
ANALYSIS OF TIMED AND LONG-RUN OBJECTIVES FOR MARKOV AUTOMATA *

DENNIS GUCK ^a, HASSAN HATEFI ^b, HOLGER HERMANN ^c,
JOOST-PIETER KATOEN ^d, AND MARK TIMMER ^e

^{a,e} Formal Methods and Tools, University of Twente, The Netherlands
e-mail address: {d.guck,m.timmer}@utwente.nl

^{b,c} Dependable Systems and Software, Saarland University, Germany
e-mail address: {hhatefi,hermanns}@depend.cs.uni-saarland.de

^d Formal Methods and Tools, University of Twente, The Netherlands
Software Modelling and Verification, RWTH Aachen University, Germany
e-mail address: katoen@cs.rwth-aachen.de

ABSTRACT. Markov automata (MAs) extend labelled transition systems with random delays and probabilistic branching. Action-labelled transitions are instantaneous and yield a distribution over states, whereas timed transitions impose a random delay governed by an exponential distribution. MAs are thus a nondeterministic variation of continuous-time Markov chains. MAs are compositional and are used to provide a semantics for engineering frameworks such as (dynamic) fault trees, (generalised) stochastic Petri nets, and the Architecture Analysis & Design Language (AADL). This paper considers the quantitative analysis of MAs. We consider three objectives: expected time, long-run average, and timed (interval) reachability. Expected time objectives focus on determining the minimal (or maximal) expected time to reach a set of states. Long-run objectives determine the fraction of time to be in a set of states when considering an infinite time horizon. Timed reachability objectives are about computing the probability to reach a set of states within a given time interval. This paper presents the foundations and details of the algorithms and their correctness proofs. We report on several case studies conducted using a prototypical tool implementation of the algorithms, driven by the MAPA modelling language for efficiently generating MAs.

2012 ACM CCS: [**Mathematics of computing**]: Probability and statistics—Stochastic processes—Markov processes; [**Theory of computation**]: Models of computation; Formal languages and automata theory.

Key words and phrases: Quantitative analysis, Markov automata, continuous time, expected time, long-run average, timed reachability.

* This paper is the extended version of the QEST 2013 paper entitled “Modeling, Reduction, and Analysis of Markov Automata” [19]. The current paper focuses on the quantitative analysis of Markov automata, contains all full proofs, and has more extensive explanations.

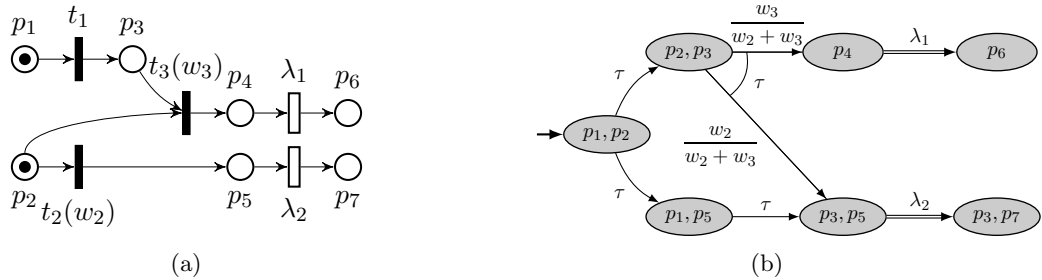


Figure 1: (a) Confused GSPN [1, Fig. 21] with partial weights and (b) its MA semantics.

1. INTRODUCTION

Markov automata (MAs, for short) have been introduced in [16] as a continuous-time version of Segala’s probabilistic automata [31]. Closed under operators such as parallel composition and hiding, they provide a compositional formalism for concurrent soft real time systems. A transition in an MA is either labelled with a positive real number representing the rate of a negative exponential distribution, or with an action. An action transition leads to a discrete probability distribution over states. MAs can thus model action transitions as in labelled transition systems, probabilistic branching as found in (discrete time) Markov chains and Markov decision processes, as well as delays that are governed by exponential distributions as in continuous-time Markov chains.

The semantics of MAs has been recently investigated in quite some detail. Weak and strong (bi)simulation semantics have been presented in [16, 15], whereas it is shown in [13] that weak bisimulation provides a sound and complete proof methodology for reduction barbed congruence. A process algebra with data for the efficient modelling of MAs, accompanied with some reduction techniques using static analysis, has been presented in [35], and model checking of MAs against Continuous Stochastic Logic (CSL) is discussed in [21]. Although the MA model raises several challenging theoretical issues, both from a semantical and from an analytical point of view, our main interest is in their practical applicability. As MAs extend Hermanns’ interactive Markov chains (IMCs) [23], they inherit IMC application domains, ranging from GALS hardware designs [9] and dynamic fault trees [6] to the standardised modelling language AADL [7, 22]. The additional feature of probabilistic branching yields additional expressivity and thereby enriches the spectrum of application contexts further. This expressivity also makes them a natural semantic model for other formalisms. Among others, MAs are expressive enough to provide a natural operational model for generalised stochastic Petri nets (GSPNs) [2] and stochastic activity networks (SANs) [27], both popular modelling formalisms for performance and dependability analysis. Let us briefly motivate this by considering GSPNs. Whereas in SPNs all transitions are subject to an exponentially distributed delay, GSPNs also incorporate immediate transitions, transitions that happen instantaneously. The traditional GSPN semantics yields a continuous-time Markov chain (CTMC), i.e., an MA without action transitions. However, that semantics is restricted to a subclass of GSPNs, namely those that are *confusion free*. Confusion [1] is related to the presence of nondeterminism. Confused GSPNs are traditionally considered as semantically ambiguous and thus precluded from any kind of

analysis. This gap is particularly disturbing because several published semantics for higher-level modelling formalisms—e.g., UML, AADL, WSDL—map onto GSPNs without ensuring the mapping to be free of confusion, therefore possibly inducing confused models.

