = s\})} \\
= & \frac{\Pr^{\mathcal{C}}\{\{\rho \mid \forall i.\rho[0..i] \models \int 1 \leq T \rightarrow \varphi \text{ and } \rho @T = s\}\}}{\Pr^{\mathcal{C}}(\{\rho \mid \rho @T = s\})} \\
= & \frac{\Pr^{\mathcal{C}[s]}(\{\rho \mid \rho \models^{\star}_{\{s\},T} \varphi)}{\Pr^{\mathcal{C}}(\{\rho \mid \rho @T = s\})}.
\end{aligned}$$

Note that, for the last step, we use Lemma 4.14 and Definition 4.3. It follows that:

$$\begin{aligned}
&\Pr^{\mathcal{C}}(\{\rho \mid \rho \models^{\star} \Phi\}) \\
= &\sum_{s \in S} \frac{\Pr^{\mathcal{C}[s]}(\{\rho \mid \rho \models^{\star}_{\{s\},T} \varphi)}{\Pr^{\mathcal{C}}(\{\rho \mid \rho @ T = s\})} \cdot \Pr^{\mathcal{C}}(\{\rho \mid \rho @ T = s\}) \\
= &\sum_{s \in S} \mathrm{Prob}(\mathcal{C}[s] \models^{\star}_{\{s\},T} \varphi).
\end{aligned}$$

This completes the proof. □

The solution boils down to the computation of $\mathrm{Prob}(\mathcal{C}[s] \models^{\star}_{\{s\},T} \varphi)$ for each state s, for which we can apply Algorithm 4 for approximations. A detailed description is given in Algorithm 6.

ALGORITHM 6: Compute $\widetilde{\mathrm{Prob}}(\mathcal{C} \models^{\star} \Phi)$

Input: A CTMC $\mathcal{C}$, an LDF Φ and ε

$\mathrm{Prob} = 0$;

Chose $N \geq \Lambda T e^2 + \ln\left(\frac{|S|\cdot|\sqrt{G}|}{\varepsilon}\right)$;

for $s \in S$ **do**

  $\mathrm{Prob}{+}= \widetilde{\mathrm{Prob}}_N(\mathcal{C}[s] \models^{\star}_{\{s\},T} \varphi)$;

end

return Prob;

We also have the following error bound.

THEOREM 4.16. *Given ε and $N \in \mathbb{N}$, it holds that*

$$\mathrm{Prob}(\mathcal{C} \models^{\star} \Phi) - \widetilde{\mathrm{Prob}}(\mathcal{C} \models^{\star} \Phi) < \varepsilon.$$

PROOF. For each s, we compute $\mathrm{Prob}(\mathcal{C}[s] \models^{\star}_{\{s\},T} \varphi)$ up to $\frac{\varepsilon}{|S|\cdot\sqrt{|G|}}$. Namely, we choose N such that $N \geq \Lambda T e^2 + \ln\left(\frac{|S|\cdot\sqrt{|G|}}{\varepsilon}\right)$. It follows that

$$\mathrm{Prob}(\mathcal{C} \models^{\star} \Phi) - \widetilde{\mathrm{Prob}}(\mathcal{C} \models^{\star} \Phi) \leq |S| \cdot \frac{\varepsilon}{|S|} \leq \varepsilon.$$

This completes the proof. □

5. EXTENSIONS TO PREFIX-ACCUMULATION ASSERTIONS

In this section, we show how to extend our results to the *prefix-accumulation assertions* studied in [Boker et al. 2011]. Three prefix-accumulation assertions, namely `Sum` (summation), `Avg` (average) and `cAvg` (controlled accumulation) are introduced in [Boker et al. 2011] in the setting of *quantitative Kripke structures* (QKS). The idea is to adapt the construction used on QKSs to the settings of CTMCs. We first recall some definitions.

