Verification of Open Interactive Markov Chains

Tomáš Brázdil[1], Holger Hermanns[2], Jan Krčál[1], Jan Křetínský[1,3], and Vojtěch Řehák[1]

1 **Faculty of Informatics, Masaryk University, Czech Republic**
`{brazdil,krcal,jan.kretinsky,rehak}@fi.muni.cz`
2 **Saarland University – Computer Science, Saarbrücken, Germany**
`hermanns@cs.uni-saarland.de`
3 **Institut für Informatik, Technical University Munich, Germany**

Abstract

Interactive Markov chains (IMC) are compositional behavioral models extending both labeled transition systems and continuous-time Markov chains. IMC pair modeling convenience - owed to compositionality properties - with effective verification algorithms and tools - owed to Markov properties. Thus far however, IMC verification did not consider compositionality properties, but considered closed systems. This paper discusses the evaluation of IMC in an open and thus compositional interpretation. For this we embed the IMC into a game that is played with the environment. We devise algorithms that enable us to derive bounds on reachability probabilities that are assured to hold in any composition context.

1998 ACM Subject Classification D.4.8 Performance

Keywords and phrases IMC, compositional verification, synthesis, time bounded reachability, discretization

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2012.474

1 Introduction

With the increasing complexity of systems and software reuse, component based development concepts gain more and more attention. In this setting developers are often facing the need to develop a component with only partial information about the surrounding components at hand, especially when relying on third-party components to be inter-operated with. This motivates verification approaches that ensure the functionality of a component in an environment whose behavior is unknown or only partially known. *Compositional verification* approaches aim at methods to prove guarantees on isolated components in such a way that when put together, the entire system's behavior has the desired properties based on the individual guarantees.

The assurance of reliable functioning of a system relates not only to its correctness, but also to its performance and dependability. This is a major concern especially in embedded system design. A natural instantiation of the general component-based approach in the continuous-time setting are *interactive Markov chains* [24]. Interactive Markov chains (IMC) are equipped with a sound compositional theory. IMC arise from classical labeled transition systems by incorporating the possibility to change state according to a random delay governed by some negative exponential distribution. This twists the model to one that is running in continuous real time. State transitions may be triggered by delay expirations, or may be triggered by the execution of actions. By dropping the new type of transitions, labeled transition systems are regained in their entirety. By dropping action-labeled transitions instead, one arrives at one of the simplest but also most widespread class of performance and de-

© Tomáš Brázdil, Holger Hermanns, Jan Krčál, Jan Křetínský, and Vojtěch Řehák;
licensed under Creative Commons License NC-ND
32nd Int'l Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2012).
Editors: D. D'Souza, J. Radhakrishnan, and K. Telikepalli; pp. 474–485
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

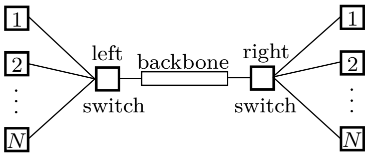
pendability models, *continuous-time Markov chains* (CTMC). IMC have a well-understood compositional theory, rooted in process algebra [3], and are in use as semantic backbones for dynamic fault trees [6], architectural description languages [5, 8], generalized stochastic Petri nets [25] and Statemate [4] extensions, and are applied in a large spectrum of practical applications, ranging from networked hardware on chips [15] to water treatment facilities [21] and ultra-modern satellite designs [16].

In recent years, various analysis techniques have been proposed [18, 27, 23, 26, 34, 19] for IMC. The pivotal verification problem considered is that of *time-bounded reachability*. It is the problem to calculate or approximate the probability that a given state (set) is reached within a given deadline. However, despite the fact that IMC support compositional model generation and minimization very well, the analysis techniques considered thus far are not compositional. They are all bound to the assumption that the analyzed IMC is closed, i.e. does not depend on interaction with the environment. Technically, this is related to the *maximal-progress assumption* governing the interplay of delay and action execution of an IMC component: Internal actions are assumed to happen instantaneously and therefore take precedence over delay transitions while external actions do not. External actions are the process algebraic means for interaction with other components. Remarkably, in all the published IMC verification approaches, all occurring actions are assumed to be internal (respectively internalized by means of a hiding operator prior to analysis).

In this paper, we instead consider *open* IMC, where the control over external actions is in the hands of and possibly delayed by an environment. The environment can be thought of as summarizing the behavior of one or several interacting components. As a consequence, we find ourselves in the setting of a timed game, where the environment has the (timed) control over external actions, while the IMC itself controls choices over internal actions. The resulting game turns out to be remarkably difficult, owed to the interplay of timed moves with external and internal moves of both players.

Concretely, assume we are given an IMC $\mathcal{C}$ which contains some internal non-deterministic transitions and also offers some external actions for synchronization to an unknown environment. Our goal is to synthesize a scheduler controlling the internal transitions which maximizes the probability of reaching a set G of goal states, in time T no matter what and when the environment E decides to synchronize with the external actions. The environment E ranges over all possible IMC able to synchronize with the external actions of $\mathcal{C}$.

To get a principal understanding of the complications faced, we need to consider a restricted setting, where $\mathcal{C}$ does not enable internal and external transitions at the same state. We provide an algorithm which approximates the probability in question up to a given precision $\varepsilon > 0$ and also computes an ε-optimal scheduler. The algorithm consists of two steps. First, we reduce the problem to a game where the environment is not an IMC but can decide to execute external actions at *non-deterministically* chosen time instances. In a second step, we solve the resulting game on $\mathcal{C}$ using discretization. Our discretization is based on the same approach as the algorithm of [34]. However, the algorithm as well as its proof of correctness is considerably more complicated due to presence of non-deterministic choices of the player controlling the environment. We finally discuss what happens if we allow internal and external transitions to be enabled at the same time.



Example. To illustrate the concepts by an example application, we can consider a variant of the *fault-tolerant workstation cluster* [22] depicted on the right. The overall system consists of two sub-clusters connected via a backbone; each of them contains N workstations. Any component can fail and then needs to be repaired to become operational again. There is a single repair unit (not depicted) which must take decisions what to repair next when multiple components are failed. The entire system can be modelled using the IMC composition operators [22], but we are now also in the position to study a partial model, where some components, such as one of the switches, are left unspecified. We seek for the optimal repair schedule regardless of how the unknown components are implemented. We can answer questions such as: *"What is the worst case probability to hit a state in which premium service is not guaranteed within T time units?"* with premium service only being guaranteed if there are at least N operational workstations connected to each other via operational switches.