It has recently been detailed in [24, 14] that MAs are a natural semantic model for *every* GSPN. To give some intuitive insight into this achievement, consider the GSPN in Fig. 1(a). This net is confused: In Petri net jargon, the transitions t_1 and t_2 are not in conflict, but firing transition t_1 leads to a conflict between t_2 and t_3 , which does not occur if t_2 fires before t_1 . Though decisive, the firing order between t_1 and t_2 is not determined. Transitions t_2 and t_3 are weighted so that in a marking $\{p_2, p_3\}$ in which both transitions are enabled, t_2 fires with probability $\frac{w_2}{w_2+w_3}$ and t_3 with its complement probability. The weight of transition t_1 is not relevant; we assume t_1 is not equipped with a weight. Classical GSPN semantics and analysis algorithms cannot cope with this net due to the presence of confusion (i.e., nondeterminism). Figure 1(b) depicts the MA semantics of this net. Here, states correspond to sets of net places that contain a token. In the initial state, there is a nondeterministic choice between the transitions t_1 and t_2 . Note that the presence of weights is naturally represented by discrete probabilistic branching as reflected in the outgoing transition from state $\{p_2, p_3\}$. One can show that the MA semantics conservatively extends the classical semantics, in the sense that the former and the latter are weakly bisimilar [14] on confusion-free GSPNs. Thus, if transition t_1 in our example is assigned some weight w_1 , the GSPN has no confusion. This would be reflected in the MA semantics by replacing the nondeterministic branching in state $\{p_1, p_2\}$ by a single transition, yielding $\{p_2, p_3\}$ with probability $\frac{w_1}{w_1+w_2}$ and state $\{p_1, p_5\}$ with the complement probability.

This paper focuses on the quantitative analysis of MAs—and thus implicitly of (possibly confused) GSPNs, of AADL specifications containing error models, and so on. We present analysis algorithms for three objectives: expected time, long-run average, and timed (interval) reachability. As the model exhibits nondeterminism, we focus on maximal and minimal values for all three objectives. We show that expected-time and long-run average objectives can be efficiently reduced to well-known problems on MDPs such as stochastic shortest path, maximal end-component decomposition, and long-run ratio objectives. This generalises (and slightly improves) the results reported in [18] for IMCs to MAs. Secondly, we present a discretisation algorithm for timed interval reachability objectives which extends [38]. Finally, we present the MAMA tool chain, an easily accessible publicly available tool chain¹ for the specification, mechanised simplification—such as confluence reduction [36], a form of on-the-fly partial-order reduction—and quantitative evaluation of MAs. We describe the overall architectural design, as well as the tool components, and report on empirical results obtained with MAMA on a selection of case studies taken from different domains. The experiments give insight into the effectiveness of the reduction techniques in MAMA and demonstrate that MAs provide the basis of a very expressive stochastic timed modelling approach without sacrificing the ability of time and memory efficient numerical evaluation.

Organisation of the paper. We introduce Markov automata in Section 2. Section 3 considers the evaluation of expected-time properties. Section 4 discusses the analysis of long-run properties, and Section 5 focuses on timed reachability properties with time-interval bounds.

¹Stand-alone download as well as web-based interface available from <http://fmt.cs.utwente.nl/~timmer/mama>.

Implementation details of our tool, a compositional modelling formalism as well as experimental results are discussed in detail in Section 6. Section 7 concludes the paper. We provide the proofs for our main results in the appendix.

2. PRELIMINARIES

2.1. Markov automata. An MA is a transition system with two types of transitions: probabilistic (as in PAs) and Markovian transitions (as in CTMCs). Let Act be a countable universe of actions with internal action $\tau \in Act$, and $\text{Distr}(S)$ denote the set of discrete probability distribution functions over the countable set S . Let α, β range over Act and μ, ν over $\text{Distr}(S)$. Actions such as α can be used for interaction with other MAs [16]. This does not apply to the internal action τ , which is executed autonomously.

Definition 2.1 (Markov automaton). A *Markov automaton* (MA) is a tuple $\mathcal{M} = (S, A, \rightarrow, \Longrightarrow, s_0)$ where S is a nonempty, finite set of states with *initial state* $s_0 \in S$, $A \subseteq Act$ is a finite set of actions with $\tau \in A$, and

- $\rightarrow \subseteq S \times A \times \text{Distr}(S)$ is the *probabilistic* transition relation, and
- $\Longrightarrow \subseteq S \times \mathbb{R}_{>0} \times S$ is the *Markovian* transition relation.

