Abstraction for Stochastic Systems by Erlang's Method of Stages*

Joost-Pieter Katoen[1], Daniel Klink[1], Martin Leucker[2], and Verena Wolf[3]

[1] RWTH Aachen University
[2] TU Munich
[3] EPF Lausanne

Abstract. This paper proposes a novel abstraction technique based on Erlang's method of stages for continuous-time Markov chains (CTMCs). As abstract models *Erlang-k interval processes* are proposed where state residence times are governed by Poisson processes and transition probabilities are specified by intervals. We provide a three-valued semantics of CSL (Continuous Stochastic Logic) for Erlang-k interval processes, and show that both affirmative and negative verification results are preserved by our abstraction. The feasibility of our technique is demonstrated by a quantitative analysis of an enzyme-catalyzed substrate conversion, a well-known case study from biochemistry.

1 Introduction

This paper is concerned with a novel abstraction technique for timed probabilistic systems, in particular continuous-time Markov chains, CTMCs for short. These models are omnipresent in performance and dependability analysis, as well as in areas such as systems biology. In recent years, they have been the subject of study in concurrency theory and model checking. CTMCs are a prominent operational model for stochastic process algebras [13] and have a rich theory of behavioral (both linear-time and branching-time) equivalences, see, e.g., [4,26]. Efficient numerical, as well as simulative verification algorithms have been developed [1,3,27] and have become an integral part of dedicated probabilistic model checkers such as PRISM and act as backend to widely accepted performance analysis tools like GreatSPN and the PEPA Workbench.

Put in a nutshell, CTMCs are transition systems whose transitions are equipped with discrete probabilities and state residence times are determined by negative exponential distributions. Like transition systems, they suffer from the state-space explosion problem. To overcome this problem, several abstraction-based approaches have recently been proposed. Symmetry reduction [20], bisimulation minimization [16], and advances in quotienting algorithms for simulation pre-orders [28] show encouraging experimental results. Tailored abstraction techniques for regular infinite-state CTMCs have been reported [22], as well as bounding techniques that approximate CTMCs by ones having a special structure allowing closed-form solutions [21]. Predicate abstraction techniques have been extended to (among others) CTMCs [14]. There is a wide range of

* The research has been partially funded by the DFG Research Training Group 1298 (AlgoSyn), the Swiss National Science Foundation under grant 205321-111840 and the EU FP7 project Quasimodo.

F. van Breugel and M. Chechik (Eds.): CONCUR 2008, LNCS 5201, pp. 279–294, 2008.

related work on abstraction of discrete-time probabilistic models such as MDPs, see e.g., [9,8,19]. Due to the special treatment of state residence times, these techniques are not readily applicable to the continuous-time setting.

This paper generalizes and improves upon our three-valued abstraction technique for CTMCs [17]. We adopt a three-valued semantics, i.e., an interpretation in which a logical formula evaluates to either true, false, or indefinite. In this setting, abstraction preserves a simulation relation on CTMCs and is conservative for both positive and negative verification results. If the verification of the abstract model yields an indefinite answer, the validity in the concrete model is unknown. In order to avoid the grouping of states with distinct residence time distributions, the CTMC is *uniformized* prior to abstraction. This yields a weak bisimilar CTMC [4] in which all states have identical residence time distributions. Transition probabilities of single transitions are abstracted by intervals, yielding continuous-time variants of interval DTMCs [10,24].

This, however, may yield rather coarse abstractions (see below). This paper suggests to overcome this inaccuracy. The crux of our approach is to collapse *transition sequences* of a given fixed length k, say. Our technique in [17] is obtained if k=1. This paper presents the theory of this abstraction technique, shows its correctness, and shows its application by a quantitative analysis of an enzyme-catalyzed substrate conversion, a well-known case study from biochemistry [5].

Let us illustrate the main idea of the abstraction by means of an example. Consider the CTMC shown on the right (top). Intuitively, a CTMC can be considered as a transition system whose transitions are labeled with *transition probabilities*. Moreover, a CTMC comes with an *exit rate* identifying the *residence times* of the states (one, say), which is exponentially distributed. The essence of CTMC model checking is to compute the probability to reach a certain set of goal states within a given deadline [3].

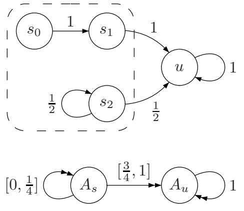

Fig. 1.

A rather common approach to abstraction is to partition the state space into classes, e.g., let us group states s_0, s_1, and s_2 into the abstract state A_s, and u into A_u. The probability to move from A_s to A_u by a single transition is either 0, $\frac{1}{2}$, or 1, as the respective (time-abstract) probability to move to u *in one transition* is 0, 1, and $\frac{1}{2}$. The approach in [17] yields the interval $[0, 1]$ for the transition from A_s to A_u. This is not very specific. A more narrow interval is obtained when considering two consecutive transitions. Then, the probability from A_s to A_u is 1 or $\frac{3}{4}$. Using intervals, this yields the two-state abstract structure depicted above (bottom).

Put in a nutshell, the abstraction technique proposed in this paper is to generalize this approach towards considering transition sequences of a given length $k > 0$, say. State residence times are, however, then no longer exponentially distributed, but Erlang-k distributed. Moreover, taking each time k steps at once complicates the exact calculation of time-bounded reachability probabilities: Let us consider first the case that n is the number of transitions taken in the concrete system to reach a certain goal state. Let ℓ and j be such that $n = \ell{\cdot}k{+}j$ and $j \in \{0, \ldots, k{-}1\}$. Clearly, the number of transitions in the abstract system corresponds exactly to a multiple of the number of transitions in the concrete system, only if the remainder j equals 0. As this is generally not the case,

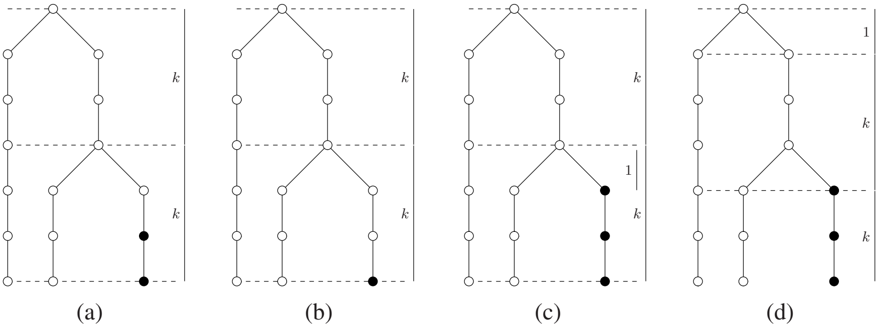

Fig. 2. Reaching goals in stages of length k

we restrict to computing lower and upper bounds for the probability of reaching a set of goal states. Let us be more precise: Consider the tree of state sequences as shown in Fig. 2(a). Let the black nodes denote the set of goal states. Taking the right branch, 5 transitions are needed to reach a goal state. For $k = 3$, this implies that 2 abstract transitions lead to a goal state. However, as $2 \cdot 3 = 6$, computing with 2 transitions and Erlang-3 distributed residence times will not give the exact probability for reaching a goal state, but, as we show, a *lower bound*. Intuitively, the probability for reaching a goal state in Fig. 2(b) is computed. For an upper bound, one might first consider all states from the fourth state on in the right branch as goal states. This would give a rather coarse upper bound. We follow instead the idea depicted in Fig. 2(c): We consider 2 transitions for reaching the goal state, however, use the Erlang-3 distribution for assessing the first transition, but use the Erlang-1 distribution for assessing the last transition of a sequence of transitions. That is, we compute the reachability probability for the goal states as depicted in Fig. 2(c). Technically, it is beneficial to understand the situation as depicted in Fig. 2(d), i.e., to first consider one transition with Erlang-1 distribution and then to consider a sequence of transitions which are Erlang-k distributed.

Outline of the paper. Section 2 gives some necessary background. We introduce Erlang-k interval processes in Section 3 which serve as abstract model for CTMCs in Section 4. In Section 5, we focus on reachability analysis of Erlang-k interval processes and utilize it for model checking in Section 6. The feasibility of our approach is demonstrated in Section 7 by a case study from biology and Section 8 concludes the paper. A full version with detailed proofs can be found in [18].

2 Preliminaries

Let X be a finite set. For $Y, Y' \subseteq X$ and function $f : X \times X \to \mathbb{R}$ let $f(Y, Y') := \sum_{y \in Y, y' \in Y'} f(y, y')$ (for singleton sets, brackets may be omitted). The function $f(x, \cdot)$ is given by $x' \mapsto f(x, x')$ for all $x \in X$. Function f is a *distribution on* X iff $f : X \to [0, 1]$ and $f(X) := \sum_{x \in X} f(x) = 1$. The set of all distributions on X is denoted by *distr*(X). Let *AP* be a fixed, finite set of atomic propositions and $\mathbb{B}_2 := \{\bot, \top\}$ the two-valued truth domain.