Our contribution. We investigate the problem of compositionally verifying open IMC. In particular, we introduce the problem of synthesizing optimal control for time-bounded reachability in an IMC interacting in an unknown environment, provided no state enables internal and external transition. Thereafter, we solve the problem of finding ε-optimal schedulers using the established method of discretization, give bounds on the size of the game to be solved for a given ε and thus establish upper complexity bound for the problem. Complete proofs and further relevant details can be found in the full version [10].

Related work. Model checking of *open* systems has been proposed in [28]. The synthesis problem is often stated as a *game* where the first player controls a component and the second player simulates an environment [31]. There is a large body of literature on games in verification, including recent surveys [1, 13]. *Stochastic* games have been applied to e.g. concurrent program synthesis [33] and for collaboration strategies among compositional stochastic systems [14]. Although most papers deal with discrete time games, lately games with stochastic *continuous-time* have gained attention [7, 30, 9, 11]. Some of the games we consider in the present paper exploit special cases of the games considered in [7, 11]. However, both papers prove decidability only for qualitative reachability problems and do not discuss compositionality issues. Further, while systems of [30, 9] are very similar to ours, the structure of the environment is fixed there and the verification is thus not compositional. The same holds for [32, 20], where time is under the control of the components.

The *time-bounded reachability* problem for closed IMC has been studied in [23, 34] and compositional abstraction techniques to compute it are developed in [26]. In the closed interpretation, IMC have some similarities with continuous-time Markov decision processes, CTMDP. Algorithms for time-bounded reachability in CTMDP and corresponding games are developed in [2, 9, 30]. A numerically stable algorithm for time-bounded properties for CTMDP is developed in [12].

2 Interactive Markov Chains

In this section, we introduce the formalism of interactive Markov chains together with the standard way to compose them. After giving the operational interpretation for closed systems, we define the fundamental problem of our interest, namely we define the value of time-bounded reachability and introduce the studied problems.

We denote by $\mathbb{N}$, $\mathbb{N}_0$, $\mathbb{R}_{>0}$, and $\mathbb{R}_{\geq 0}$ the sets of natural numbers, natural numbers with zero, positive real numbers and non-negative real numbers, respectively.

▶ **Definition 1** (IMC). An interactive Markov chain (IMC) is a tuple $\mathcal{C} = (S, \mathbb{A}\mathrm{ct}^\tau, \hookrightarrow, \rightsquigarrow, s_0)$ where S is a finite set of *states*, $\mathbb{A}\mathrm{ct}^\tau$ is a finite set of *actions* containing a designated *internal action* τ, $s_0 \in S$ is an *initial* state,

- $\hookrightarrow \subseteq S \times \mathbb{A}\mathrm{ct}^\tau \times S$ is an *interactive transition* relation, and
- $\rightsquigarrow \subseteq S \times \mathbb{R}_{>0} \times S$ is a *Markovian transition* relation.

Elements of $\mathbb{A}\mathrm{ct} := \mathbb{A}\mathrm{ct}^\tau \setminus \{\tau\}$ are called *external actions*. We write $s \overset{a}{\hookrightarrow} t$ whenever $(s, a, t) \in \hookrightarrow$, and further $\mathrm{succ}_e(s) = \{t \in S \mid \exists a \in \mathbb{A}\mathrm{ct} : s \overset{a}{\hookrightarrow} t\}$ and $\mathrm{succ}_\tau(s) = \{t \in S \mid s \overset{\tau}{\hookrightarrow} t\}$. Similarly, we write $s \overset{\lambda}{\rightsquigarrow} t$ whenever $(s, \lambda, t) \in \rightsquigarrow$ where λ is called a *rate* of the transition, and $\mathrm{succ}_M(s) = \{t \in S \mid \exists \lambda : s \overset{\lambda}{\rightsquigarrow} t\}$. We assume w.l.o.g. that for each pair of states s and t, there is at most one Markovian transition from s to t. We say that an external, or internal, or Markovian transition is available in s if $\mathrm{succ}_e(s) \neq \emptyset$, or $\mathrm{succ}_\tau(s) \neq \emptyset$, or $\mathrm{succ}_M(s) \neq \emptyset$, respectively.

We also define a *total exit rate* function $\mathbf{E} : S \to \mathbb{R}_{\geq 0}$ which assigns to each state the sum of rates of all outgoing Markovian transitions, i.e. $\mathbf{E}(s) = \sum_{s \overset{\lambda}{\rightsquigarrow} t} \lambda$ where the sum is zero if $\mathrm{succ}_M(s)$ is empty. Furthermore, we define a probability matrix $\mathbf{P}(s, t) = \lambda / \mathbf{E}(s)$ if $s \overset{\lambda}{\rightsquigarrow} t$; and $\mathbf{P}(s, t) = 0$, otherwise.

IMC are well suited for compositional modeling, where systems are built out of smaller ones using composition operators. Parallel composition and hiding operators are central to the modeling style, where parallel components synchronize using shared action, and further synchronization can be prohibited by hiding (i.e. internalizing) some actions. IMC employ the *maximal progress assumption*: Internal actions take precedence over the advance of time [24].

▶ **Definition 2** (Parallel composition). For IMC $\mathcal{C}_1 = (S_1, \mathbb{A}\mathrm{ct}_1^\tau, \hookrightarrow_1, \rightsquigarrow_1, s_{01})$ and $\mathcal{C}_2 = (S_2, \mathbb{A}\mathrm{ct}_2^\tau, \hookrightarrow_2, \rightsquigarrow_2, s_{02})$ and a *synchronization alphabet* $A \subseteq \mathbb{A}\mathrm{ct}_1 \cap \mathbb{A}\mathrm{ct}_2$, the parallel composition $\mathcal{C}_1 \parallel_A \mathcal{C}_2$ is the IMC $\mathcal{C}_1 = (S_1 \times S_2, \mathbb{A}\mathrm{ct}_1^\tau \cup \mathbb{A}\mathrm{ct}_2^\tau, \hookrightarrow, \rightsquigarrow, (s_{01}, s_{02}))$ where $\hookrightarrow$ and $\rightsquigarrow$ are defined as the smallest relations satisfying