We abbreviate $(s, \alpha, \mu) \in \rightarrow$ by $s \xrightarrow{\alpha} \mu$ and $(s, \lambda, s') \in \Longrightarrow$ by $s \xrightarrow{\lambda} s'$. An MA can evolve via its probabilistic and Markovian transitions. If $s \xrightarrow{\alpha} \mu$, it can leave state s by executing the action α , after which the probability of going to some state $s' \in S$ is given by $\mu(s')$. If $s \xrightarrow{\lambda} s'$ is the only transition emanating from s , a state transition from s to s' can occur after an exponentially distributed delay with rate λ . That is to say, the expected delay from s to s' is $\frac{1}{\lambda}$. If $s \xrightarrow{\lambda} s'$ and $s \xrightarrow{\tau} \mu$ for some μ , however, always the τ -transition is taken and never the Markovian one. This is the *maximal progress* assumption [16]. The rationale behind this assumption is that internal (i.e., τ -labelled) transitions are not subject to interaction and thus can happen immediately, whereas the probability of a Markovian transition to immediately happen is zero. Thus, $s \xrightarrow{\lambda} s'$ almost never fires instantaneously. Note that the maximal progress assumption does not apply in case $s \xrightarrow{\lambda} s'$ and $s \xrightarrow{\alpha} \mu$ with $\alpha \neq \tau$, as α -transitions – unlike τ -transitions – can be used for synchronisation and thus be subject to a delay. In this case, the transition $s \xrightarrow{\lambda} s'$ may happen with positive probability. The semantics of several Markovian transitions in a state is as follows. For a state with one or more Markovian transitions, let $\mathbf{R}(s, s') = \sum \{\lambda \mid s \xrightarrow{\lambda} s'\}$ be the total rate of moving from state s to state s' , and let $E(s) = \sum_{s' \in S} \mathbf{R}(s, s')$ be the total outgoing rate of s . If s has more than one outgoing Markovian transition, a *competition* between its Markovian transitions exists. Then, the probability of moving from s to state s' within d time units is

$$\frac{\mathbf{R}(s, s')}{E(s)} \cdot (1 - e^{-E(s) \cdot d}).$$

After a delay of at most d time units (second factor) in state s , the MA moves to a direct successor state s' with probability $\mathbf{P}(s, s') = \frac{\mathbf{R}(s, s')}{E(s)}$. Note that also in this case, the maximal progress assumption applies: if $s \xrightarrow{\tau} \mu$ and s has several Markovian transitions, only the τ -transition can occur and no delay occurs in s . The behaviour of an MA in states with only Markovian transitions is thus the same as in CTMCs [3]. Fig. 2 depicts a sample MA. Note that this MA only contains τ -labelled probabilistic transitions; by maximal progress, any state has only Markovian transitions or only τ -labelled transitions. In case several

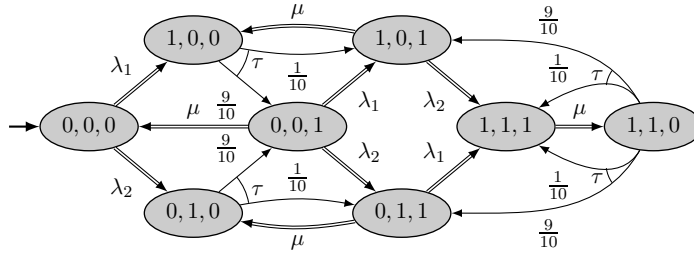


Figure 2: A queueing system (taken from [35]), consisting of a server and two stations. Each state is represented as a tuple (s_1, s_2, j) , with s_i the number of jobs in station i , and j the number of jobs in the server. The two stations have incoming requests with rates λ_1, λ_2 , which are stored until fetched by the server. If both stations contain a job, the server chooses nondeterministically (in state $(1,1,0)$). Jobs are processed with rate μ , and when polling a station, with probability $\frac{1}{10}$ the job is erroneously kept in the station after being fetched. For simplicity we assume that each component can hold at most one job.

τ -transitions emanate from a state, a nondeterministic choice between these transitions exists.

2.2. Actions. Actions different from τ can be used to compose MAs from smaller MAs using parallel composition. For instance, $\mathcal{M}_1 \parallel_H \mathcal{M}_2$ denotes the parallel composition of MA \mathcal{M}_1 and \mathcal{M}_2 in which actions in the set $H \subseteq Act$ with $\tau \notin H$ need to be executed by both MAs simultaneously, and actions not in H are performed autonomously by \mathcal{M}_i . In this paper, we will not cover the details of such composition operation (see [16]); it suffices to understand that the distinction between τ and $\alpha \neq \tau$ is relevant when composing MAs from component MAs. We assume in the sequel that the MAs to be analysed are *single, monolithic* MAs. These MAs are not subject to any interaction with other MAs. Hence, we assume that all transitions are labelled by τ -actions. (This amounts to the assumption that prior to the analysis all actions needed to compose several MAs are explicitly turned into internal actions by hiding.) Due to the maximal progress assumption, the outgoing transitions of each state are either all probabilistic or all Markovian. We can therefore partition the states into a set of probabilistic states, denoted $PS \subseteq S$, and a set of Markovian states, denoted $MS \subseteq S$. We denote the set of enabled actions in s with $Act(s)$, where $Act(s) = \{\alpha \in A \mid \exists \mu \in \text{Distr}(S) . s \xrightarrow{\alpha} \mu\}$ if $s \in PS$, and $Act(s) = \{\perp\}$ otherwise.