Continuous-time Markov chains. A (uniform) *CTMC* $\mathcal{C}$ is a tuple $(S, \mathbf{P}, \lambda, L, s_0)$ with a finite non-empty set of states S, a transition probability function $\mathbf{P} : S \times S \to [0, 1]$ such that $\mathbf{P}(s, S) = 1$ for all $s \in S$, an exit rate $\lambda \in \mathbb{R}_{>0}$, a labeling function $L : S \times AP \to \mathbb{B}_2$, and an initial state $s_0 \in S$. This definition deviates from the literature as i) we assume a uniform exit rate and ii) we separate the discrete-time behavior specified by $\mathbf{P}$ and the residence times determined by λ. Restriction i) is harmless, as every (non-uniform) CTMC can be transformed into a weak bisimilar, uniform CTMC by adding self-loops [25]. For ii), note that $\mathbf{P}(s, s')(1-e^{\lambda t})$ equals the probability to reach s' from s in one step and within time interval $[0, t)$. Thus, the above formulation is equivalent to the standard one. The expected state residence time is $1/\lambda$. Let $\mathbf{P}^k(s, s')$ denote the time-abstract probability to enter state s' after k steps while starting from s, which is obtained by taking the kth-power of $\mathbf{P}$ (understood as a transition probability matrix).

We recall some standard definitions for Markov chains [11,23]. An infinite *path* σ is a sequence $s_0\, t_0\, s_1\, t_1\, \ldots$ with $s_i \in S$, $\mathbf{P}(s_i, s_{i+1}) > 0$ and $t_i \in \mathbb{R}_{>0}$ for $i \in \mathbb{N}$. The time stamps t_i denote the residence time in state s_i. Let $\sigma@t$ denote the state of a path σ occupied at time t, i.e. $\sigma@t = s_i$ with i the smallest index such that $t < \sum_{j=0}^{i} t_j$. The set of all (infinite) paths in $\mathcal{C}$ is denoted by $Path_{\mathcal{C}}$. Let Pr be the probability measure on sets of paths that results from the standard cylinder set construction.

Poisson processes. Let $(N_t)_{t\geq 0}$ be a counting process and let the corresponding interarrival times be independent and identically exponentially distributed with parameter $\lambda > 0$. Then $(N_t)_{t\geq 0}$ is a *Poisson process* and the number k of arrivals in time interval $[0, t)$ is Poisson distributed, i.e., $P(N_t = k) = e^{-\lambda t}(\lambda t)^k/k!$. The time until k arrivals have occurred is Erlang-k distributed, i.e., $F_{\lambda,k}(t) := P(T_k \leq t) = 1 - \sum_{i=0}^{k-1} e^{-\lambda t}\frac{(\lambda t)^i}{i!}$ where T_k is the time instant of the k-th arrival in $(N_t)_{t\geq 0}$. Consequently, the probability that $(N_t)_{t\geq 0}$ is in the range $\{k, k+1, \ldots, k+\ell-1\}$, $\ell \geq 1$ is given by

$$\psi_{\lambda,t}(k, \ell) := P(T_k \leq t < T_{k+\ell}) = \sum_{i=k}^{k+\ell-1} e^{-\lambda t}\frac{(\lambda t)^i}{i!} \ .$$

A CTMC $\mathcal{C} = (S, \mathbf{P}, \lambda, L, s_0)$ can be represented as a discrete-time Markov chain with transition probabilities $\mathbf{P}$ where the times are implicitly driven by a Poisson process with parameter λ, i.e., the probability to reach state s' from s within $[0, t)$ is:

$$\sum_{i=0}^{\infty} \mathbf{P}^i(s, s') \cdot e^{-\lambda t}\frac{(\lambda t)^i}{i!} \ .$$

This relationship can be used for an efficient transient analysis of CTMCs and is known as *uniformization*. A truncation point of the infinite sum can be calculated such that the approximation error is less than an a priori defined error bound [25].

Continuous Stochastic Logic. CSL [1,3] extends PCTL [12] by equipping the until-operator with a time bound. Its syntax is given by:

$$\varphi ::= \mathit{true} \mid a \mid \varphi \wedge \varphi \mid \neg\varphi \mid \mathcal{P}_{\bowtie p}(\varphi\, \mathcal{U}^I \varphi)$$

where $I \in \{[0, t), [0, t], [0, \infty) \mid t \in \mathbb{R}_{>0}\}$, $\bowtie\, \in \{<, \leq, \geq, >\}$, $p \in [0, 1]$ and $a \in AP$. The formal semantics of CSL is given in Table 1. CSL model checking [3] is performed inductively on the structure of φ like for CTL model checking. Checking time-bounded until-formulas boils down to computing time-bounded reachability probabilities. These

Table 1. Semantics of CSL

$\llbracket \mathit{true} \rrbracket(s) \quad = \top$	$\llbracket a \rrbracket(s) \quad = L(s,a)$
$\llbracket \varphi_1 \wedge \varphi_2 \rrbracket(s) = \llbracket \varphi_1 \rrbracket(s) \sqcap \llbracket \varphi_2 \rrbracket(s)$	$\llbracket \neg\varphi \rrbracket(s) = (\llbracket \varphi \rrbracket(s))^c$
$\llbracket \mathcal{P}_{\bowtie p}(\varphi_1 \mathcal{U}^I \varphi_2) \rrbracket(s) = \top$, iff $Pr(\{\sigma \in \mathit{Path}_s^{\mathcal{M}} \mid \llbracket \varphi_1 \mathcal{U}^I \varphi_2 \rrbracket(\sigma) = \top\}) \bowtie p$	
$\llbracket \varphi_1 \mathcal{U}^I \varphi_2 \rrbracket(\sigma) = \top$, iff $\exists\, t \in I : (\llbracket \varphi_2 \rrbracket(\sigma @ t) = \top \wedge \forall\, t' \in [0,t) : \llbracket \varphi_1 \rrbracket(\sigma @ t') = \top)$	

probabilities can be obtained by a reduction to transient analysis yielding a time complexity in $\mathcal{O}(|S|^2\lambda t)$ where t is the time bound.

Three-valued domain. Let $\mathbb{B}_3 := \{\bot, ?, \top\}$ be the complete lattice with ordering $\bot < ? < \top$, meet ($\sqcap$) and join ($\sqcup$) as expected, and complementation $\cdot^c$ such that $\top$ and $\bot$ are complementary to each other and $?^c = ?$. When a formula evaluates to $\bot$ or $\top$, the result is *definitely* true or false respectively, otherwise it is *indefinite*.

3 Erlang-*k* Interval Processes

Erlang-k interval processes are determined by two ingredients: a discrete probabilistic process with intervals of transition probabilities (like in [10,24]) and a Poisson process. The former process determines the probabilistic branching whereas residence times are governed by the latter. More precisely, the state residence time is the time until j further arrivals occur according to the Poisson process where $j \in \{1, \ldots, k\}$ is nondeterministically chosen. Thus, the residence times are Erlang-j distributed.

Definition 1 (Erlang-k interval process). *An* Erlang-k interval process *is a tuple $\mathcal{E} = (S, \mathbf{P}_l, \mathbf{P}_u, \lambda, k, L, s_0)$, with S and $s_0 \in S$ as before, and $\mathbf{P}_l, \mathbf{P}_u : S \times S \to [0,1]$, transition probability bounds such that for all $s \in S$: $\mathbf{P}_l(s,S) \leq 1 \leq \mathbf{P}_u(s,S)$, $\lambda \in \mathbb{R}_{>0}$, a parameter of the associated Poisson process, $k \in \mathbb{N}^+$, and $L : S \times AP \to \mathbb{B}_3$.*

An Erlang-1 interval process is an *abstract continuous-time Markov chain* (ACTMC) [17]. If additionally all intervals are singletons, the process is equivalent to a CTMC with $\mathbf{P}_l = \mathbf{P}_u = \mathbf{P}$. The set of transition probability functions for $\mathcal{E}$ is:

$$\begin{aligned}\mathbf{T}_{\mathcal{E}} := \{\mathbf{P} : S \times S \to [0,1] \mid \forall s \in S :\ & \mathbf{P}(s,S) = 1, \\ & \forall s, s' \in S :\ \mathbf{P}_l(s,s') \leq \mathbf{P}(s,s') \leq \mathbf{P}_u(s,s')\}\end{aligned}$$

Let $\mathbf{T}_{\mathcal{E}}(s) := \{\mathbf{P}(s,\cdot) \mid \mathbf{P} \in \mathbf{T}_{\mathcal{E}}\}$ be the set of distributions in s.