- $s_1 \overset{a}{\hookrightarrow} s_1'$ and $s_2 \overset{a}{\hookrightarrow} s_2'$ and $a \in A$ implies $(s_1, s_2) \overset{a}{\hookrightarrow} (s_1', s_2')$,
- $s_1 \overset{a}{\hookrightarrow} s_1'$ and $a \notin A$ implies $(s_1, s_2) \overset{a}{\hookrightarrow} (s_1', s_2)$ for each $s_2 \in S_2$,
- $s_2 \overset{a}{\hookrightarrow} s_2'$ and $a \notin A$ implies $(s_1, s_2) \overset{a}{\hookrightarrow} (s_1, s_2')$ for each $s_1 \in S_1$,
- $s_1 \overset{\lambda}{\rightsquigarrow} s_1'$ implies $(s_1, s_2) \overset{\lambda}{\rightsquigarrow} (s_1', s_2)$ for each $s_2 \in S_2$, and
- $s_2 \overset{\lambda}{\rightsquigarrow} s_2'$ implies $(s_1, s_2) \overset{\lambda}{\rightsquigarrow} (s_1, s_2')$ for each $s_1 \in S_1$.

▶ **Definition 3** (Hiding). For an IMC $\mathcal{C} = (S, \mathbb{A}\mathrm{ct}^\tau, \hookrightarrow, \rightsquigarrow, s_0)$ and a *hidden alphabet* $A \subseteq \mathbb{A}\mathrm{ct}$, the hiding $\mathcal{C} \setminus A$ is the IMC $(S, \mathbb{A}\mathrm{ct}^\tau \setminus A, \hookrightarrow', \rightsquigarrow, s_0)$ where $\hookrightarrow'$ is the smallest relation satisfying for each $s \overset{a}{\hookrightarrow} s'$ that $a \in A$ implies $s \overset{\tau}{\hookrightarrow}{}' s'$, and $a \notin A$ implies $s \overset{a}{\hookrightarrow}{}' s'$.

The analysis of IMC has thus far been restricted to *closed* IMC [18, 27, 23, 26, 34, 19]. In a closed IMC, external actions do not appear as transition labels (i.e. $\hookrightarrow \subseteq S \times \{\tau\} \times S$). In practice, this is achieved by an outermost hiding operator $\setminus \mathbb{A}\mathrm{ct}$ closing the composed system. Non-determinism among internal τ transitions is resolved using a (history-dependent) scheduler σ [34].

Let us fix a *closed* IMC $\mathcal{C} = (S, \mathbb{A}\mathrm{ct}^\tau, \hookrightarrow, \rightsquigarrow, s_0)$. The IMC $\mathcal{C}$ under a scheduler σ moves from state to state, and in every state may wait for a random time. This produces a *run* which is an infinite sequence of the form $s_0\, t_0\, s_1\, t_1 \cdots$ where s_n is the n-th visited state and t_n is the time spent there. After n steps, the scheduler resolves the non-determinism based on the *history* $\mathfrak{h} = s_0\, t_0 \cdots s_{n-1}\, t_{n-1}\, s_n$ as follows.

▶ **Definition 4** (Scheduler). A scheduler[1] for an IMC $\mathcal{C} = (S, \mathbb{A}\mathrm{ct}^\tau, \hookrightarrow, \rightsquigarrow, s_0)$ is a measurable[2] function $\sigma : (S \times \mathbb{R}_{\geq 0})^* \times S \to S$ such that for each history $\mathfrak{h} = s_0\, t_0\, s_1 \cdots s_n$ with $\mathrm{succ}_\tau(s_n) \neq \emptyset$ we have $\sigma(\mathfrak{h}) \in \mathrm{succ}_\tau(s_n)$. The set of all schedulers for $\mathcal{C}$ is denoted by $\mathfrak{S}(\mathcal{C})$.

[1] For the sake of simplicity, we only consider deterministic schedulers in this paper.

[2] More precisely, $\sigma^{-1}(s)$ is measurable in the product topology of the discrete topology on S and the Borel topology on $\mathbb{R}_{\geq 0}$.

The decision of the scheduler $\sigma(\mathfrak{h})$ determines t_n and s_{n+1} as follows. If $\mathrm{succ}_\tau(s_n) \neq \emptyset$, then the run proceeds immediately, i.e. in time $t_n := 0$, to the state $s_{n+1} := \sigma(\mathfrak{h})$. Otherwise, if $\mathrm{succ}_\tau(s_n) = \emptyset$, then only Markovian transitions are available in s_n. In such a case, the run moves to a randomly chosen next state s_{n+1} with probability $\mathbf{P}(s_n, s_{n+1})$ after waiting for a random time t_n chosen according to the exponential distribution with the rate $\mathbf{E}(s_n)$.

One of the fundamental problems in verification and performance analysis of continuous-time stochastic systems is the time-bounded reachability. Given a set of goal states $G \subseteq S$ and a time bound $T \in \mathbb{R}_{\geq 0}$, the *value of time-bounded reachability* is defined as $\sup_{\sigma \in \mathfrak{S}(\mathcal{C})} \mathcal{P}_{\mathcal{C}}^{\sigma}\left[\Diamond^{\leq T} G\right]$ where $\mathcal{P}_{\mathcal{C}}^{\sigma}\left[\Diamond^{\leq T} G\right]$ denotes the probability that a run of $\mathcal{C}$ under the scheduler σ visits a state of G before time T. The pivotal problem in the algorithmic analysis of IMC is to compute this value together with a scheduler that achieves the supremum. As the value is not rational in most cases, the aim is to provide an efficient approximation algorithm and compute an ε-optimal scheduler. The value of time-bounded reachability can be approximated up to a given error tolerance $\varepsilon > 0$ in time $\mathcal{O}(|S|^2 \cdot (\lambda T)^2/\varepsilon)$ [29], where λ is the maximal rate of $\mathcal{C}$, and the procedure also yields an ε-optimal scheduler. We generalize both the notion of the value as well as approximation algorithms to the setting of *open* IMC, i.e. those that are not closed, and motivate this extension in the next section.

3 Compositional Verification

In this section we turn our attention to the central questions studied in this paper. How can we decide how well an IMC component $\mathcal{C}$ performs (w.r.t. time-bounded reachability) when acting in parallel with an unknown environment? And how to control the component to establish a guarantee as high as possible?