Definition 5.1 (*Quantitative Kripke structure*). A quantitative Kripke structure is a tuple $\mathcal{K} = (P, V, S, s_{in}, R, L)$ where:

— P is a finite set of Boolean variables;
— V is a finite set of numeric variables;

— S is a finite set of states, with initial state $s_{in} \in S$;
— $R \subseteq S \times S$ is a total transition relation; and
— $L : S \to 2^P \times \mathbb{Q}^V$ is a labelling function.

For the rest of this section, we fix a QKS $\mathcal{K} = (P, V, S, s_{in}, R, L)$. A computation of $\mathcal{K}$ is an infinite sequence of states $\pi = s_0, s_1, \ldots$ such that $s_0 = s_{in}$ and $(s_i, s_{i+1}) \in R$ for every $i \geq 0$. In the sequel, $[\![p]\!]_s \in \{\mathbf{T}, \mathbf{F}\}$ and $[\![v]\!]_s \in \mathbb{Q}$ respectively denote the value of a Boolean variable $p \in P$ and a numeric variable $v \in V$ in a state s of $\mathcal{K}$.

Definition 5.2 (***D-tree***). Given a finite set D of directions, a ***D-tree*** is a set $T \subseteq D^*$ such that, if $x \cdot d \in T$ where $x \in D^*$ and $d \in D$, then also $x \in T$. The elements of T are called *nodes*, and the empty word ε is the root of T. Thus, given two nodes x and y, we say that $x \leq y$ iff there is some $z \in D^*$ such that $y = x \cdot z$. For every $x \in T$, the nodes $x \cdot d$, for $d \in D$, are the successors of x. A node is a *leaf* if it has no successors. A *path* of T is a minimal set $\pi \subseteq T$ such that $\varepsilon \in \pi$ and for every $y \in \pi$, either y is a leaf or there exists a unique $d \in D$ such that $y \cdot d \in \pi$. For a set Z, a ***Z-labelled D-tree*** is a pair (T, τ) where T is a D-tree and $\tau : T \to Z$ maps each node of T to an element in Z.

The QKS $\mathcal{K}$ induces the *computation tree* $(T_{\mathcal{K}}, \tau_{\mathcal{K}})$ which corresponds to the computations of $\mathcal{K}$. Formally, $(T_{\mathcal{K}}, \tau_{\mathcal{K}})$ is a $(2^P \times \mathbb{Q}^V)$-labelled S-tree, where $\mathtt{state}(x)$ denotes the rightmost state in a node of x of $T_{\mathcal{K}}$ and $\tau_{\mathcal{K}}(x) = L(\mathtt{state}(x))$. The prefix-accumulation values (Sum and Avg) of a numeric variable v at a node x of $(T_{\mathcal{K}}, \tau_{\mathcal{K}})$ are the following:

— $[\![\mathtt{Sum}(v)]\!]_x = \sum\limits_{y \leq x} [\![v]\!]_y$, and
— $[\![\mathtt{Avg}(v)]\!]_x = \frac{[\![\mathtt{Sum}(v)]\!]_x}{|x|+1}$.

The same definition applies for Boolean variables by viewing them as numerical variables with $\mathbf{F} = 0$ and $\mathbf{T} = 1$.

The prefix-accumulation values Sum and Avg are fairly simple. In practice, one may wish to control and decide when the accumulation is done in order to take into account more complex behaviors. For this reason, [Boker et al. 2011] introduce the *controlled accumulation* $\mathtt{cAvg}(u, r_1, v, r_2)$, where u, v are numeric variables and r_1, r_2 are regular expressions over 2^P. The value of a *controlled accumulation* expression at a node x of the computation tree is defined as follows (we use $r(y)$ to indicate that the prefix y is a member in the language of the regular expression r):

$$[\![\mathtt{cAvg}(u, r_1, v, r_2)]\!]_x = \frac{\sum_{y \leq x | r_1(y)} [\![u]\!]_y}{\sum_{y \leq x | r_2(y)} [\![v]\!]_y}.$$

Intuitively, $\mathtt{cAvg}(u, r_1, v, r_2)$ considers the value of u accumulated only over the points in time where the regular expression r_1 is valid and it averages u against v, where v is the accumulated value of the variable v over the points in time where the regular expression r_2 is valid.