Paths in Erlang-k interval processes. A *path* σ in $\mathcal{E}$ is an infinite sequence $s_0 t_0 s_1 t_1 \ldots$ with $s_i \in S$, $t_i \in \mathbb{R}_{>0}$ for which there exists $\mathbf{P}_0, \mathbf{P}_1, \ldots \in \mathbf{T}_{\mathcal{E}}$ such that $\mathbf{P}_i(s_i, s_{i+1}) > 0$ for all $i \in \mathbb{N}$. A *path fragment* ξ is a prefix of a path that ends in a state denoted $\xi\!\downarrow$. The set of all path fragments ξ (untimed path fragments) in $\mathcal{E}$ is denoted by $\mathit{Pathf}_{\mathcal{E}}$ ($\mathit{uPathf}_{\mathcal{E}}$, respectively) whereas the set of paths is denoted by $\mathit{Path}_{\mathcal{E}}$.

We depict Erlang-k interval processes by drawing the state-transition graph of the discrete part, i.e., the associated interval DTMC with transitions labeled by $[\mathbf{P}_l(s,s'),$

$\mathbf{P}_u(s, s')]$ (see, e.g., Fig. 3). The Poisson process that determines the residence times, as well as the marking of the initial state are omitted.

Normalization. Erlang-k interval process $\mathcal{E}$ is called *delimited*, if every possible selection of a transition probability in a state can be extended to a distribution [17], i.e., if for any $s, s' \in S$ and $p \in [\mathbf{P}_l(s, s'), \mathbf{P}_u(s, s')]$, we have $\mu(s') = p$ for some $\mu \in \mathbf{T}_{\mathcal{E}}(s)$. An Erlang-$k$ interval process $\mathcal{E}$ can be normalized into a delimited one $\mathit{norm}(\mathcal{E})$ such that $\mathbf{T}_{\mathit{norm}(\mathcal{E})} = \mathbf{T}_{\mathcal{E}}$. Formally, $\mathit{norm}(\mathcal{E}) = (S, \tilde{\mathbf{P}}_l, \tilde{\mathbf{P}}_u, \lambda, k, L, s_0)$ with for all $s, s' \in S$:

$$\begin{aligned} \tilde{\mathbf{P}}_l(s, s') &= \max\{\mathbf{P}_l(s, s'), 1 - \mathbf{P}_u(s, S \setminus \{s'\})\} \quad \text{and} \\ \tilde{\mathbf{P}}_u(s, s') &= \min\{\mathbf{P}_u(s, s'), 1 - \mathbf{P}_l(s, S \setminus \{s'\})\}. \end{aligned}$$

Example 1. The Erlang-k interval process in Fig. 3, left, is delimited. Selecting $\frac{1}{4}$ for the transition from s to u_2 yields a non-delimited process (Fig. 3, middle). Applying normalization results in the Erlang-k interval process shown in Fig. 3, right.

An Erlang-k interval process contains two sources of nondeterminism: in each state, (i) a distribution according to the transition probability intervals, and (ii) the number $j \in \{1, \ldots, k\}$ of arrivals in the Poisson process may be chosen. As usual, nondeterminism is resolved by a scheduler:

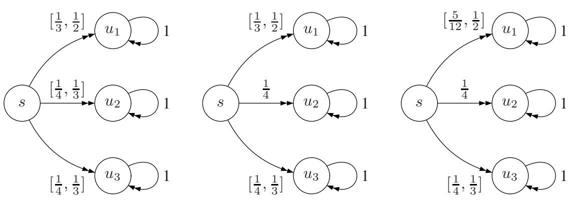
$[\frac{1}{3}, \frac{1}{2}]$
u_1
1
s
$[\frac{1}{4}, \frac{1}{3}]$
u_2
1
$[\frac{1}{4}, \frac{1}{3}]$
u_3
1
$[\frac{1}{3}, \frac{1}{2}]$
u_1
1
s
$\frac{1}{4}$
u_2
1
$[\frac{1}{4}, \frac{1}{3}]$
u_3
1
$[\frac{5}{12}, \frac{1}{2}]$
u_1
1
s
$\frac{1}{4}$
u_2
1
$[\frac{1}{4}, \frac{1}{3}]$
u_3
1

Fig. 3. Normalization

Definition 2 (Scheduler). *Let $\mathcal{E}$ be an Erlang-k interval process. A* history-dependent deterministic scheduler *is a function $D : \mathit{uPathf}_{\mathcal{E}} \to \mathit{distr}(S) \times \{1, \ldots, k\}$ such that $D(\xi) \in \mathbf{T}_{\mathcal{E}}(\xi\downarrow) \times \{1, \ldots, k\}$ for all $\xi \in \mathit{uPathf}_{\mathcal{E}}$. The set of all history-dependent deterministic schedulers of $\mathcal{E}$ is denoted as $\mathcal{HD}^{\mathcal{E}}$.*

Note that a richer class of schedulers is obtained if the scheduler's choice may also depend on the residence times of the states visited so far. We show below that the class of history-dependent deterministic schedulers suffices when Erlang-k interval processes are used for abstracting CTMCs.

Probability measure. For Erlang-k interval process $\mathcal{E}$, let $\Omega = \mathit{Path}_{\mathcal{E}}$ be the sample space and $\mathcal{B}$ the Borel field generated by the basic cylinder sets $\mathcal{C}(s_0\, I_0\, \ldots\, I_{n-1}\, s_n)$ where $s_i \in S$, $0 \leq i \leq n$ and $I_\ell = [0, x_\ell) \subseteq \mathbb{R}_{\geq 0}$ is a non-empty interval for $0 \leq \ell < n$. The set $\mathcal{C}(s_0\, I_0\, \ldots\, I_{n-1}\, s_n)$ contains all paths of $\mathcal{E}$ with prefix $\hat{s}_0\, t_0\, \ldots\, t_{n-1}\, \hat{s}_n$ such that $s_i = \hat{s}_i$ and $t_\ell \in I_\ell$. A scheduler $D \in \mathcal{HD}^{\mathcal{E}}$ induces a probability space $(\Omega, \mathcal{B}, \mathit{Pr}^D)$ where Pr^D is uniquely given by $\mathit{Pr}^D(C(s_0)) := 1$ and for $n \geq 0$

$$\begin{aligned} \mathit{Pr}^D(C(s_0\, I_0 \ldots I_n\, s_{n+1})) &:= \mathit{Pr}^D(C(s_0\, I_0 \ldots I_{n-1}\, s_n)) \cdot F_{\lambda, j_n}(\sup I_n) \cdot \mu_n(s_{n+1}) \\ &= \textstyle\prod_{i=0}^{n} \left(F_{\lambda, j_i}(\sup I_i) \cdot \mu_i(s_{i+1})\right) \end{aligned}$$

where $(\mu_i, j_i) =: D(s_0\, s_1\, \ldots\, s_i)$. Additionally, we define the time-abstract probability measure induced by D as $Pr_{ta}^D(C(s_0)) := 1$ and

$$Pr_{ta}^D(C(s_0\, I_0\, \ldots\, I_n\, s_{n+1})) := \prod_{i=0}^n \mu_i(s_{i+1}).$$

We are interested in the supremum/infimum (ranging over all schedulers) of the probability of measurable sets of paths. Clearly, the choice of j_i, the number of steps in the associated Poisson process in state s_i, may influence such quantities. For instance, on increasing j_i, time-bounded reachability probabilities will decrease as the expected state residence time (in s_i) becomes longer. We discuss the nondeterministic choice in the Poisson process in subsequent sections, and now focus on the choice of distribution μ_i according to the probability intervals.

Definition 3 (Extreme distributions). *Let $\mathcal{E}$ be an Erlang-k interval process, $s \in S$ and $S' \subseteq S$. We define extr$(\mathbf{P}_l, \mathbf{P}_u, S', s) \subseteq \mathbf{T}_{\mathcal{E}}(s)$ such that $\mu \in extr(\mathbf{P}_l, \mathbf{P}_u, S', s)$ iff either $S' = \emptyset$ and $\mu = \mathbf{P}_l(s, \cdot) = \mathbf{P}_u(s, \cdot)$ or one of the following conditions holds*[1]*:*

- $\exists s' \in S' : \mu(s') = \mathbf{P}_l(s, s')$ *and* $\mu \in$ *extr*$(\mathbf{P}_l, \mathbf{P}_u[(s, s') \mapsto \mu(s')], S' \setminus \{s'\}, s)$
- $\exists s' \in S' : \mu(s') = \mathbf{P}_u(s, s')$ *and* $\mu \in$ *extr*$(\mathbf{P}_l[(s, s') \mapsto \mu(s')], \mathbf{P}_u, S' \setminus \{s'\}, s)$

We call $\mu \in \mathbf{T}_{\mathcal{E}}(s)$ an extreme distribution *if $\mu \in$ extr$(\mathbf{P}_l, \mathbf{P}_u, S, s)$.*

A scheduler $D \in \mathcal{HD}^{\mathcal{E}}$ is called *extreme* if all choices $D(\xi)$ are extreme distributions. For a subset $\mathcal{D} \subseteq \mathcal{HD}^{\mathcal{E}}$ let $\mathcal{D}_{extr} \subseteq \mathcal{D}$ denote the subset of all extreme schedulers in $\mathcal{D}$.