Speaking thus far in vague terms, this amounts to finding a scheduler σ for $\mathcal{C}$ which maximizes the probability of reaching a target set G before T no matter what environment E is composed with $\mathcal{C}$. As we are interested in compositional modeling using IMC, the environments are supposed to be IMC with the same external actions as $\mathcal{C}$ (thus resolving the external non-determinism of $\mathcal{C}$). We also need to consider all resolutions of the internal non-determinism of E as well as the non-determinism arising from synchronization of $\mathcal{C}$ and E using another scheduler π. So we are interested in the following value:

$$\sup_{\sigma} \inf_{E,\pi} \mathcal{P}[G \text{ is reached in composition of } \mathcal{C} \text{ and } E \text{ before } T \text{ using } \sigma \text{ and } \pi].$$

Now, let us be more formal and fix an IMC $\mathcal{C} = (S, \mathbb{A}\mathrm{ct}^\tau, \hookrightarrow, \rightsquigarrow, s_0)$. For a given environment IMC E with the same action alphabet $\mathbb{A}\mathrm{ct}^\tau$, we introduce a composition

$$\mathcal{C}(E) = (\mathcal{C} \parallel_{\mathbb{A}\mathrm{ct}} E) \setminus \mathbb{A}\mathrm{ct}$$

where all open actions are hidden, yielding a closed system. Note that the states of $\mathcal{C}(E)$ are pairs (c, e) where c is a state of $\mathcal{C}$ and e is a state of E. We consider a scheduler σ of $\mathcal{C}$ and a scheduler π of $\mathcal{C}(E)$ respecting σ on internal actions of $\mathcal{C}$. We say that π *respects* σ, denoted by $\pi \in \mathfrak{S}(\mathcal{C}(E), \sigma)$, if for every history $\mathfrak{h} = (c_0, e_0)\, t_0 \cdots t_{n-1} (c_n, e_n)$ of $\mathcal{C}(E)$ the scheduler π satisfies one of the following conditions:

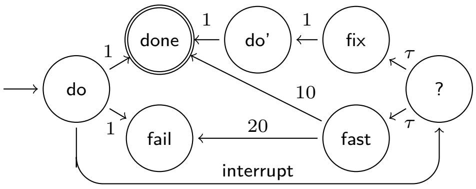
- $\pi(\mathfrak{h}) = (c, e)$ where $c_n \xhookrightarrow{a} c$ and $e_n \xhookrightarrow{a} e$ (π resolves synchronization)
- $\pi(\mathfrak{h}) = (c_n, e)$ where $e_n \xhookrightarrow{\tau} e$ (π chooses a move in the environment)
- $\pi(\mathfrak{h}) = (\sigma(\mathfrak{h}_{\mathcal{C}}), e_n)$ where $\mathfrak{h}_{\mathcal{C}} = c_0 t_0 \cdots t_{n-1} c_n$ (π chooses a move in $\mathcal{C}$ according to σ).

Given a set of goal states $G \subseteq S$ and a time bound $T \in \mathbb{R}_{\geq 0}$, the *value of compositional* *time-bounded reachability* is defined as

$$\sup_{\sigma\in\mathfrak{S}(\mathcal{C})} \inf_{\substack{E\in\mathrm{ENV}\\ \pi\in\mathfrak{S}(\mathcal{C}(E),\sigma)}} \mathcal{P}^{\pi}_{\mathcal{C}(E)}\left[\Diamond^{\leq T} G_E\right] \qquad (*)$$

where ENV denotes the set of all IMC with the action alphabet $\mathbb{A}\mathrm{ct}^\tau$ and $G_E = G \times S_E$ where S_E is the set of states of E. As for the closed IMC, our goal is to efficiently approximate this value together with a maximizing scheduler. Before we present an approximation algorithm based on discretization, we illustrate some of the effects of the open system perspective.

Example. The figure on the right depicts an IMC on which we approximate the value $(*)$ for $T = 2$ and $G = \{\mathsf{done}\}$. From the initial state do, the system may go randomly either to the target done or to fail. Concurrently, the external action interrupt may switch the run to the state ?, where the scheduler σ chooses between two successors (1) the state fast allowing fast but risky run to the target and (2) the state fix that guarantees reaching the target but takes longer time. The value $(*)$ is approximately 0.47 and an optimal scheduler goes to fix only if there are more than 1.2 minutes left. Note that the probability of reaching the target in time depends on when the external action interrupt is taken. The most "adversarial" environment executes interrupt after 0.8 minutes from the start.



Results. We now formulate our main result concerning efficient approximation of the value of compositional time-bounded reachability. In fact, we provide an approximation algorithm for a restricted subclass of IMC defined by the following two assumptions:

▶ **Assumption 1.** *Each cycle contains a Markovian transition.*

This assumption is standard over all analysis techniques published for IMC [18, 27, 23, 26, 34, 19]. It implies that the probability of taking infinitely many transitions in finite time, i.e. of Zeno behavior, is zero. This is a rather natural assumption and does not restrict the modeling power much, since no real system will be able to take infinitely many transitions in finite time anyway. Furthermore, the assumed property is a compositional one, i.e. it is preserved by parallel composition and hiding.

▶ **Assumption 2.** *Internal and external actions are not enabled at the same time, i.e. for each state s, either* $\mathrm{succ}_e(s) = \emptyset$ *or* $\mathrm{succ}_\tau(s) = \emptyset$.

Note that both assumptions are met by the above mentioned example. However, Assumption 2 is not compositional; specifically, it is not preserved by applications of the hiding operator. A stronger assumption would require the environment not to trigger external actions in zero time after a state change. This is indeed implied by Assumption 2 which basically asks *internal* transitions of the component to be executed before any *external* actions are taken into account.[3] In fact, the reverse precedence cannot be implemented in real systems, if internal actions are assumed to be executed without delay. Any procedure implemented in $\mathcal{C}$ for checking the availability of external actions will involve some non-zero delay (unless one resorts to quantum effects). From a technical point of view, lifting Assumption 2 makes the studied problems considerably more involved; see Section 6 for further discussion.

[3] To see this one can construct a weak simulation relation between a system violating Assumption 2 and one satisfying it, where any state with both internal and external transitions is split into two: the first one enabling the internal transitions and a new τ to the second one only enabling the external ones.