2.3. Paths. A path in an MA is an infinite sequence $\pi = s_0 \xrightarrow{\sigma_0, \mu_0, t_0} s_1 \xrightarrow{\sigma_1, \mu_1, t_1} \dots$ with $s_i \in S$, $\sigma_i = \tau$ or $\sigma_i = \perp$, $\mu_i \in \text{Distr}(S)$ and $t_i \in \mathbb{R}_{\geq 0}$. For $\sigma_i = \tau$, $s_i \xrightarrow{\sigma_i, \mu_i, t_i} s_{i+1}$ denotes that after residing $t_i = 0$ time units in s_i , the MA moved via action σ_i to s_{i+1} with probability $\mu_i(s_{i+1})$. In case $\sigma_i = \perp$, $s_i \xrightarrow{\perp, \mu_i, t_i} s_{i+1}$ denotes that after residing t_i time units in s_i , a Markovian transition led to s_{i+1} with probability $\mu_i(s_{i+1}) = \mathbf{P}(s_i, s_{i+1})$. For $t \in \mathbb{R}_{\geq 0}$, let $\pi@t$ denote the *sequence* of states that π occupies at time t . Due to instantaneous probabilistic transitions, $\pi@t$ is a sequence of states, as an MA may occupy various states at the same time instant. Let $Paths$ denote the set of infinite paths and $Paths^*$ be the set of finite prefixes thereof (called finite paths). The time elapsed along the infinite

path π is given by $\sum_{i=0}^{\infty} t_i$. Path π is Zeno whenever this sum converges. As the probability of a Zeno path in an MA that only contains Markovian transitions is zero [3, Prop. 1], an MA is non-Zeno if and only if no SCC with only probabilistic states is reachable (with positive probability). As such SCC contains no Markovian transitions, it can be traversed infinitely often without any passage of time. In the rest of this paper, we assume MAs to be non-Zeno.

2.4. Policies. Nondeterminism occurs when there is more than one probabilistic transition emanating from a state. To define a probability space over sets of infinite paths, we adopt the approach as for MDPs [30] and resolve the nondeterminism by a *policy*. A policy is a function that yields for each finite path ending in state s a probability distribution over the set of enabled transitions in s .

Formally, a policy is a function $D: Paths^* \rightarrow \text{Distr}((Act \cup \{\perp\}) \times \text{Distr}(S))$. Of course, policies should only choose from available transitions, so we require for each path π ending in a state s_n that $D(\pi)(\alpha, \mu) > 0$ implies $s_n \xrightarrow{\alpha} \mu$ and $D(\pi)(\perp, \mu) > 0$ implies that s_n is Markovian and $\mu = \mathbf{P}(s_n, \cdot)$. Let *GM* (generic measurable) denote the most general class of such policies that are still measurable; see [28] for details on measurability. In general, a policy randomly picks an enabled action and probability distribution in the final state of a given path. This is also known as a *history-dependent randomised* policy. If a policy always selects an action and probability distribution according to a Dirac distribution, it is called a *deterministic* policy. Policies are also classified based on the level of information they use for the resolution of nondeterminism. In the most general setting, a policy may use all information in a finite path, e.g., the states along the path, their ordering in the path, the amount of time spent in each state, and so forth. A *stationary* policy only bases its decision on the current state, and not on anything else. That is, D is stationary whenever $D(\pi_1) = D(\pi_2)$ for any finite paths π_1 and π_2 that have the same last state. A stationary deterministic policy can be viewed as a function $D: PS \rightarrow Act \times \text{Distr}(S)$ that maps each probabilistic state s to an action $\alpha \in Act$ and probability distribution $\mu \in \text{Distr}(S)$ such that $s \xrightarrow{\alpha} \mu$; such policies always take the same decision every time they are in the same state. A *time-abstract* policy resolves nondeterminism based on the alternating sequence of states and transitions visited so far, but not on the state residence times. Let *TA* denote the set of time-abstract policies. For more details on different classes of policies (and their relationship) on models such as MAs, we refer to [28]. Like for MDPs [30], a stationary or time-abstract policy on an MA induces a countable stochastic process that is equivalent to a (continuous-time) Markov chain. Using a standard cylinder-set construction on infinite paths in such Markov chains [3] we obtain a σ -algebra of subsets of *Paths*; given a policy D and an initial state s , a measurable set of paths is equipped with probability measure $\text{Pr}_{s,D}$.

To ease the development of the theory, and without loss of generality, we assume that each internal action induces a unique probability distribution. Note that this is no restriction: if there are multiple τ -transitions emerging from a state $s \in PS$, we may replace the τ by internal actions τ_1 to τ_n , where n is the out-degree of s .