Theorem 1 (Extrema). *Let $\mathcal{E}$ be an Erlang-k interval process and $\mathcal{D} \subseteq \mathcal{HD}^{\mathcal{E}}$. For every measurable set $Q \in \mathcal{B}$ of the induced probability space:*

$$\inf_{D \in \mathcal{D}_{extr}} Pr^D(Q) = \inf_{D \in \mathcal{D}} Pr^D(Q), \quad \sup_{D \in \mathcal{D}_{extr}} Pr^D(Q) = \sup_{D \in \mathcal{D}} Pr^D(Q).$$

4 Abstraction

This section makes the abstraction by stages as motivated in the introduction precise. We define an abstraction operator based on the idea of partitioning the concrete states to form abstract states. This yields an Erlang-k interval process. Moreover, we introduce a simulation relation relating one transition in the abstract system to a sequence of k transitions in the concrete system. We show that the abstraction operator yields an Erlang-k interval process simulating the original CTMC.

Definition 4 (Abstraction). *Let abstr$(\mathcal{C}, \mathcal{A}, k) := (\mathcal{A}, \mathbf{P}_l, \mathbf{P}_u, \lambda, k, L', A_0)$ be the* abstraction *of CTMC $\mathcal{C} = (S, \mathbf{P}, \lambda, L, s_0)$ induced by partitioning $\mathcal{A} = \{A_0, \ldots, A_n\}$ of S and $k \in \mathbb{N}^+$ such that for all $1 \le i, j \le n$:*

- $\mathbf{P}_l(A_i, A_j) = \min_{s \in A_i} \mathbf{P}^k(s, A_j)$, *and* $\mathbf{P}_u(A_i, A_j) = \max_{s \in A_i} \mathbf{P}^k(s, A_j)$
- $L'(A, a) = \begin{cases} \top & \textit{if for all } s \in A : L(s, a) = \top \\ \bot & \textit{if for all } s \in A : L(s, a) = \bot \\ ? & \textit{otherwise} \end{cases}$
- $A_0 \in \mathcal{A}$ *with* $s_0 \in A_0$.

[1] $f[y \mapsto x]$ denotes the function that agrees everywhere with f except at y where it equals x.

Lemma 1. *For any CTMC $\mathcal{C}$, any partitioning $\mathcal{A}$ of S and $k \in \mathbb{N}^+$, abstr$(\mathcal{C}, \mathcal{A}, k)$ is an Erlang-k interval process.*

Example 2. Reconsider the CTMC $\mathcal{C}$ from Section 1 (Fig. 1), top, with exit rate $\lambda = 1$ and partitioning $\{A_s, A_u\}$ with $A_s = \{s_0, s_1, s_2\}$, $A_u = \{u\}$. As remarked above, in the Erlang-1 interval process *abstr*$(\mathcal{C}, \{A_s, A_u\}, 1)$ (not shown) the probability interval for a transition from A_s to A_u is $[0, 1]$. However, choosing $k = 2$ yields smaller intervals. The resulting Erlang-2 interval process is depicted in Fig. 1, bottom. The plot in Fig. 4 shows the probability to reach $A_u = \{u\}$ within t time units if the Erlang-2 interval process starts at time 0 in A_s and the CTMC in s_0, s_1 or s_2, respectively. For the Erlang-2 interval process, the infimum over all schedulers is taken and it is obviously smaller than all the concrete probabilities in the CTMC (the supremum coincides with the probabilities for s_1). A detailed discussion on which *schedulers* yield the infimum or supremum is given in the next section.

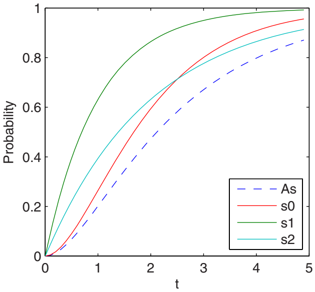

Fig. 4. Concrete vs. abstract behavior over time

Definition 5 **(k-step forward simulation).** *Let $\mathcal{C} = (S_\mathcal{C}, \mathbf{P}, \lambda, L_\mathcal{C}, s_\mathcal{C})$ be a CTMC and $\mathcal{E} = (S_\mathcal{E}, \mathbf{P}_l, \mathbf{P}_u, \lambda, k, L_\mathcal{E}, s_\mathcal{E})$ an Erlang-k interval process. Relation $\mathcal{R}_k \subseteq S_\mathcal{C} \times S_\mathcal{E}$ is a k-*step forward simulation *on $\mathcal{C}$ and $\mathcal{E}$ iff for all $s \in S_\mathcal{C}$, $s' \in S_\mathcal{E}$, $s\mathcal{R}_k s'$ implies:*

1. *Let $\mu := \mathbf{P}^k(s, \cdot)$. Then there exists $\mu' \in \mathbf{T}_\mathcal{E}(s')$ and $\Delta : S_\mathcal{C} \times S_\mathcal{E} \to [0, 1]$ s.t.*
   (a) $\Delta(u, v) > 0 \Rightarrow u\mathcal{R}_k v$, (b) $\Delta(u, S_\mathcal{E}) = \mu(u)$, (c) $\Delta(S_\mathcal{C}, v) = \mu'(v)$.
2. *For all $a \in AP$, $L_\mathcal{E}(s', a) \neq ?$ implies that $L_\mathcal{E}(s', a) = L_\mathcal{C}(s, a)$.*

We write $s \preceq_k s'$ if $s\mathcal{R}_k s'$ for some k-step forward simulation $\mathcal{R}_k$, and $\mathcal{C} \preceq_k \mathcal{E}$ if $s_\mathcal{C}\mathcal{R}_k s_\mathcal{E}$. In the sequel, we often omit subscript k. The main difference with existing simulation relations is that k steps in $\mathcal{C}$ are matched with a single step in $\mathcal{E}$. For k=1, our definition coincides with the standard notion of forward simulation on CTMCs [4].

Theorem 2 **(Abstraction).** *Let $\mathcal{C}$ be a CTMC and let $\mathcal{A}$ be a partitioning on the state space S. Then for all $k \in \mathbb{N}^+$ we have $\mathcal{C} \preceq$ abstr$(\mathcal{C}, \mathcal{A}, k)$.*

It is important to understand that the k-step forward simulation relates the transition probabilities of one transition in the abstract system to k-transitions in the concrete system. However, it does not say anything about the number $j \in \{1, \ldots, k\}$ of arrivals in the Poisson process, which has to be chosen appropriately to guarantee that the probability for reaching certain states within a given time bound is related in the concrete and the abstract system. This issue will be approached in the next section.

5 Reachability

We now show that the abstraction method proposed above can be used to efficiently derive bounds for the probability to reach a set $B \subseteq S_\mathcal{C}$ in a CTMC $\mathcal{C} = (S_\mathcal{C}, \mathbf{P}, \lambda, L_\mathcal{C}, s_\mathcal{C})$.

For that we consider an Erlang-k interval process $\mathcal{E}$ with state space $S_{\mathcal{E}}$ and $\mathcal{C} \preceq \mathcal{E}$. For $B' \subseteq S_{\mathcal{E}}, t \geq 0$ let $\mathit{Reach}_{\leq t}(B') := \{\sigma \in \mathit{Path}_{\mathcal{E}} \mid \exists t' \in [0,t] : \sigma@t' \in B'\}$.

Since a CTMC is also an Erlang-k interval process, $\mathit{Reach}_{\leq t}(B) \subseteq \mathit{Path}_{\mathcal{C}}$ is defined in the same way. We assume that $\mathbf{P}(s,s) = 1$ for all $s \in B$ as the behavior of $\mathcal{C}$ after visiting B can be ignored. We say that B and B' are *compatible* iff $s \preceq s'$ implies that $s \in B$ iff $s' \in B'$, for all $s \in S_{\mathcal{C}}$, $s' \in S_{\mathcal{E}}$. For example, in Fig. 4, $B = \{u\}$ and $B' = \{A_u\}$, as well as, $B = \{s_0, s_1, s_2\}$ and $B' = \{A_s\}$ are compatible.