▶ **Theorem 5.** *Let $\varepsilon > 0$ be an approximation bound and $\mathcal{C} = (S, \mathbb{A}\mathrm{ct}^\tau, \hookrightarrow, \rightsquigarrow, s_0)$ be an IMC satisfying Assumptions 1 and 2. Then one can approximate the value of compositional time-bounded reachability of $\mathcal{C}$ up to ε and compute an ε-optimal scheduler in time $\mathcal{O}(|S|^2 \cdot (\lambda T)^2/\varepsilon)$, where λ is the maximal rate of $\mathcal{C}$ and T is the reachability time-bound.*

In the remainder of the paper, we prove this theorem and discuss its restrictions. First, we introduce a new kind of real-time games, called CE games, that are played on open IMC. Then we reduce the compositional time-bounded reachability of $\mathcal{C}$ to time-bounded reachability objective in the CE game played just on the component $\mathcal{C}$ (see Proposition 6). In Section 5, we show how to reduce, using discretization, the time-bounded reachability in CE games to step-bounded reachability in discrete-time stochastic games (see Proposition 8), that in turn can be solved using simple backward propagation. Finally, we show, in Proposition 9, how to transform optimal strategies in the discretized stochastic games to ε-optimal schedulers for $\mathcal{C}$.

4 Game of Controller and Environment

In order to approximate $(*)$, the value of compositional time-bounded reachability, we turn the IMC $\mathcal{C}$ into a two-player *controller–environment game* (CE game) $\mathcal{G}$. The CE game naturally combines two approaches to real-time systems, namely the *stochastic* flow of time as present in CTMC with the *non-deterministic* flow of time as present in timed automata. The game $\mathcal{G}$ is played on the graph of an IMC $\mathcal{C}$ played by two players: **con** (controlling the component $\mathcal{C}$) and **env** (controlling/simulating the environment). In essence, **con** chooses in each state with internal transitions one of them, and **env** chooses in each state with external (and hence synchronizing) transitions either which of them should be taken, or a delay $t_e \in \mathbb{R}_{>0}$. Note that, due to Assumption 2, the players control the game in disjoint sets of states, hence $\mathcal{G}$ is a turn-based game. The internal and external transitions take zero time to be executed once chosen. If no zero time transition is chosen, the delay t_e determined by **env** competes with the Markovian transitions, i.e. with a random time sampled from the exponential distribution with the rate $\mathbf{E}(s)$. We consider time-bounded reachability objective, so the goal of **con** is to reach a given subset of states G before a given time T, and **env** opposes it.

Formally, let us fix an IMC $\mathcal{C} = (S, \mathbb{A}\mathrm{ct}^\tau, \hookrightarrow, \rightsquigarrow, s_0)$ and thus a CE game $\mathcal{G}$. A *run* of $\mathcal{G}$ is again an infinite sequence $s_0\, t_0\, s_1\, t_1 \cdots$ where $s_n \in S$ is the n-th visited state and $t_n \in \mathbb{R}_{\geq 0}$ is the time spent there. Based on the *history* $s_0\, t_0 \cdots t_{n-1}\, s_n$ went through so far, the players choose their moves as follows.

- If $\mathrm{succ}_\tau(s_n) \neq \emptyset$, the player **con** chooses a state $s_\tau \in \mathrm{succ}_\tau(s_n)$.
- Otherwise, the player **env** chooses either a state $s_e \in \mathrm{succ}_e(s_n)$, or a delay $t_e \in \mathbb{R}_{>0}$. (Note that if $\mathrm{succ}_e(s_n) = \emptyset$ only a delay can be chosen.)

Subsequently, Markovian transitions (if available) are resolved by randomly choosing a target state s_M according to the distribution $\mathbf{P}(s_n, \cdot)$ and randomly sampling a time t_M according to the exponential distribution with rate $\mathbf{E}(s_n)$. The next waiting time t_n and state s_{n+1} are given by the following rules in the order displayed.

- If $\mathrm{succ}_\tau(s_n) \neq \emptyset$ and s_τ was chosen, then $t_n = 0$ and $s_{n+1} = s_\tau$.
- If s_e was chosen, then $t_n = 0$ and $s_{n+1} = s_e$.
- If t_e was chosen then:
  - if $\mathrm{succ}_M(s_n) = \emptyset$, then $t_n = t_e$ and $s_{n+1} = s_n$;
  - if $t_e \leq t_M$, then $t_n = t_e$ and $s_{n+1} = s_n$;
  - if $t_M < t_e$, then $t_n = t_M$ and $s_{n+1} = s_M$.

According to the definition of schedulers in IMC, we formalize the choice of **con** as a *strategy* $\sigma : (S \times \mathbb{R}_{\geq 0})^* \times S \to S$ and the choice of **env** as a strategy $\pi : (S \times \mathbb{R}_{\geq 0})^* \times S \to S \cup \mathbb{R}_{>0}$. We denote by Σ and Π the sets of all strategies of the players **con** and **env**, respectively. In order to keep CE games out of Zeno behavior, we consider in Π only those strategies of the player **env** for which the induced Zeno runs have zero measure, i.e. the sum of the chosen delays diverges almost surely no matter what **con** is doing.

Given goal states $G \subseteq S$ and a time bound $T \in \mathbb{R}_{\geq 0}$, the *value of* $\mathcal{G}$ is defined as

$$\sup_{\sigma \in \Sigma} \inf_{\pi \in \Pi} \mathcal{P}_{\mathcal{G}}^{\sigma,\pi}\left[\Diamond^{\leq T} G\right] \tag{$**$}$$

where $\mathcal{P}_{\mathcal{G}}^{\sigma,\pi}\left[\Diamond^{\leq T} G\right]$ is the probability of all runs of $\mathcal{G}$ induced by σ and π and reaching a state of G before time T. We now show that the value of the CE game coincides with the value of compositional time-bounded reachability. This result is interesting and important as it allows us to replace unknown probabilistic behaviour with non-deterministic choices.