2.5. Stochastic shortest path (SSP) problems. As some objectives on MAs can be reduced to SSP problems, we briefly introduce them. An MDP is a tuple (S, A, \mathbf{P}, s_0) where S is a finite set of states, $A \subseteq Act$ is a set of actions, $\mathbf{P}: S \times A \times S \rightarrow [0, 1]$ such that for each state s and each α , $\sum_{s' \in S} \mathbf{P}(s, \alpha, s') \in \{0, 1\}$, and $s_0 \in S$ is the initial state. It is assumed

that in each state at least one action is enabled, i.e., $\mathbf{P}(s, \alpha, s') > 0$ for each s , for some α . A *non-negative SSP problem* is a tuple $(S, A, \mathbf{P}, s_0, G, c, g)$ where the first four elements represent its underlying MDP accompanied by a set $G \subseteq S$ of goal states, cost function $c: (S \setminus G) \times A \rightarrow \mathbb{R}_{\geq 0}$ and terminal cost function $g: G \rightarrow \mathbb{R}_{\geq 0}$. A path through an MDP is an alternating sequence $s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \dots$ such that $\mathbf{P}(s_i, \alpha_i, s_{i+1}) > 0$, for all i . The accumulated cost along a path π through the MDP before reaching G , denoted by $C_G(\pi)$, is $\sum_{j=0}^{k-1} c(s_j, \alpha_j) + g(s_k)$ where k is the state index of reaching G . If π does not reach G , then $C_G(\pi)$ equals ∞ . As standard in MDPs [30], nondeterminism between different actions in a state is resolved using policies; similar to the notion for MAs, a stationary deterministic policy is a function $D: PS \rightarrow Act$. Let $cR^{\min}(s, \diamond G)$ denote the minimum expected cost reachability of G in the SSP (under all policies) when starting from s . It is a well-known result that stationary policies suffice to achieve $cR^{\min}(s, \diamond G)$. This expected cost can be obtained by solving an LP (linear programming) problem [5].

3. EXPECTED TIME OBJECTIVES

Let \mathcal{M} be an MA with state space S and $G \subseteq S$ a set of goal states. Define the (extended) random variable $V_G: Paths \rightarrow \mathbb{R}_{\geq 0}^{\infty}$ as the elapsed time before first visiting some state in G . That is, for an infinite path $\pi = s_0 \xrightarrow{\sigma_0, \mu_0, t_0} s_1 \xrightarrow{\sigma_1, \mu_1, t_1} \dots$, let $V_G(\pi) = \min \{t \in \mathbb{R}_{\geq 0} \mid G \cap \pi@t \neq \emptyset\}$ where $\min(\emptyset) = \infty$. (With slight abuse of notation we use $\pi@t$ as the set of states occurring in the sequence $\pi@t$.) The minimal expected time to reach G from $s \in S$ is defined by

$$eT^{\min}(s, \diamond G) = \inf_{D \in GM} \mathbb{E}_{s, D}(V_G) = \inf_{D \in GM} \int_{Paths} V_G(\pi) \cdot \Pr_{s, D}(d\pi)$$

where D is a generic measurable policy on \mathcal{M} . (In the sequel, we assume that eT^{\min} is a function indexed by G .) Note that by definition of V_G , only the amount of time before entering the first G -state is relevant. Hence, we may turn all G -states into absorbing without affecting the expected time reachability. It is done via replacing all of their emanating transitions by a single Markovian self loop (a Markovian transition to the state itself) with an arbitrary rate. In the remainder of this section we assume all goal states to be absorbing. Let μ_α^s be the distribution such that $s \xrightarrow{\alpha} \mu_\alpha^s$. As we assume that all action labels of the transitions emanating a state are unique (by numbering them), this distribution is unique.

Theorem 3.1. *The function eT^{\min} is a fixpoint of the Bellman operator*

$$[L(v)](s) = \begin{cases} \frac{1}{E(s)} + \sum_{s' \in S} \mathbf{P}(s, s') \cdot v(s') & \text{if } s \in MS \setminus G \\ \min_{\alpha \in Act(s)} \sum_{s' \in S} \mu_\alpha^s(s') \cdot v(s') & \text{if } s \in PS \setminus G \\ 0 & \text{if } s \in G, \end{cases}$$

where $Act(s) = \{\tau_i \mid s \xrightarrow{\tau_i} \mu\}$ and $\mu_\alpha^s \in \text{Distr}(S)$ is as formerly defined.

We will later see that eT^{\min} is in fact the unique fixpoint of the Bellman operator. Let us explain the above result. For a goal state, the expected time obviously is zero. For a Markovian state $s \notin G$, the minimal expected time to reach some state in G is the expected sojourn time in s (which equals $\frac{1}{E(s)}$) plus the expected time to reach some state in G via one of its successor states. For a probabilistic state, an action is selected that minimises

the expected time according to the distribution μ_α^s corresponding to α in state s . The characterisation of $eT^{\min}(s, \diamond G)$ in Thm. 3.1 allows us to reduce the problem of computing the minimum expected time reachability in an MA to a non-negative SSP problem [5, 12]. This goes as follows.

Definition 3.2 (SSP for minimum expected time reachability). The SSP of MA $\mathcal{M} = (S, A, \rightarrow, \Longrightarrow, s_0)$ for the expected time reachability of $G \subseteq S$ is

$$\text{ssp}_{et}(\mathcal{M}) = (S, A \cup \{\perp\}, \mathbf{P}, s_0, G, c, g)$$

where $g(s) = 0$ for all $s \in G$ and

$$\mathbf{P}(s, \sigma, s') = \begin{cases} \frac{\mathbf{R}(s, s')}{E(s)} & \text{if } s \in MS, \sigma = \perp \\ \mu_\sigma^s(s') & \text{if } s \in PS, s \xrightarrow{\sigma} \mu_\sigma^s \\ 0 & \text{otherwise, and} \end{cases} \quad c(s, \sigma) = \begin{cases} \frac{1}{E(s)} & \text{if } s \in MS \setminus G, \sigma = \perp \\ 0 & \text{otherwise.} \end{cases}$$

Terminal costs are zero. Transition probabilities are defined in the standard way. The cost of a Markovian state is its expected sojourn time, whereas that of a probabilistic one is zero.