The k-step forward simulation (cf. Def. 5) is useful for relating transition probabilities in the concrete and the abstract system. However, to relate *timed* reachability probabilities of concrete and abstract systems, we have to assess the time abstract transitions with the *right* number j of new arrivals in the Poisson process associated with $\mathcal{E}$. In other words, we have to check for which choice of the number of arrivals, we obtain lower and upper bounds of the timed reachability probabilities. As motivated in the introduction (Fig. 2) and stated in Theorem 3 (see below), a tight bound for

- the minimum probability is obtained when the scheduler chooses for number j always k, and a tight bound for
- the maximum probability is obtained when the scheduler chooses once $j = 1$ and for the remaining transitions $j = k$.

Consequently, we restrict our attention to the following scheduler classes:

$$\mathcal{HD}^{\mathcal{E}}_l := \{D \in \mathcal{HD}^{\mathcal{E}} \mid \forall \xi \exists \mu_\xi : D(\xi) = (\mu_\xi, k)\}$$
$$\mathcal{HD}^{\mathcal{E}}_u := \{D \in \mathcal{HD}^{\mathcal{E}} \mid \forall \xi \exists \mu_\xi : D(\xi) = (\mu_\xi, 1) \text{ if } \xi = s_{\mathcal{E}}, D(\xi) = (\mu_\xi, k) \text{ otherwise}\}$$

where $s_{\mathcal{E}}$ is the initial state of the Erlang-k interval process $\mathcal{E}$.

Theorem 3. *Let $\mathcal{C}$ be a CTMC and $\mathcal{E}$ an Erlang-k interval process with $\mathcal{C} \preceq \mathcal{E}$. For $t \in \mathbb{R}_{\geq 0}$, compatible sets B and B', there exist schedulers $D \in \mathcal{HD}^{\mathcal{E}}_l$, $D' \in \mathcal{HD}^{\mathcal{E}}_u$ with*

$$\mathit{Pr}^D(\mathit{Reach}_{\leq t}(B')) \leq \mathit{Pr}^{\mathcal{C}}(\mathit{Reach}_{\leq t}(B)) \leq \mathit{Pr}^{D'}(\mathit{Reach}_{\leq t}(B')).$$

Let

$$\mathit{Pr}^{\mathcal{E}}_l(\mathit{Reach}_{\leq t}(B')) := \inf\nolimits_{D \in \mathcal{HD}^{\mathcal{E}}_l} \mathit{Pr}^D(\mathit{Reach}_{\leq t}(B'))$$
$$\mathit{Pr}^{\mathcal{E}}_u(\mathit{Reach}_{\leq t}(B')) := \sup\nolimits_{D \in \mathcal{HD}^{\mathcal{E}}_u} \mathit{Pr}^D(\mathit{Reach}_{\leq t}(B')).$$

The following corollary is a direct result of the theorem above. It states that when comparing reachability probabilities of a CTMC with those of a simulating Erlang-k interval process $\mathcal{E}$, in the *worst (best) case* $\mathcal{E}$ will have a smaller (larger) time-bounded reachability probability, when restricting to the scheduler class $\mathcal{HD}^{\mathcal{E}}_l$ ($\mathcal{HD}^{\mathcal{E}}_u$).

Corollary 1. *Let $\mathcal{C}$ be a CTMC and $\mathcal{E}$ an Erlang-k interval process with $\mathcal{C} \preceq \mathcal{E}$. Let $t \in \mathbb{R}_{\geq 0}$ and B be compatible with B'. Then:*

$$\mathit{Pr}^{\mathcal{E}}_l(\mathit{Reach}_{\leq t}(B')) \leq \mathit{Pr}^{\mathcal{C}}(\mathit{Reach}_{\leq t}(B)) \leq \mathit{Pr}^{\mathcal{E}}_u(\mathit{Reach}_{\leq t}(B'))$$

Similar to the uniformization method for CTMCs (see Section 2), we can efficiently calculate time-bounded reachability probabilities in $\mathcal{E}$, using time-abstract reachability probabilities and the probability for the number of Poisson arrivals in a certain range.

More specifically, after i transitions in $\mathcal{E}$, the number of arrivals in the associated Poisson process is among $i \cdot k, i \cdot k{+}1, \dots, i \cdot k{+}(k{-}1)$, if $D \in \mathcal{HD}^{\mathcal{E}}_{l}$, and $(i{-}1) \cdot k{+}1, (i{-}1) \cdot k{+}2, \dots, i{\cdot}k$, if $D \in \mathcal{HD}^{\mathcal{E}}_{u}$. For $B \subseteq S_{\mathcal{E}}$, $i \in \mathbb{N}$ let $\mathit{Reach}^{=i}(B) := \{\sigma \in \mathit{Path}_{\mathcal{E}} \mid \sigma[i] \in B\}$. Using $\psi_{\lambda,t}$ for the respective Poisson probabilities, we thus obtain:

Lemma 2. *Let $\mathcal{E}$ be an Erlang-k interval process, $t \in \mathbb{R}_{\geq 0}$ and $B \subseteq S_{\mathcal{E}}$. Then*

$$\mathit{Pr}^{D}(\mathit{Reach}_{\leq t}(B)) = \sum_{i=0}^{\infty} \left(\mathit{Pr}^{D}_{\mathit{ta}}(\mathit{Reach}^{=i}(B)) \cdot \psi_{\lambda,t}(\textstyle\sum_{h=0}^{i-1} j_h, j_i)\right)$$

where $j_i = k$ for all $i \in \mathbb{N}$ if $D \in \mathcal{HD}^{\mathcal{E}}_{l}$ and $j_0 = 1$, $j_i = k$ for $i \in \mathbb{N}^{+}$ if $D \in \mathcal{HD}^{\mathcal{E}}_{u}$.

Similar as in [2], we can approximate the supremum/infimum w.r.t. the scheduler classes $\mathcal{HD}^{\mathcal{E}}_{l}$ and $\mathcal{HD}^{\mathcal{E}}_{u}$ by applying a greedy strategy for the optimal choices of distributions $\mathbf{P} \in \mathbf{T}_{\mathcal{E}}$. A truncated, step-dependent scheduler is sufficient to achieve an accuracy of $1 - \epsilon$ where the error bound $\epsilon > 0$ is specified a priori. The decisions of this scheduler only depend on the number of transitions performed so far and its first $N := N(\epsilon)$ decisions can be represented by a sequence $\mathbf{P}_1, \dots, \mathbf{P}_N \in \mathbf{T}_{\mathcal{E}}$. As discussed in Section 3, it suffices if the matrices are such that only extreme distributions are involved. As the principle for the greedy algorithm is similar for suprema and infima, we focus on the former. Let i_B be the vector of size $|S_{\mathcal{E}}|$ with $\mathsf{i}_B(s) = 1$ iff $s \in B$. Furthermore, $\mathbf{P}_0 := \mathbf{I}$ and $\mathsf{v}_i := \prod_{m=0}^{i} \mathbf{P}_m \cdot \mathsf{i}_B$. We choose matrices $\mathbf{P}_i$, $i \geq 1$ such that

$$|\mathit{Pr}^{\mathcal{E}}_{u}(\mathit{Reach}_{\leq t}(B)) - \textstyle\sum_{i=0}^{N} \mathsf{v}_i(s_{\mathcal{E}}) \cdot \psi_{\lambda,t}(\sum_{h=0}^{i-1} j_h, j_i)| < \epsilon.$$

The algorithm is illustrated in Fig. 5 and has polynomial time complexity. Starting in a backward manner, i.e., with $\mathbf{P}_N$, vector q^{u}_{i} is maximized by successively assigning as much proportion as possible to the transition leading to the successor s' for which $q^{u}_{i+1}(s')$ is maximal. For every choice of a value $\mathbf{P}_i(s, s')$ the transition probability intervals for the remaining choices are normalized (compare Example 1). Note that the algorithm computes bounds which may be with an error bound ϵ *below* the actual value. Thus, the computed lower bound may be lower than the actual lower bound. To assure that the upper bound exceeds the actual upper bound, we add ϵ to q^{u}_{0}.

The following lemma is an adaptation of [2, Th. 5] and states that the results are indeed ϵ-approximations of the supremum/infimum of the reachability probabilities.