▶ **Proposition 6.** *$(*) = (**)$, i.e.*

$$\sup_{\sigma \in \mathfrak{S}(\mathcal{C})} \inf_{\substack{E \in \mathrm{ENV} \\ \pi \in \mathfrak{S}(\mathcal{C}(E),\sigma)}} \mathcal{P}_{\mathcal{C}(E)}^{\pi}\left[\Diamond^{\leq T} G_E\right] = \sup_{\sigma \in \Sigma} \inf_{\pi \in \Pi} \mathcal{P}_{\mathcal{G}}^{\sigma,\pi}\left[\Diamond^{\leq T} G\right]$$

Proof Idea. We start with the inequality $(*) \geq (**)$. Let $\sigma \in \Sigma$ $(= \mathfrak{S}(\mathcal{C}))$ and let us fix an environment E together with a scheduler $\pi \in \mathfrak{S}(\mathcal{C}(E), \sigma)$. The crucial observation is that the purpose of the environment E (controlled by π) is to choose delays of external actions (the delay is determined by a sequence of internal and Markovian actions of E executed before the external action), which is in fact similar to the role of the player **env** in the CE game. The only difference is that the environment E "chooses" the delays randomly as opposed to deterministic strategies of **env**. However, using a technically involved argument, we show how to get rid of this randomization and obtain a strategy π' in the CE game satisfying $\mathcal{P}_{\mathcal{G}}^{\sigma,\pi'}\left[\Diamond^{\leq T} G\right] \leq \mathcal{P}_{\mathcal{C}(E)}^{\pi}\left[\Diamond^{\leq T} G_E\right]$.

Concerning the second inequality $(*) \leq (**)$, we show that every strategy of **env** can be (approximately) implemented using a suitable environment together with a scheduler π. The idea is to simulate every deterministic delay, say t, chosen by **env** using a random delay tightly concentrated around t (roughly corresponding to an Erlang distribution) that is implemented as an IMC. We show that the imprecision of delays introduced by this randomization induces only negligible alteration to the value. ◀

5 Discretization

In this section we show how to approximate the value $(**)$ of the CE game up to an arbitrarily small error $\varepsilon > 0$ by reduction to a discrete-time (turn-based) stochastic game Δ.

A stochastic game Δ is played on a graph $(V, \mapsto)$ partitioned into $V_\Box \uplus V_\Diamond \uplus V_\bigcirc$. A play starts in the initial vertex v_0 and forms a run $v_0 v_1 \cdots$ as follows. For a history $v_0 \cdots v_i$, the next vertex v_{i+1} satisfying $v_i \mapsto v_{i+1}$ is determined by a strategy $\sigma \in \Sigma_\Delta$ of player $\Box$ if $v_i \in V_\Box$ and by a strategy $\pi \in \Pi_\Delta$ of player $\Diamond$ if $v_i \in V_\Diamond$. Moreover, v_{i+1} is chosen randomly according to a fixed distribution $\mathit{Prob}(v_i)$ if $v_i \in V_\bigcirc$. For a formal definition, see, e.g., [17].

Let us fix a CE game $\mathcal{G}$ and a discretization step $\delta > 0$ that divides the time bound T into $N \in \mathbb{N}$ intervals of equal length (here $\delta = T/N$). We construct a discrete-time stochastic game Δ by substituting each state of $\mathcal{G}$ by a gadget of one or two vertices (as illustrated in Figure 1).[4] Intuitively, the game Δ models passing of time as follows. Each discrete step

[4] We assume w.l.o.g. that (1) states with internal transitions have no Markovian transitions available and

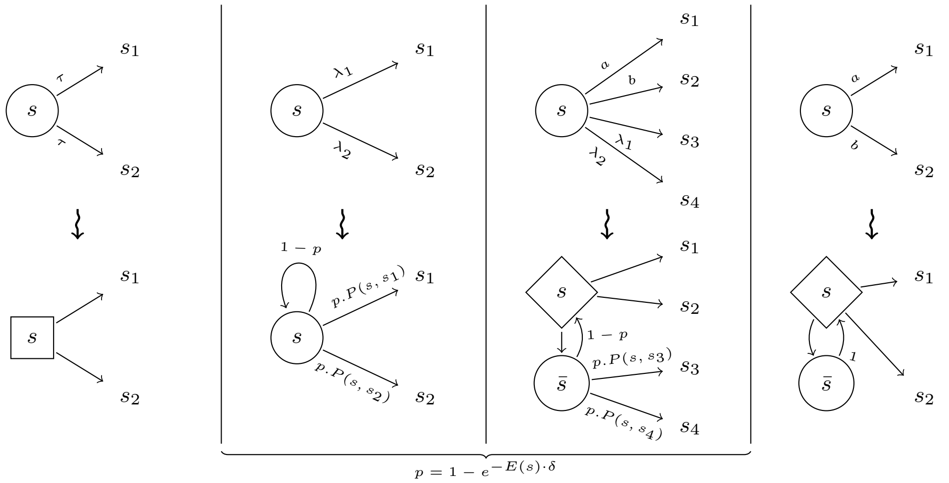


Figure 1 Four gadgets for transforming a CE game into a discrete game. The upper part shows types of states in the original CE game, the lower part shows corresponding gadgets in the transformed discrete game. In the lower part, the square-shaped, diamond-shaped and circle-shaped vertices belong to $V_\Box$, $V_\Diamond$ and $V_\bigcirc$, respectively. Binary branching is displayed only in order to simplify the figure.

"takes" either time δ or time 0. Each step from a vertex of $V_\bigcirc$ takes time δ whereas each step from vertex of $V_\Box \cup V_\Diamond$ takes zero time. The first gadget transforms internal transitions into edges of player $\Box$ taking zero time. The second gadget transforms Markovian transitions into edges of player $\bigcirc$ taking time δ where the probability p is the probability that any Markovian transition is taken in $\mathcal{G}$ before time δ. The third gadget deals with states with both external and Markovian transitions available where the player $\Diamond$ decides in vertex s in zero time whether an external transition is taken or whether the Markovian transitions are awaited in $\bar{s}$ for time δ. The fourth gadget is similar, but no Markovian transition can occur and from $\bar{s}$ the play returns into s with probability 1.

Similarly to $(*)$ and $(**)$, we define the *value of the discrete-time game* Δ as

$$\sup_{\sigma\in\Sigma_\Delta} \inf_{\pi\in\Pi_\Delta} \mathcal{P}^{\sigma,\pi}_{\Delta}\left[\Diamond^{\#_\circ \leq N} G\right] \qquad (***)$$

where $\mathcal{P}^{\sigma,\pi}_{\Delta}\left[\Diamond^{\#_\circ \leq N} G\right]$ is the probability of all runs of Δ induced by σ and π that reach G before taking more than N steps from vertices in $V_\bigcirc$. According to the intuition above, such a step bound corresponds to a time bound $N \cdot \delta = T$.

We say that a strategy *is counting* if it only considers the last vertex and the current count $\#_\circ$ of steps taken from vertices in $V_\bigcirc$. We may represent it as a function $V \times \{0,\ldots,N\} \to V$ since it is irrelevant what it does after more than N steps.