Example 5.3. Following the example in [Boker et al. 2011], we can express the average waiting time between a request (denote r) and a grant (denote g) over the alphabet Σ as $\mathtt{cAvg}(1, r_1, 1, r_2)$, where $r_1 = \Sigma^* r (\Sigma \backslash g)^*$ describes all the prefixes with a request that is not yet granted, and $r_2 = (\epsilon + \Sigma^* g)(\Sigma \backslash r)^* r$ describes all the prefixes in which a request that need a grant has been issued. Thus, $\mathtt{cAvg}(1, r_1, 1, r_2)$ is the sum of the waiting durations divided by the number of requests.

Below we show that prefix-accumulation assertions can be encoded by LDF in a precise sense. Hence, the elegant framework of [Boker et al. 2011] can also be adapted to our setting. For the two prefix-accumulation assertions $\mathtt{Sum}(v)$ and $\mathtt{Avg}(v)$ the translation is immediate. In fact, the term $\mathtt{Sum}(v)$ can be written as

$$\sum_{s \in S} v(s) \int @s,$$

where $@s$ is an atomic proposition (state formula) which holds exactly at state s. Similarly, the assertion $\mathtt{Avg}(v) \geq c$ can be encoded as

$$\sum_{s \in S} v(s) \int @s \geq c \cdot \int 1,$$

which is again an LDF after rearrangement.

The most interesting case is the controlled-average expression $\mathtt{cAvg}(u, r_1, v, r_2)$ for two numeric variables u, v and two regular expressions r_1, r_2. The idea is that we want to sum the value of u over all the points in time where r_1 is true and average this with v constrained to r_2.

First of all we construct two deterministic finite automata $\mathcal{A}_1$ and $\mathcal{A}_2$ out of r_1 and r_2, respectively. Then we build the product $\mathcal{C}' = \mathcal{C} \times \mathcal{A}_1 \times \mathcal{A}_2$. The product of a CTMC with a deterministic finite automaton is defined as follows.

Definition 5.4 (*Product* $\mathcal{C} \times \mathcal{A}$). Given a CTMC $\mathcal{C} = (S, \mathrm{AP}, L, s_0, \mathbf{P}, E)$ and a DFA $\mathcal{A} = (Q, 2^{\mathrm{AP}}, \delta, q_0, F)$ we define the product $\mathcal{C} \times \mathcal{A}$ to be the CTMC $\mathcal{C}' = (Loc, \mathrm{AP}', L', l_0, \mathbf{P}', E')$ where:

— $Loc = S \times Q$;
— $\mathrm{AP}' = \mathrm{AP} \cup \{p\}$;
— $l_0 = \langle s_0, q_0 \rangle$;
— given $l = \langle s, q \rangle$:
  — $L'(l) = L(s)$ if $q \notin F$
  — $L'(l) = L(s) \cup \{p\}$ if $q \in F$
— given $l_1 = \langle s_1, q_1 \rangle$ and $l_2 = \langle s_2, q_2 \rangle$, $\mathbf{P}'(l_1, l_2) = \mathbf{P}(s_1, s_2)$ iff:

$$\mathbf{P}(s_1, s_2) > 0 \wedge q_1 \xrightarrow{L(s_1)} q_2,$$

and $\mathbf{P}'(l_1, l_2) = 0$ otherwise.
— given $l_1 = \langle s_1, q_1 \rangle$ and $l_2 = \langle s_2, q_2 \rangle$, $E'(l_1, l_2) = E(s_1, s_2)$

where the label p indicates that the regular expression is true in the state labelled with it.