Input: Erlang-k interval process $\mathcal{E}$, time bound t, set of states B	Input: Erlang-k interval process $\mathcal{E}$, time bound t, set of states B
Output: ϵ-approx. q^{l}_{0} of $\mathit{Pr}^{\mathcal{E}}_{l}(\mathit{Reach}_{\leq t}(B))$	Output: ϵ-approx. q^{u}_{0} of $\mathit{Pr}^{\mathcal{E}}_{u}(\mathit{Reach}_{\leq t}(B))$
Minimize q^{l}_{0} *where for* $1 \leq i \leq N$	*Maximize* q^{u}_{0} *where for* $1 \leq i \leq N$
$q^{l}_{0} = \psi_{\lambda,t}(0, k)\, \mathsf{i}_B + q^{l}_{1}$	$q^{u}_{0} = \psi_{\lambda,t}(0, 1)\, \mathsf{i}_B + q^{u}_{1} + \epsilon$
$q^{l}_{i} = \psi_{\lambda,t}(ik, k)\, \mathbf{P}_i \mathsf{i}_B + \mathbf{P}_i\, q^{l}_{i+1}$	$q^{u}_{i} = \psi_{\lambda,t}(1 + (i{-}1)k, k)\, \mathbf{P}_i \mathsf{i}_B + \mathbf{P}_i\, q^{u}_{i+1}$
$q^{l}_{N+1} = \underline{0}$	$q^{u}_{N+1} = \underline{0}$

Fig. 5. Greedy algorithm for infimum (left) and supremum (right) of time-bounded reachability probabilities

Table 2. Three-valued semantics of CSL

$$[\![\mathit{true}]\!](s) = \top \qquad [\![a]\!](s) = L(s,a)$$

$$[\![\varphi_1 \wedge \varphi_2]\!](s) = [\![\varphi_1]\!](s) \sqcap [\![\varphi_2]\!](s) \qquad [\![\neg\varphi]\!](s) = ([\![\varphi]\!](s))^c$$

$$[\![\varphi_1 \mathcal{U}^I \varphi_2]\!](\sigma) = \begin{cases} \top & \text{if } \exists\, t \in I : ([\![\varphi_2]\!](\sigma@t) = \top \wedge \forall\, t' \in [0,t) : [\![\varphi_1]\!](\sigma@t') = \top) \\ \bot & \text{if } \forall\, t \in I : ([\![\varphi_2]\!](\sigma@t) = \bot \vee \exists\, t' \in [0,t) : [\![\varphi_1]\!](\sigma@t') = \bot) \\ ? & \text{otherwise} \end{cases}$$

$$[\![\mathcal{P}_{\trianglerighteq p}(\varphi_1\, \mathcal{U}^I \varphi_2)]\!](s) = \begin{cases} \top & \text{if } \mathit{Pr}_l(s, \varphi_1\, \mathcal{U}^I \varphi_2) \trianglerighteq p \\ \bot & \text{if } \mathit{Pr}_u(s, \varphi_1\, \mathcal{U}^I \varphi_2) \lhd p \\ ? & \text{otherwise} \end{cases} \qquad \trianglerighteq \in \{>, \geq\}, \lhd = \begin{cases} < \text{ if } \trianglerighteq = \leq \\ \leq \text{ if } \trianglerighteq = < \end{cases}$$

$$[\![\mathcal{P}_{\trianglelefteq p}(\varphi_1\, \mathcal{U}^I \varphi_2)]\!](s) = \begin{cases} \top & \text{if } \mathit{Pr}_u(s, \varphi_1\, \mathcal{U}^I \varphi_2) \trianglelefteq p \\ \bot & \text{if } \mathit{Pr}_l(s, \varphi_1\, \mathcal{U}^I \varphi_2) \rhd p \\ ? & \text{otherwise} \end{cases} \qquad \trianglelefteq \in \{<, \leq\}, \rhd = \begin{cases} > \text{ if } \trianglelefteq = \geq \\ \geq \text{ if } \trianglelefteq = > \end{cases}$$

Lemma 3. *For an Erlang-k interval process $\mathcal{E}$, $B \subseteq S_\mathcal{E}$, $t \geq 0$, error margin $\epsilon > 0$:*

$$\mathit{Pr}_l^\mathcal{E}(\mathit{Reach}_{\leq t}(B)) \geq q_0^l(s_\mathcal{E}) \geq \mathit{Pr}_l^\mathcal{E}(\mathit{Reach}_{\leq t}(B)) - \epsilon$$
$$\mathit{Pr}_u^\mathcal{E}(\mathit{Reach}_{\leq t}(B)) \leq q_0^u(s_\mathcal{E}) \leq \mathit{Pr}_u^\mathcal{E}(\mathit{Reach}_{\leq t}(B)) + \epsilon.$$

We conclude this section with a result that allows us to use the algorithm presented above to check if a reachability probability is at least (at most) p in the abstract model and, in case the result is positive, to deduce that the same holds in the concrete model.

Theorem 4. *For a CTMC $\mathcal{C}$, an Erlang-k interval process $\mathcal{E}$ with $\mathcal{C} \preceq \mathcal{E}$, compatible sets $B \subseteq S_\mathcal{C}$, $B' \subseteq S_\mathcal{E}$, $t \geq 0$, $\epsilon > 0$, the algorithm in Fig. 5 computes q_0^l and q_0^u with:*

$$\mathit{Pr}^\mathcal{C}(\mathit{Reach}_{\leq t}(B)) \geq \mathit{Pr}_l^\mathcal{E}(\mathit{Reach}_{\leq t}(B')) \geq q_0^l(s_\mathcal{E}) \geq \mathit{Pr}_l^\mathcal{E}(\mathit{Reach}_{\leq t}(B')) - \epsilon$$
$$\mathit{Pr}^\mathcal{C}(\mathit{Reach}_{\leq t}(B)) \leq \mathit{Pr}_u^\mathcal{E}(\mathit{Reach}_{\leq t}(B')) \leq q_0^u(s_\mathcal{E}) \leq \mathit{Pr}_u^\mathcal{E}(\mathit{Reach}_{\leq t}(B')) + \epsilon.$$

6 Model Checking

The characterizations in Section 5 in terms of minimal and maximal time-bounded reachability probabilities are now employed for model checking CSL on Erlang-k interval processes. Therefore, we define a three-valued CSL semantics and show that verification results on Erlang-k interval processes carry over to their underlying CTMCs.

Three-valued semantics. For Erlang-k interval process $\mathcal{E} = (S, \mathbf{P}_l, \mathbf{P}_u, \lambda, k, L, s_0)$, we define the satisfaction function $[\![\,\cdot\,]\!] : \text{CSL} \to (S \cup \mathit{Path}_\mathcal{E} \to \mathbb{B}_3)$ as in Table 2, where $s \in S$, $\mathcal{E}_s$ is defined as $\mathcal{E}$ but with initial state s and

$$\mathit{Pr}_l(s, \varphi_1\, \mathcal{U}^I \varphi_2) = \mathit{Pr}_l^{\mathcal{E}_s}(\{\sigma \in \mathit{Path}_{\mathcal{E}_s} \mid [\![\varphi_1\, \mathcal{U}^I \varphi_2]\!](\sigma) = \top\}) \qquad (1)$$
$$\mathit{Pr}_u(s, \varphi_1\, \mathcal{U}^I \varphi_2) = \mathit{Pr}_u^{\mathcal{E}_s}(\{\sigma \in \mathit{Path}_{\mathcal{E}_s} \mid [\![\varphi_1\, \mathcal{U}^I \varphi_2]\!](\sigma) \neq \bot\}) \qquad (2)$$

For the propositional fragment the semantics is clear. A path σ satisfies until formula $\varphi_1\, \mathcal{U}^{[0,t]} \varphi_2$ if φ_1 definitely holds until φ_2 definitely holds at the latest at time t. The until-formula is violated, if either before φ_2 holds, φ_1 is violated, or if φ_2 is definitely violated up to time t. Otherwise, the result is indefinite. To determine the semantics of $\mathcal{P}_{\leq p}(\varphi_1 \, \mathcal{U}^{[0,t]} \varphi_2)$, we consider the probability of the paths for which $\varphi_1 \, \mathcal{U}^{[0,t]} \varphi_2$ is definitely satisfied or perhaps satisfied, i.e., indefinite. If this probability is at most p then $\mathcal{P}_{\leq p}(\varphi_1 \, \mathcal{U}^{[0,t]} \varphi_2)$ is definitely satisfied. Similarly, $\mathcal{P}_{\leq p}(\varphi_1 \, \mathcal{U}^{[0,t]} \varphi_2)$ is definitely violated if this probability exceeds p for those paths on which $\varphi_1 \, \mathcal{U}^{[0,t]} \varphi_2$ evaluates to $\top$. The semantics of $\mathcal{P}_{\trianglelefteq p}(\varphi_1 \, \mathcal{U}^{[0,t]} \varphi_2)$ for $\trianglelefteq \, \in \{<, >, \geq\}$ follows by a similar argumentation.

Theorem 5 (Preservation). *For a CTMC $\mathcal{C}$ and an Erlang-k interval process $\mathcal{E}$ with initial states $s_{\mathcal{C}}$ and $s_{\mathcal{E}}$, if $s_{\mathcal{C}} \preceq s_{\mathcal{E}}$ then for all CSL formulas φ:*

$$[\![\varphi]\!](s_{\mathcal{E}}) \neq ? \textit{ implies } [\![\varphi]\!](s_{\mathcal{E}}) = [\![\varphi]\!](s_{\mathcal{C}}).$$

Model checking three-valued CSL is, as usual, done bottom-up the parse tree of the formula. The main task is checking until-subformulas $\mathcal{P}_{\leq p}(a \, \mathcal{U}^{[0,t]} b)$, which can be handled as follows: As in [7], the underlying transition system is transformed such that there is one sink for all states satisfying b and another one for all states neither satisfying a nor b. Thus, all paths reaching states satisfying b are along paths satisfying a, which allows to compute the measure for reaching b states. However, before doing so, we have to account for indefinite states (?): When computing lower bounds we consider all states labeled by ? as ones labeled $\bot$, while we consider them as labeled $\top$ when computing upper bounds, following equations (1) and (2).