▶ **Lemma 7.** *There are counting strategies optimal in $(***)$. Moreover, they can be computed together with $(***)$ in time $\mathcal{O}(N|V|^2)$.*

We now show that the value $(***)$ of the discretized game Δ approximates the value $(**)$ of the CE game $\mathcal{G}$ and give the corresponding error bound.

(2) every state has at least one outgoing transition.This is no restriction since (1) Markovian transitions are never taken in such states and (2) any state without transitions can be endowed with a Markovian self-loop transition without changing the time-bounded reachability.

▶ **Proposition 8** (Error bound). *For every approximation bound $\varepsilon > 0$ and discretization step $\delta \le \varepsilon/(\lambda^2 T)$ where $\lambda = \max_{s\in S} \mathbf{E}(s)$, the value $(***)$ induced by δ satisfies*

$$(***) \;\le\; (**) \;\le\; (***) + \varepsilon.$$

Proof Idea. The proof is inspired by the techniques for closed IMC [29]. Yet, there are several new issues to overcome, caused mainly by the fact that the player **env** in the CE game may choose an arbitrary real delay $t_e > 0$ (so **env** has uncountably many choices). The discretized game Δ is supposed to simulate the original CE game but restricts possible behaviors as follows: (1) Only one Markovian transition is allowed in any interval of length δ. (2) The delay t_e chosen by player $\Diamond$ (which simulates the player **env** from the CE game) must be divisible by δ. We show that none of these restrictions affects the value.

ad (1) As pointed out in [29], the probability of two or more Markovian transitions occurring in an interval $[0, \delta]$ is bounded by $(\lambda\delta)^2/2$ where $\lambda = \max_{s\in S} \mathbf{E}(s)$. Hence, the probability of multiple Markovian transitions occurring in any of the discrete steps of Δ is $\le \varepsilon$.

ad (2) Assuming that at most one Markovian transition is taken in $[0, \delta]$ in the CE game, we reduce the decision when to take external transitions to minimization of a linear function on $[0, \delta]$, which in turn is minimized either in 0, or δ. Hence, the optimal choice for the player **env** in the CE game is either to take the transitions immediately at the beginning of the interval (before the potential Markovian transition) or to wait for time δ (after the potential Markovian transition). ◀

Finally, we show how to transform an optimal counting strategy $\sigma : V \times \{0, \dots, N\} \to V$ in the discretized game Δ into an ε-optimal scheduler $\overline{\sigma}$ in the IMC $\mathcal{C}$. For every $\mathfrak{p} = s_0\, t_0 \cdots s_{n-1}\, t_{n-1}\, s_n$ we put $\overline{\sigma}(\mathfrak{p}) = \sigma(s_n, \lceil (t_0 + \ldots + t_{n-1})/\delta \rceil)$.

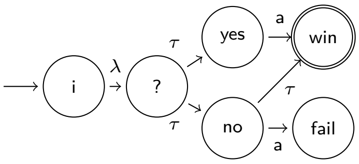
▶ **Proposition 9** (ε-optimal scheduler). *Let $\varepsilon > 0$, Δ be a corresponding discrete game, and $\overline{\sigma}$ be induced by an optimal counting strategy in Δ, then*

$$(*) \;\le\; \inf_{\substack{E\in\mathrm{ENV}\\ \pi\in\mathfrak{S}(\mathcal{C}(E),\overline{\sigma})}} \mathcal{P}^{\pi}_{\mathcal{C}(E)}\left[\Diamond^{\le T} G_E\right] + \varepsilon$$

This together with the complexity result of Lemma 7 finishes the proof of Theorem 5.

6 Summary, Discussion and Future Work

We discussed the computation of maximal timed bounded reachability for IMC operating in an unknown IMC environment to synchronize with. All prior analysis approaches considered closed systems, implicitly assuming that external actions do happen in zero time. Our analysis for open IMC works essentially with the opposite assumption, which is arguably more realistic. We have shown that the resulting stochastic two-player game has the same extremal values as a CE-game, where the player controlling the environment can choose exact times. The latter is approximated up to a given precision by discretization and the resulting control strategy translated back to a scheduler of the IMC achieving the bound.



Finally, we argue that lifting Assumption 2 makes analysis considerably more involved as the studied game may contain imperfect information and concurrent decisions. Let us illustrate the problems on an example. Consider an IMC depicted on the right. This IMC violates Assumption 2 in its state no. Let us fix an arbitrary environment E (controlled by π) and a scheduler σ. Since internal transitions of E take zero time, the environment must spend almost all the time in states without internal transitions. Hence, E is almost surely in such a state when ? is entered. Assume E is in a state with the (external) action a being available. The scheduler σ wins if he chooses the internal transition to yes since the synchronizing transition a is then taken immediately, and fails if he chooses to proceed to no, as a (reasonable) scheduler π will now force synchronization on action a. If, otherwise, on entering state ?, E is in a state without the action a being available, the scheduler σ fails if he chooses yes because a (reasonable) environment never synchronizes, and wins if he chooses no since the environment E cannot immediately synchronize and the τ transition is taken. Note that the scheduler σ cannot observe whether a is available in the current state of E. As this is crucial for the further evolution of the game from state ?, the game is intrinsically of imperfect information.

We conjecture that solving even this special case of imperfect information games is PSPACE-hard. Yet, the complexity might only increase in the number of internal transitions that can be taken in a row. For systems, where a bound on the length of internal transition sequences can be assumed, this problem would then still be feasible.

Acknowledgement

The work has been supported by the Czech Science Foundation, grant No. P202/12/P612 (T. Brázdil, J. Křetínský and V. Řehák) and No. 102/09/H042 (J. Krčál) and by the DFG as part of the SFB/TR 14 AVACS, by the DFG/NWO project ROCKS, and by the EU FP7 project MEALS, grant agreement no. 295261. (H. Hermanns).

References

1 K. Apt and E. Grädel, editors. *Lectures in Game Theory for Computer Scientists.* Cambridge, 2011.

2 C. Baier, H. Hermanns, J.-P. Katoen, and B.R. Haverkort. Efficient computation of time-bounded reachability probabilities in uniform continuous-time Markov decision processes. *Theor. Comp. Sci.*, 345(1):2–26, 2005.