We focus on $\mathtt{cAvg}(u, p, v, q) \geq c$ instead of $\mathtt{cAvg}(u, r_1, v, r_2) \geq c$, where $p = \mathbf{T}$ in the states of $\mathcal{C}'$ where r_1 is true ($\mathbf{F}$ otherwise) and $q = \mathbf{T}$ in the states of $\mathcal{C}'$ where r_2 is true ($\mathbf{F}$ otherwise). We define a new reward structure, v', in $\mathcal{C}'$ as follows:

$$v' = \begin{cases} 0 & \text{if } p = \mathtt{false} \text{ and } q = \mathtt{false} \\ -cv & \text{if } p = \mathtt{false} \text{ and } q = \mathtt{true} \\ u & \text{if } p = \mathtt{true} \text{ and } q = \mathtt{false} \\ u - cv & \text{if } p = \mathtt{true} \text{ and } q = \mathtt{true} \end{cases}$$

Similarly to [Boker et al. 2011, Proposition 7], we have the following:

PROPOSITION 5.5. *For any* CTMC $\mathcal{C}$*, reward structures* u, v *and regular expressions* r_1, r_2*, the computation of* $\mathtt{cAvg}(u, r_1, v, r_2) \geq c$ *is reduced to the computation of* $\mathtt{Sum}(v') \geq 0$ *in* $\mathcal{C}'$.

6. CONCLUSION

We have studied the problem of verifying CTMCs against linear durational properties. We focused on two classes of LDPs, namely, eventuality duration properties and invariance duration properties. The central question we solved is, what is the probability of the set of infinite timed paths of the CTMC which satisfy the given LDP? We presented different algorithms to approximate these probabilities up to a given precision, stating their complexity and error bounds.

As future work, we plan to study algorithmic verification of more complex duration properties, for instance *response* and *persistence*, as in [Bouajjani et al. 1993]. It is also interesting to study specifications combining duration properties and temporal properties (in traditional real-time logics, e.g., MTL). The verification of these specifications would be challenging. Extending the current work to continuous-time Markov decision processes is another possible direction.

ACKNOWLEDGMENTS

We thank Vojtěch Forejt, Joost-Pieter Katoen, Dave Paker, Aistis Simaitis, anonymous reviewers of [Chen et al. 2012], and anonymous reviewers of the current paper for their inspiring discussions and for constructive, detailed comments.

REFERENCES

ALUR, R., COURCOUBETIS, C., AND HENZINGER, T. A. 1997. Computing accumulated delays in real-time systems. *Formal Methods in System Design 11,* 2, 137–155.

AZIZ, A., SANWAL, K., SINGHAL, V., AND BRAYTON, R. K. 2000. Model-checking continous-time Markov chains. *ACM Trans. Comput. Log. 1,* 1, 162–170.

BAIER, C., HAVERKORT, B. R., HERMANNS, H., AND KATOEN, J.-P. 2000. On the logical characterisation of performability properties. In *ICALP*, U. Montanari, J. D. P. Rolim, and E. Welzl, Eds. Lecture Notes in Computer Science Series, vol. 1853. Springer, 780–792.

BAIER, C., HAVERKORT, B. R., HERMANNS, H., AND KATOEN, J.-P. 2003. Model-checking algorithms for continuous-time Markov chains. *IEEE Trans. Software Eng. 29,* 6, 524–541.

BAIER, C., HAVERKORT, B. R., HERMANNS, H., AND KATOEN, J.-P. 2010. Performance evaluation and model checking join forces. *Commun. ACM 53,* 9, 76–85.

BAIER, C. AND KATOEN, J.-P. 2008. *Principles of Model Checking*. MIT Press.

BARBOT, B., CHEN, T., HAN, T., KATOEN, J.-P., AND MEREACRE, A. 2011. Efficient CTMC model checking of linear real-time objectives. In *TACAS*, P. A. Abdulla and K. R. M. Leino, Eds. Lecture Notes in Computer Science Series, vol. 6605. Springer, 128–142.

BOKER, U., CHATTERJEE, K., HENZINGER, T. A., AND KUPFERMAN, O. 2011. Temporal specifications with accumulative values. In *LICS*. IEEE Computer Society, 43–52.

BOUAJJANI, A., ECHAHED, R., AND SIFAKIS, J. 1993. On model checking for real-time properties with durations. In *LICS*. IEEE Computer Society, 147–159.