Theorem 3.3. *Given an MA \mathcal{M} , $eT^{\min}(s, \diamond G)$ equals $cR^{\min}(s, \diamond G)$ in $\text{ssp}_{et}(\mathcal{M})$.*

Thus there is a stationary deterministic policy on \mathcal{M} yielding $eT^{\min}(s, \diamond G)$. Moreover, the uniqueness of the minimum expected cost of an SSP [5, 12] now yields that $eT^{\min}(s, \diamond G)$ is the unique fixpoint of L (see Thm. 3.1). This follows from the fact that the Bellman operator defined in Thm 3.1 equals the Bellman operator for $cR^{\min}(s, \diamond G)$. The uniqueness result enables the usage of standard solution techniques such as value iteration and linear programming to compute $eT^{\min}(s, \diamond G)$. For maximum expected time objectives, a similar fixpoint theorem is obtained, and it can be proven that those objectives correspond to the maximal expected reward in the SSP problem defined above. Thus far, we have assumed MAs to be non-Zeno, i.e., they do not contain a reachable cycle solely consisting of probabilistic transitions. However, the above notions can all be extended to deal with such Zeno cycles, by, e.g., setting the minimal expected time of states in Zeno BSCCs that do not contain G -states to be infinite (as such states cannot reach G). Similarly, the maximal expected time of states in Zeno end components (that do not contain G -states) can be defined as infinity, as in the worst case these states will never reach G .

4. LONG-RUN OBJECTIVES

Let \mathcal{M} be an MA with state space S and $G \subseteq S$ a set of goal states. Let $\mathbf{1}_G$ be the characteristic function of G on finite sequences, i.e., $\mathbf{1}_G(\pi) = 1$ if and only if $s \in G$ for some s in π . Following the ideas of [11, 26], the fraction of time spent in G on an infinite path π in \mathcal{M} up to time bound $t \in \mathbb{R}_{\geq 0}$ is given by the random variable $A_{G,t}(\pi) = \frac{1}{t} \int_0^t \mathbf{1}_G(\pi @ u) du$. Taking the limit $t \rightarrow \infty$, we obtain the random variable

$$A_G(\pi) = \lim_{t \rightarrow \infty} A_{G,t}(\pi) = \lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t \mathbf{1}_G(\pi @ u) du.$$

The expectation of A_G for policy D and initial state s yields the corresponding long-run average time spent in G :

$$LRA^D(s, G) = \mathbb{E}_{s,D}(A_G) = \int_{\text{Paths}} A_G(\pi) \cdot \Pr_{s,D}(d\pi).$$

The minimum long-run average time spent in G starting from state s is then:

$$LRA^{\min}(s, G) = \inf_{D \in GM} LRA^D(s, G) = \inf_{D \in GM} \mathbb{E}_{s, D}(A_G).$$

Note that $\mathbf{1}_G(\pi@u) = 1$ if and only if $\pi@u$ is a sequence containing at least one state in G . For the long-run average analysis, we assume w.l.o.g. that $G \subseteq MS$, as the long-run average time spent in any probabilistic state is always 0. This claim follows directly from the fact that probabilistic states are instantaneous, i.e. their sojourn time is 0 by definition. Note that in contrast to the expected time analysis, G -states cannot be made absorbing in the long-run average analysis.

First we need to introduce *maximal end components*. A sub-MA of MA \mathcal{M} is a pair (S', K) where $S' \subseteq S$ and $K: S' \rightarrow 2^A$ is a function such that: (i) $K(s) \neq \emptyset$, (ii) $s \in S'$ and $\alpha \in K(s)$ and $s \xrightarrow{\alpha} \mu$ with $\mu(s') > 0$ implies $s' \in S'$, and (iii) $s \in S'$ and $s \xrightarrow{\lambda} s'$ implies $s' \in S'$. A sub-MA (S', K) is contained in a sub-MA (S'', K') if $S' \subseteq S''$ and $K(s) \subseteq K'(s)$ for all $s \in S'$. An *end component* is a sub-MA whose underlying graph is strongly connected; it is *maximal* w.r.t. K if it is not contained in any other end component (S'', K') of \mathcal{M} .

In the remainder of this section, we discuss in detail how to compute the minimum long-run average fraction of time spent in G in an MA \mathcal{M} with initial state s_0 . The general idea is the following three-step procedure:

- (1) Determine the maximal end components $\{\mathcal{M}_1, \dots, \mathcal{M}_k\}$ of MA \mathcal{M} .
- (2) Determine $LRA^{\min}(G)$ in maximal end component \mathcal{M}_j for all $j \in \{1, \dots, k\}$.
- (3) Reduce the computation of $LRA^{\min}(s_0, G)$ in MA \mathcal{M} to an SSP problem.

The first phase can be performed by a graph-based algorithm [10, 8], whereas the last two phases boil down to solving (distinct) LP problems.