Example 3. Consider Ex. 2 where state u (and thus A_u) are labeled *goal*, and CSL formula $\varphi = \mathcal{P}_{\leq 0.9}(\mathit{true} \, \mathcal{U}^{\leq 1.2} \mathit{goal})$. Then $[\![\varphi]\!](A_s) = \top = [\![\varphi]\!](s_0)$ (compare Fig. 4). If s_1 was labeled *goal* as well then $L(A_s, \mathit{goal}) = ?$. Checking φ for satisfaction requires an optimistic relabeling, i.e. we set $L(A_s, \mathit{goal}) = \top$. Obviously, then φ is not satisfied for sure. Analyzing the pessimistic instance with $L(A_s, \mathit{goal}) = \bot$ however yields that φ is neither violated for sure (cf. Fig. 4). Therefore $[\![\varphi]\!](A_s) = ?$ implying that either the partitioning or the choice of k has to be revised in order to get conclusive results.

Theorem 6 (Complexity). *Given an Erlang-k interval process $\mathcal{E}$, a CSL formula φ, and an error margin ϵ, we can approximate $[\![\varphi]\!]$ in time polynomial in the size of $\mathcal{E}$ and linear in the size of φ, the exit rate λ and the highest time bound t occurring in φ (dependency on ϵ is omitted as ϵ is linear in λt). In case the approximation yields $\top$ or $\bot$, the result is correct.*

7 Case Study: Enzymatic Reaction

Markovian models are well established for the analysis of biochemical reaction networks [5,15]. Typically, such networks are described by a set of reaction types and the involved molecular species, e.g., the different types of molecules. The occurrence of a reaction changes the species' populations as molecules are produced and/or consumed.

Enzyme-catalyzed substrate conversion. We focus on an enzymatic reaction network with four molecular species: enzyme (E), substrate (S), complex (C) and product (P) molecules. The three reaction types R_1, R_2, R_3 are given by the following rules:

$$R_1 : E + S \xrightarrow{c_1} C, \quad R_2 : C \xrightarrow{c_2} E + S, \quad R_3 : C \xrightarrow{c_3} E + P$$

The species on the left hand of the arrow (also called *reactants*) describe how many molecules of a certain type are consumed by the reaction and those on the right hand describe how many are produced. For instance, one molecule of type E and S is consumed by reaction R_1 and one C molecule is produced. The constants $c_1, c_2, c_3 \in \mathbb{R}_{>0}$ determine the probability of the reactions as explained below.

Concrete model. The temporal evolution of the system is represented by a CTMC as follows (cf. [6]): A state corresponds to a population vector $x = (x_E, x_S, x_C, x_P) \in \mathbb{N}^4$ and transitions are triggered by chemical reactions. The change of the current population vector x caused by a reaction of type R_m, $m \in \{1, 2, 3\}$ is expressed as a vector v_m where $v_1 := (-1, -1, 1, 0)$, $v_2 := (1, 1, -1, 0)$ and $v_3 := (1, 0, -1, 1)$. Obviously, reaction R_m is only possible if vector $x + v_m$ contains no negative entries. Given an initial state $s := (s_E, s_S, 0, 0)$, it is easy to verify that the set of reachable states equals $S := \{(x_E, x_S, x_C, x_P) \mid x_E + x_C = s_E, x_S + x_C + x_P = s_S\}$.

The probability that a reaction of type R_m occurs within a certain time interval is determined by the function $\alpha_m : S \to \mathbb{R}_{\geq 0}$. The value $\alpha_m(x)$ is proportional to the number of distinct combinations of R_m's reactants: $\alpha_1(x) := c_1 x_E x_S$, $\alpha_2(x) := c_2 x_C$ and $\alpha_3(x) := c_3 x_C$. We define the transition matrix $\mathbf{P}$ of the CTMC by $\mathbf{P}(x, x + v_m) := \alpha_m(x)/\lambda$ with exit rate $\lambda \geq \max_{x \in S}(\alpha_1(x) + \alpha_2(x) + \alpha_3(x))$. Thus, state x has outgoing transitions $x \xrightarrow{\alpha_m(x)/\lambda} x + v_m$ for all m with $x + v_m \geq \underline{0}$ and the self-loop probability in x is $\mathbf{P}(x, x) := 1 - \big(\alpha_1(x) + \alpha_2(x) + \alpha_3(x)\big)/\lambda$.

We are interested in the probability that within time t the number of type P molecules reaches threshold $n := s_S$, the maximum number of P molecules. We apply labels $AP := \{0, 1, \dots, n\}$ and for $0 \leq a \leq n$ let $L(x, a) := \top$ if $x = (x_E, x_S, x_C, x_P)$ with $x_P = a$ and $L(x, a) := \bot$ otherwise. For the initial populations, we fix $s_E = 20$ and vary s_S between 50 and 2000.

Stiffness. In many biological systems, components act on time scales that differ by several orders of magnitude which leads to *stiff* models. Traditional numerical analysis methods perform poorly in the presence of stiffness because a large number of very small time steps has to be considered. For the enzymatic reaction, stiffness arises if $c_2 \gg c_3$ and results in a high self-loop probability in most states because λ is large compared to $\alpha_1(x) + \alpha_2(x) + \alpha_3(x)$. Thus, even in case of a small number $|S|$ of reachable states, model checking properties like $\mathcal{P}_{\leq 0.9}(\textit{true}\, \mathcal{U}^{[0,t]} n)$ is extremely time consuming. We show how our abstraction method can be used to efficiently verify properties of interest even for stiff parameter sets. We choose a realistic parameter set of $c_1 = c_2 = 1$ and $c_3 = 0.001$. Note that the order of magnitude of the expected time until threshold $n = s_S = 300$ is reached is 10^4 for these parameters.

Abstract model. For the CTMC $\mathcal{C} := (S, \mathbf{P}, \lambda, L, s)$ described above, we choose partitioning $\mathcal{A} := \{A_0, \dots, A_n\}$ with $A_a := \big\{x \in S \mid L(x, a) = \top\big\}$, that is, we group all states in which the number of molecules of type P is the same. Some important remarks are necessary at this point. Abstraction techniques rely on the construction of small abstract models by disregarding details of the concrete model as the latter is too large to be solved efficiently. In this example, we have the additional problem of stiffness and the abstraction method proposed here can tackle this by choosing high values for k. Then one step in the Erlang-k interval process happens after a large number of arrivals in the underlying Poisson process and the self-loop probability in the abstract model is much smaller than in the concrete one. We chose $k \in \{2^{10}, 2^{11}, 2^{12}\}$ for the construction of the Erlang-k interval process *abstr*$(\mathcal{C}, \mathcal{A}, k)$ and calculate the transition probability intervals by taking the k-th matrix power of $\mathbf{P}$. The choice for k is reasonable, since for a given error bound $\epsilon = 10^{-10}$, $s_S = 300$ and $t = 10000$, a transient analysis of the concrete model via uniformization would require around $6 \cdot 10^7$ steps. By contrast, our method considers k steps in the concrete model and around $(6 \cdot 10^7)/k$ steps in the smaller abstract model. Thus, although the construction of the Erlang-k interval process is expensive, the total time savings are enormous. We used the MATLAB software for our prototypical implementation and the calculation of $\mathbf{P}^k$ could be performed efficiently because $\mathbf{P}^{2^j}$ can be computed using j matrix multiplications. As for non-stiff models a smaller value is chosen for k, it is obvious that upper and lower bounds for the k-step transition probabilities can be obtained in a local fashion, i.e. by computing the k-th matrix power of submatrices of $\mathbf{P}$. Therefore, we expect our method to perform well even if $|S|$ is large. However, for stiff *and* large concrete models more sophisticated techniques for the construction of the abstract model must be applied that exploit the fact that only upper and lower bounds are needed.