BOUYER, P., FAHRENBERG, U., LARSEN, K. G., AND MARKEY, N. 2010. Timed automata with observers under energy constraints. In *HSCC*, K. H. Johansson and W. Yi, Eds. ACM, 61–70.

BOUYER, P., FAHRENBERG, U., LARSEN, K. G., MARKEY, N., AND SRBA, J. 2008. Infinite runs in weighted timed automata with energy constraints. In *FORMATS*, F. Cassez and C. Jard, Eds. Lecture Notes in Computer Science Series, vol. 5215. Springer, 33–47.

CHEN, T., DICIOLLA, M., KWIATKOWSKA, M. Z., AND MEREACRE, A. 2011a. Time-bounded verification of CTMCs against real-time specifications. In *FORMATS*, U. Fahrenberg and S. Tripakis, Eds. Lecture Notes in Computer Science Series, vol. 6919. Springer, 26–42.

CHEN, T., DICIOLLA, M., KWIATKOWSKA, M. Z., AND MEREACRE, A. 2012. Verification of linear duration properties over continuous-time Markov chains. In *HSCC*, T. Dang and I. M. Mitchell, Eds. ACM, 265–274.

CHEN, T., HAN, T., KATOEN, J.-P., AND MEREACRE, A. 2009. Quantitative model checking of continuous-time Markov chains against timed automata specifications. In *LICS*. IEEE Computer Society, 309–318.

CHEN, T., HAN, T., KATOEN, J.-P., AND MEREACRE, A. 2011b. Model checking of continuous-time Markov chains against timed automata specifications. *Logical Methods in Computer Science 7,* 1–2, 1–34.

CLOTH, L. 2006. Model checking algorithms for Markov reward models. Ph.D. thesis, Univerisity of Twente, The Netherlands.

COURCOUBETIS, C. AND YANNAKAKIS, M. 1995. The complexity of probabilistic verification. *J. ACM 42,* 4, 857–907.

DAHLQUIST, G. 1958. Stability and error bounds in the numerical integration of ordinary differential equations. Ph.D. thesis, Stockholm College.

DAVIS, M. H. A. 1993. *Markov Models and Optimization*. Chapman and Hall.

ETESSAMI, K., STEWART, A., AND YANNAKAKIS, M. 2012. Polynomial time algorithms for branching Markov decision processes and probabilistic min(max) polynomial bellman equations. *CoRR abs/1202.4798*.

GRIBAUDO, M. AND TELEK, M. 2007. Fluid models in performance analysis. In *SFM*, M. Bernardo and J. Hillston, Eds. Lecture Notes in Computer Science Series, vol. 4486. Springer, 271–317.

GUELEV, D. P. AND HUNG, D. V. 2010. Reasoning about QoS contracts in the probabilistic duration calculus. *Electr. Notes Theor. Comput. Sci. 238,* 6, 41–62.

HAVERKORT, B. R., CLOTH, L., HERMANNS, H., KATOEN, J.-P., AND BAIER, C. 2002. Model checking performability properties. In *DSN*. IEEE Computer Society, 103–112.

HIGHAM, N. J. 2002. *Accuracy and Stability of Numerical Algorithms.* Second Ed. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

HORN, R. A. AND JOHNSON, C. R. 1990. *Matrix Analysis*. Cambridge University Press.

HORTON, G., KULKARNI, V. G., NICOL, D. M., AND TRIVEDI, K. S. 1998. Fluid stochastic petri nets: Theory, applications, and solution techniques. *European Journal of Operational Research 105,* 1, 184–201.

HUNG, D. V. AND ZHANG, M. 2007. On verification of probabilistic timed automata against probabilistic duration properties. In *RTCSA*. IEEE Computer Society, 165–172.

HUNG, D. V. AND ZHOU, C. 1999. Probabilistic duration calculus for continuous time. *Formal Asp. Comput. 11,* 1, 21–44.