4.1. Unichain MA. We first show that for unichain MAs computing $LRA^{\min}(s, G)$ can be reduced to determining long-run ratio objectives in MDPs. The notion of unichain is standard in MDPs [30] and is adopted to MAs in a straightforward manner. An MA is unichain if for any stationary deterministic policy the induced stochastic process consists of a single ergodic class plus a possibly non-empty set of transient states². Let us first explain the long-run ratio objectives. Let $M = (S, A, \mathbf{P}, s_0)$ be an MDP. Assume w.l.o.g. that for each $s \in S$ there exists $\alpha \in A$ such that $\mathbf{P}(s, \alpha, s') > 0$ for some $s' \in S$. Let $c_1, c_2: S \times A \rightarrow \mathbb{R}_{\geq 0}$ be cost functions. The operational interpretation is that a cost $c_1(s, \alpha)$ is incurred when selecting action α in state s , and similar for c_2 . Our interest is the *ratio* between c_1 and c_2 along a path. The *long-run ratio* \mathcal{R} between the accumulated costs c_1 and c_2 along the infinite path $\pi = s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \dots$ in the MDP M is defined by:

$$\mathcal{R}(\pi) = \lim_{n \rightarrow \infty} \frac{\sum_{i=0}^{n-1} c_1(s_i, \alpha_i)}{\sum_{j=0}^{n-1} c_2(s_j, \alpha_j)}.$$

The minimum long-run ratio objective for state s of MDP M is defined by:

$$R^{\min}(s) = \inf_D \mathbb{E}_{s, D}(\mathcal{R}) = \inf_D \sum_{\pi \in Paths} \mathcal{R}(\pi) \cdot \Pr_{s, D}(\pi).$$

²State s is *transient* if and only if the probability of the set of paths that start from s but never return back to it is positive, otherwise it is *recurrent*. An MA is *ergodic* if for all stationary deterministic policies the induced stochastic process consists of a single recurrent class.

Here, $Paths$ is the set of paths in the MDP, D is a stationary deterministic MDP-policy, and \Pr is the probability measure on MDP-paths. From [10, Th. 6.14], it follows that $R^{\min}(s)$ can be obtained by solving the following LP problem with real variables k and non-negative x_s for each $s \in S$: Maximise k subject to:

$$x_s \leq c_1(s, \alpha) - k \cdot c_2(s, \alpha) + \sum_{s' \in S} \mathbf{P}(s, \alpha, s') \cdot x_{s'} \quad \text{for each } s \in S, \alpha \in A.$$

We now transform an MA into an MDP with two cost functions as follows.

Definition 4.1 (From MA to 2-cost MDPs). Let $\mathcal{M} = (S, A, \rightarrow, \Longrightarrow, s_0)$ be an MA and $G \subseteq S$ a set of goal states. The MDP $\mathbf{mdp}(\mathcal{M}) = (S, A \cup \{\perp\}, \mathbf{P}, s_0)$, where \mathbf{P} is defined as in Def. 3.2, is extended with cost functions c_1 and c_2 defined by:

$$c_1(s, \sigma) = \begin{cases} \frac{1}{E(s)} & \text{if } s \in MS \cap G \wedge \sigma = \perp \\ 0 & \text{otherwise,} \end{cases} \quad c_2(s, \sigma) = \begin{cases} \frac{1}{E(s)} & \text{if } s \in MS \wedge \sigma = \perp \\ 0 & \text{otherwise.} \end{cases}$$

Observe that cost function c_2 keeps track of the average residence time in state s whereas c_1 only does so for states in G . Furthermore, \mathcal{R} is well-defined in this setting, since the cost functions c_1 and c_2 are obtained from non-Zeno MA. In other words, the probability of the set of paths with ill-defined long-run ratio is zero.

Theorem 4.2. *For unichain MA \mathcal{M} , $LRA^{\min}(s, G)$ equals $R^{\min}(s)$ in $\mathbf{mdp}(\mathcal{M})$.*

To summarise, computing the minimum long-run average fraction of time that is spent in some goal state in $G \subseteq S$ in a unichain MA \mathcal{M} equals the minimum long-run ratio objective in an MDP with two cost functions. The latter can be obtained by solving an LP problem. Observe that for any two states s, s' in a unichain MA, $LRA^{\min}(s, G)$ and $LRA^{\min}(s', G)$ coincide. We therefore omit the state and simply write $LRA^{\min}(G)$ when considering unichain MAs.

4.2. Arbitrary MA. Let \mathcal{M} be an MA with initial state s_0 and maximal end components $\{\mathcal{M}_1, \dots, \mathcal{M}_k\}$ for $k > 0$ where MA \mathcal{M}_j has state space S_j .

Lemma 4.3. Let \mathcal{M} be a maximal end component and D a stationary deterministic policy inducing a multichain on \mathcal{M} . Then there exists a stationary deterministic policy D' inducing a unichain on \mathcal{M} such that the long-run ratio is at least as good as for D .

Therefore, we can say that each \mathcal{M}_j induces a unichain MA for the optimal long-run ratio. Using this decomposition of \mathcal{M} into maximal end components, we obtain the following result:

Theorem 4.4. ³For MA $\mathcal{M} = (S, A, \rightarrow, \Longrightarrow, s_0)$ with MECs $\{\mathcal{M}_1, \dots, \mathcal{M}_k\}$ with state spaces $S_1, \dots, S_k \subseteq S$, and set of goal states $G \subseteq S$:

$$LRA^{\min}(s_0, G) = \inf_{D \in GM} \sum_{j=1}^k LRA_j^{\min}(G) \cdot \Pr_{s_0, D}(\diamond \square S_j),$$

where $\Pr_{s_0, D}(\diamond \square S_j)$ is the probability to eventually reach and continuously stay in some states in S_j from s_0 under policy D and $LRA_j^{\min}(G)$ is the LRA of $G \cap S_j$ in unichain MA \mathcal{M}_j .