3 J.A. Bergstra, A. Ponse, and S.A. Smolka, editors. *Handbook of Process Algebra.* Elsevier, 2001.

4 E. Böde, M. Herbstritt, H. Hermanns, S. Johr, T. Peikenkamp, R. Pulungan, J. Rakow, R. Wimmer, and B. Becker. Compositional dependability evaluation for STATEMATE. *IEEE Trans. on Soft. Eng.*, 35(2):274–292, 2009.

5 H. Boudali, P. Crouzen, B.R. Haverkort, M. Kuntz, and M. I. A. Stoelinga. Architectural dependability evaluation with Arcade. In *Proc. of DSN*, pages 512–521. IEEE, 2008.

6 H. Boudali, P. Crouzen, and M. Stoelinga. A rigorous, compositional, and extensible framework for dynamic fault tree analysis. *IEEE Trans. on DSC*, 7(2):128–143, 2010.

7 P. Bouyer and V. Forejt. Reachability in stochastic timed games. In *Proc. of ICALP*, volume 5556 of *LNCS*, pages 103–114. Springer, 2009.

8 M. Bozzano, A. Cimatti, J.-P. Katoen, V.Y. Nguyen, T. Noll, and M. Roveri. Safety, dependability and performance analysis of extended AADL models. *The Computer Journal*, 54(5):754–775, 2011.

9 T. Brázdil, V. Forejt, J. Krčál, J. Křetínský, and A. Kučera. Continuous-time stochastic games with time-bounded reachability. In *Proc. of FSTTCS*, volume 4 of *LIPIcs*, pages 61–72. Schloss Dagstuhl, 2009.

10 T. Brázdil, H. Hermanns, J. Krčál, J. Křetínský, and V. Řehák. Verification of open interactive markov chains. Technical Report FIMU-RS-2012-04, Faculty of Informatics MU, 2012.

11 T. Brázdil, J. Krčál, J. Křetínský, A. Kučera, and V. Řehák. Stochastic real-time games with qualitative timed automata objectives. In *Proc. of CONCUR*, volume 6269 of *LNCS*, pages 207–221. Springer, 2010.

12 P. Buchholz and I. Schulz. Numerical Analysis of Continuous Time Markov Decision processes over Finite Horizons. *Computers and Operations Research*, 38:651–659, 2011.

13 K. Chatterjee and T.A. Henzinger. A survey of stochastic ω-regular games. *J. Comput. Syst. Sci.*, 78(2):394–413, 2012.

14 T. Chen, V. Forejt, M.Z. Kwiatkowska, D. Parker, and A. Simaitis. Automatic verification of competitive stochastic systems. In *Proc. of TACAS*, volume 7214 of *LNCS*, pages 315–330. Springer, 2012.

15 N. Coste, H. Hermanns, E. Lantreibecq, and W. Serwe. Towards performance prediction of compositional models in industrial GALS designs. In *Proc. of CAV*, volume 5643, pages 204–218. Springer, 2009.

16 M.-A. Esteve, J.-P. Katoen, V.Y. Nguyen, B. Postma, and Y. Yushtein. Formal correctness, safety, dependability and performance analysis of a satellite. In *Proc. of ICSE*. ACM and IEEE press, 2012.

17 J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer, 1996.

18 H. Garavel, R. Mateescu, F. Lang, and W. Serwe. CADP 2006: A toolbox for the construction and analysis of distributed processes. In *Proc. of CAV*, volume 4590 of *LNCS*, pages 158–163. Springer, 2007.

19 D. Guck, T. Han, J.-P. Katoen, , and M.R. Neuhäußer. Quantitative timed analysis of interactive markov chains. In *NFM*, volume 7226 of *LNCS*, pages 8–23. Springer, 2012.

20 E.M. Hahn, G. Norman, D. Parker, B. Wachter, and L. Zhang. Game-based abstraction and controller synthesis for probabilistic hybrid systems. In *QEST*, pages 69–78, 2011.

21 B.R. Haverkort, M. Kuntz, A. Remke, S. Roolvink, and M.I.A. Stoelinga. Evaluating repair strategies for a water-treatment facility using Arcade. In *Proc. of DSN*, pages 419–424, 2010.

22 H. Hermanns and S. Johr. Uniformity by construction in the analysis of nondeterministic stochastic systems. In *DSN*, pages 718–728. IEEE Computer Society, 2007.

23 H. Hermanns and S. Johr. May we reach it? Or must we? In what time? With what probability? In *Proc. of MMB*, pages 125–140. VDE Verlag, 2008.

24 H. Hermanns and J.-P. Katoen. The how and why of interactive Markov chains. In *Proc. of FMCO*, volume 6286 of *LNCS*, pages 311–337. Springer, 2009.

25 H. Hermanns, J.-P. Katoen, M. R. Neuhäußer, and L. Zhang. GSPN model checking despite confusion. Technical report, RWTH Aachen University, 2010.

26 J.-P. Katoen, D. Klink, and M. R. Neuhäußer. Compositional abstraction for stochastic systems. In *Proc. of FORMATS*, volume 5813 of *LNCS*, pages 195–211. Springer, 2009.

27 J.-P. Katoen, I.S. Zapreev, E.M. Hahn, H. Hermanns, and D.N. Jansen. The ins and outs of the probabilistic model checker MRMC. *Performance Evaluation*, 68(2):90–104, 2011.

28 O. Kupferman and M. Vardi. Module checking. In *CAV*, volume 1102 of *LNCS*, pages 75–86. Springer, 1996.

29 M.R. Neuhäußer. *Model checking nondeterministic and randomly timed systems*. PhD thesis, University of Twente, 2010.

30 M.N. Rabe and S. Schewe. Finite optimal control for time-bounded reachability in CTMDPs and continuous-time Markov games. *Acta Informatica*, 48(5-6):291–315, 2011.

31 P.J.G. Ramadge and W.M. Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77(1), 1989.

32 J. Sproston. Discrete-time verification and control for probabilistic rectangular hybrid automata. In *QEST*, pages 79–88, 2011.

33 P. Černý, K. Chatterjee, T.A. Henzinger, A. Radhakrishna, and R. Singh. Quantitative synthesis for concurrent programs. In *CAV*, pages 243–259, 2011.

34 L. Zhang and M.R. Neuhäußer. Model checking interactive Markov chains. In *Proc. of TACAS*, volume 6015 of *LNCS*, pages 53–68. Springer, 2010.