Experimental results. For $s_S = 200$ we compared the results of our abstraction method for the probability to reach A_n within time bound t with results for the concrete model that were obtained using PRISM. While it took more than one day to generate the plot for the concrete model in Fig. 7, right, our MATLAB implementation took less than one hour for all three pairs of upper and lower probability bounds and different values of t.[2] Our method is accurate as the obtained intervals are small, e.g., for $s_S = 200$, $k = 2^{12}$, $t = 14000$ the relative interval width is

$\|\mathcal{A}\|$	$\|S\|$	time
50	861	$0m\ \ 5s$
300	6111	$37m\ 36s$
500	10311	$70m\ 39s$
1000	20811	$144m\ 49s$
1500	31311	$214m\ \ 2s$
2000	41811	$322m\ 50s$

Fig. 6. Computation times

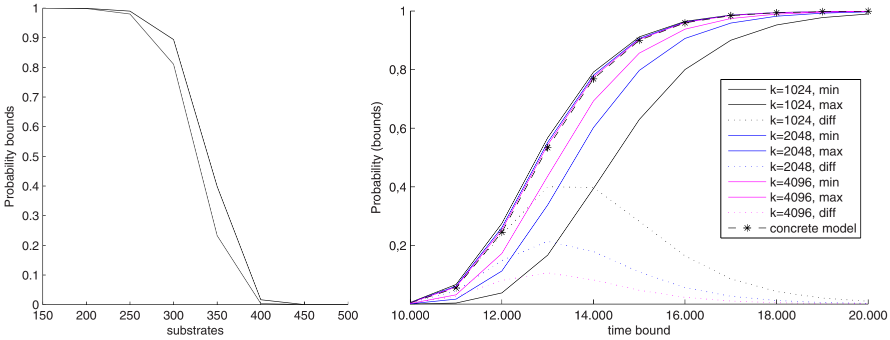

Fig. 7. Time-bounded reachability

[2] Both jobs were run on the same desktop computer (Athlon 64 X2 3800+, 2GB RAM).

10.7%. Fig. 7, left, shows the lower and upper probability bounds using $k = 2^{12}$, $t = 20000$ and varying s_S. For high values of s_S, e.g., $s_S = 500$ the construction of the Erlang-k interval process took more than 99% of the total computation time as the size of the transition matrix $\mathbf{P}$ is $10^4 \times 10^4$ and sparsity is lost during matrix multiplication. We conclude this section with the additional experimental details on computation times[3], given in Fig. 6, using $k = 2^{12}$, $t = 50000$ (and $s_S = 200$).

Note that for this case study exact abstraction techniques such as lumping do not yield any state-space reduction.

8 Conclusion

We have presented an abstraction technique for model checking of CTMCs, presented its theoretical underpinnings, as well as an the application of the abstraction technique to a well-known case study from biochemistry. The main novel aspect of our approach is that besides the abstraction of transition probabilities by intervals [10,17], sequences of transitions may be collapsed yielding an approximation of the timing behavior. Abstract Erlang k-interval processes are shown to provide under- and overapproximations of time-bounded reachability probabilities. Our case study confirms that these bounds may be rather accurate. Future work will focus on automatically finding suitable state-space partitionings, and on guidelines for selecting k appropriately. As shown by our case study, for stiff CTMCs, a high value of k is appropriate. This is, however, not the case in general. We anticipate that graph analysis could be helpful to select a “good” value for k. Moreover, we plan to investigate memory-efficient techniques for computing k-step transition probabilities and counterexample-guided abstraction refinement.

References

1. Aziz, A., Sanwal, K., Singhal, V., Brayton, R.: Model-checking continuous time Markov chains. ACM TOCL 1, 162–170 (2000)
2. Baier, C., Hermanns, H., Katoen, J.P., Haverkort, B.R.: Efficient computation of time-bounded reachability probabilities in uniform continuous-time Markov decision processes. TCS 345, 2–26 (2005)
3. Baier, C., Haverkort, B., Hermanns, H., Katoen, J.-P.: Model-checking algorithms for continuous-time Markov chains. IEEE TSE 29, 524–541 (2003)
4. Baier, C., Katoen, J.-P., Hermanns, H., Wolf, V.: Comparative branching-time semantics for Markov chains. Information and Computation 200, 149–214 (2005)
5. Bower, J.M., Bolouri, H.: Computational Modeling of Genetic and Biochemical Networks. MIT Press, Cambridge (2001)
6. Busch, H., Sandmann, W., Wolf, V.: A numerical aggregation algorithm for the enzyme-catalyzed substrate conversion. In: Priami, C. (ed.) CMSB 2006. LNCS (LNBI), vol. 4210, pp. 298–311. Springer, Heidelberg (2006)
7. Courcoubetis, C., Yannakakis, M.: The complexity of probabilistic verification. Journal of the ACM 42, 857–907 (1995)

[3] Run on a workstation (Xeon 5140 – 2.33 GHz, 32GB RAM).

8. D'Argenio, P.R., Jeannet, B., Jensen, H.E., Larsen, K.G.: Reachability analysis of probabilistic systems by successive refinements. In: de Luca, L., Gilmore, S. (eds.) PROBMIV 2001. LNCS, vol. 2165, pp. 39–56. Springer, Heidelberg (2001)
9. de Alfaro, L., Pritam, R.: Magnifying-lens abstraction for Markov decision processes. In: Damm, W., Hermanns, H. (eds.) CAV 2007. LNCS, vol. 4590, pp. 325–338. Springer, Heidelberg (2007)
10. Fecher, H., Leucker, M., Wolf, V.: Don't know in probabilistic systems. In: Valmari, A. (ed.) SPIN 2006. LNCS, vol. 3925, pp. 71–88. Springer, Heidelberg (2006)
11. Feller, W.: An Introduction to Probability Theory and its Applications, vol. I. John Wiley & Sons, Inc., Chichester (1968)
12. Hansson, H., Jonsson, B.: A logic for reasoning about time and reliability. Formal Aspects of Computing 6, 512–535 (1994)
13. Hermanns, H., Herzog, U., Katoen, J.-P.: Process algebra for performance evaluation. TCS 274, 43–87 (2002)
14. Hermanns, H., Wachter, B., Zhang, L.: Probabilistic CEGAR. In: Gupta, A., Malik, S. (eds.) CAV 2008. LNCS, vol. 5123. Springer, Heidelberg (2008)
15. Kampen, N.G.v.: Stochastic Processes in Physics and Chemistry, 3rd edn. Elsevier, Amsterdam (2007)
16. Katoen, J.-P., Kemna, T., Zapreev, I., Jansen, D.N.: Bisimulation minimisation mostly speeds up probabilistic model checking. In: Grumberg, O., Huth, M. (eds.) TACAS 2007. LNCS, vol. 4424, pp. 87–102. Springer, Heidelberg (2007)
17. Katoen, J.-P., Klink, D., Leucker, M., Wolf, V.: Three-valued abstraction for continuous-time Markov chains. In: Damm, W., Hermanns, H. (eds.) CAV 2007. LNCS, vol. 4590, pp. 316–329. Springer, Heidelberg (2007)
18. Katoen, J.-P., Klink, D., Leucker, M., Wolf, V.: Abstraction for stochastic systems by Erlang's method of stages. Technical Report AIB-2008-12, RWTH Aachen University (2008)
19. Kwiatkowska, M., Norman, G., Parker, D.: Game-based abstraction for Markov decision processes. In: QEST, pp. 157–166. IEEE CS Press, Los Alamitos (2006)
20. Kwiatkowska, M., Norman, G., Parker, D.: Symmetry reduction for probabilistic model checking. In: Ball, T., Jones, R.B. (eds.) CAV 2006. LNCS, vol. 4144, pp. 234–248. Springer, Heidelberg (2006)
21. Mamoun, M.B., Pekergin, N., Younes, S.: Model checking of CTMCs by closed-form bounding distributions. In: QEST, pp. 189–199. IEEE CS Press, Los Alamitos (2006)
22. Remke, A., Haverkort, B., Cloth, L.: CSL model checking algorithms for QBDs. TCS 382, 24–41 (2007)
23. Ross, S.: Stochastic Processes. John Wiley and Sons, Chichester (1996)
24. Sen, K., Viswanathan, M., Agha, G.: Model-checking Markov chains in the presence of uncertainties. In: Hermanns, H., Palsberg, J. (eds.) TACAS 2006. LNCS, vol. 3920, pp. 394–410. Springer, Heidelberg (2006)
25. Stewart, W.: Introduction to the Numerical Solution of Markov Chains. Princeton University Press, Princeton (1995)
26. Wolf, V., Baier, C., Majster-Cederbaum, M.: Trace machines for observing continuous-time Markov chains. ENTCS 153, 259–277 (2004)
27. Younes, H., Simmons, R.: Statistical probabilistic model checking with a focus on time-bounded properties. Inf. and Comp. 204, 1368–1409 (2007)
28. Zhang, L., Hermanns, H., Eisenbrand, F., Jansen, D.N.: Flow faster: efficient decision algorithms for probabilistic simulations. In: Grumberg, O., Huth, M. (eds.) TACAS 2007. LNCS, vol. 4424, pp. 155–170. Springer, Heidelberg (2007)