JENSEN, A. 1953. Markoff chains as an aid in the study of Markoff processes. *Skand. Aktuarietidskrift 36*, 87–91.

KESTEN, Y., PNUELI, A., SIFAKIS, J., AND YOVINE, S. 1999. Decidable integration graphs. *Inf. Comput. 150,* 2, 209–243.

KWIATKOWSKA, M., NORMAN, G., AND PARKER, D. 2007. Stochastic model checking. In *Formal Methods for the Design of Computer, Communication and Software Systems: Performance Evaluation (SFM'07)*, M. Bernardo and J. Hillston, Eds. LNCS (Tutorial Volume) Series, vol. 4486. Springer, 220–270.

KWIATKOWSKA, M., NORMAN, G., AND PARKER, D. 2011. PRISM 4.0: Verification of probabilistic real-time systems. In *Proc. 23rd International Conference on Computer Aided Verification (CAV'11)*, G. Gopalakrishnan and S. Qadeer, Eds. LNCS Series, vol. 6806. Springer, 585–591.

LASSERRE, J. B. AND ZERON, E. S. 2001. A Laplace transform algorithm for the volume of a convex polytope. *J. ACM 48,* 6, 1126–1140.

LI, X., HUNG, D. V., AND ZHENG, T. 1997. Checking hybrid automata for linear duration invariants. In *ASIAN*, R. K. Shyamasundar and K. Ueda, Eds. Lecture Notes in Computer Science Series, vol. 1345. Springer, 166–180.

MEYN, S. P. AND TWEEDIE, R. L. 1996. *Markov Chains and Stochastic Stability*. Springer (3rd printing).

NEUTS, M. 1981. *Matrix-Geometric solutions in stochastic models: An algorithimic approach*. John Hopkins University Press.

NIELSEN, B. F., NIELSON, F., AND NIELSON, H. R. 2010. Model checking multivariate state rewards. In *QEST*. IEEE Computer Society, 7–16.

NORMAN, G., PARKER, D., KWIATKOWSKA, M., SHUKLA, S., AND GUPTA, R. 2005. Using probabilistic model checking for dynamic power management. *Formal Aspects of Computing 17,* 2, 160–176.

PATTIPATI, K. R., MALLUBHATLA, R., GOPALAKRISHNA, V., AND VISWANATHAM, N. 1993. Markov-reward models and hyperbolic systems. *2nd Int. Workshop Performability Modeling of Computer and Communication Systems*.

QIU, Q., QU, Q., AND PEDRAM, M. 2001. Stochastic modeling of a power-managed system-construction and optimization. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 20,* 10, 1200 –1217.

QURESHI, M. A. AND SANDERS, W. H. 1994. Reward model solution methods with impulse and rate rewards: an algorithm and numerical results. *Performance Evaluation 20,* 4, 413 – 436.

THAI, P. H. AND HUNG, D. V. 2004. Verifying linear duration constraints of timed automata. In *ICTAC*, Z. Liu and K. Araki, Eds. Lecture Notes in Computer Science Series, vol. 3407. Springer, 295–309.

ZHANG, L., JANSEN, D. N., NIELSON, F., AND HERMANNS, H. 2012. Efficient CSL model checking using stratification. *Logical Methods in Computer Science 8,* 2.

ZHANG, M., HUNG, D. V., AND LIU, Z. 2008. Verification of linear duration invariants by model checking CTL properties. In *ICTAC*, J. S. Fitzgerald, A. E. Haxthausen, and H. Yenigün, Eds. Lecture Notes in Computer Science Series, vol. 5160. Springer, 395–409.

ZHOU, C., HOARE, C. A. R., AND RAVN, A. P. 1991. A calculus of durations. *Inf. Process. Lett. 40,* 5, 269–276.

ZHOU, C., ZHANG, J., YANG, L., AND LI, X. 1994. Linear duration invariants. In *FTRTFT*, H. Langmaack, W. P. de Roever, and J. Vytopil, Eds. Lecture Notes in Computer Science Series, vol. 863. Springer, 86–109.