³This theorem corrects a small flaw in the corresponding theorem for IMCs in [18].

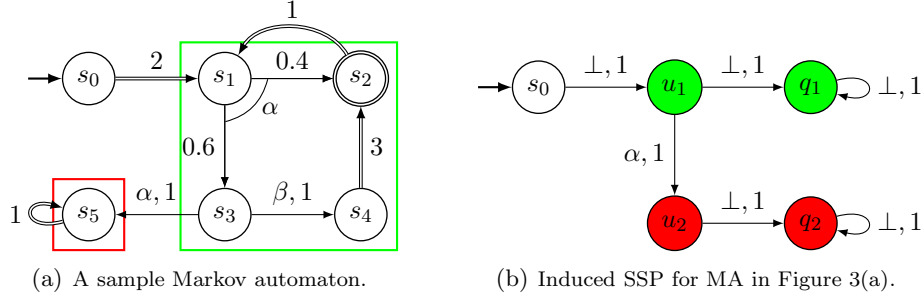


Figure 3: Example for Definition 4.5.

Computing the minimal LRA for arbitrary MAs is now reducible to a non-negative SSP problem. This proceeds as follows. In MA \mathcal{M} , we replace each maximal end component \mathcal{M}_j by two fresh states q_j and u_j . Intuitively, q_j represents \mathcal{M}_j whereas u_j can be seen as the gate to and from \mathcal{M}_j . Thus, state u_j has a Dirac transition to q_j as well as all probabilistic transitions leaving S_j . Let U denote the set of u_j states and Q the set of q_j states. For simplicity of the definition we assume w.l.o.g. that each probabilistic state induces a τ -transition with an index of the state. Further, the τ -transitions of each state $s_k \in PS$ are numbered from 1 to $n_{s_k} \in \mathbb{N}$, where n_{s_k} is the number of probability distributions induced by τ_{s_k} . Thus, we denote an action in state s_k with $\tau_{s_{kl}}$ with $l \in \{1 \dots n_{s_k}\}$.

Definition 4.5 (SSP for long-run average). The SSP of MA \mathcal{M} for the LRA in $G \subseteq S$ is $\mathbf{ssp}_{lra}(\mathcal{M}) = ((S \setminus \bigcup_{i=1}^k S_i) \cup U \cup Q, A \cup \{\perp\}, \mathbf{P}', s_0, Q, c, g)$, where $g(q_i) = LRA_i^{\min}(G)$ for $q_i \in Q$ and $c(s, \sigma) = 0$ for all s and $\sigma \in A \cup \{\perp\}$. \mathbf{P}' is defined as follows. Let $S' = S \setminus \bigcup_{i=1}^k S_i$. $\mathbf{P}'(s, \sigma, s')$ equals $\mathbf{P}(s, \sigma, s')$ for all $s, s' \in S'$ and $\sigma \in A \cup \{\perp\}$. For the new states u_j :

$$\begin{aligned} \mathbf{P}'(u_j, \tau_{s_{kl}}, s') &= \mathbf{P}(s_k, \tau_{s_{kl}}, s') \quad \text{if } s' \in S' \wedge s_k \in S_j \wedge l \in \{1 \dots n_{s_k}\} & \text{and} \\ \mathbf{P}'(u_i, \tau_{s_{kl}}, u_j) &= \mathbf{P}(s_k, \tau_{s_{kl}}, S_j) \quad \text{if } s_k \in S_i \wedge l \in \{1 \dots n_{s_k}\} \wedge \tau_{s_{ki}} \notin A_i \end{aligned}$$

Finally, we have: $\mathbf{P}'(q_j, \perp, q_j) = 1 = \mathbf{P}'(u_j, \perp, q_j)$ and $\mathbf{P}'(s, \sigma, u_j) = \mathbf{P}(s, \sigma, S_j)$.

Here, $\mathbf{P}(s, \alpha, S')$ is a shorthand for $\sum_{s' \in S'} \mathbf{P}(s, \alpha, s')$ and A_i denotes the action set of maximal end component \mathcal{M}_i . The terminal costs of the new q_i -states are set to $LRA_i^{\min}(G)$.

Example 4.6. Consider the MA \mathcal{M} from Figure 3(a), having MECs \mathcal{M}_1 with $S_1 = \{s_1, s_2, s_3, s_4\}$ and \mathcal{M}_2 with $S_2 = \{s_5\}$. For the simplification of the action notation, we use α and β instead of τ . Let $G = \{s_2\}$. By Definition 4.5, $\mathbf{ssp}_{lra}(\mathcal{M})$ is defined as follows. As $k=2$, $U = \{u_1, u_2\}$ and $Q = \{q_1, q_2\}$. Hence, $S_{\mathbf{ssp}} = \{s_0, u_1, u_2, q_1, q_2\}$. First consider $s, s' \in S'$. Since, $S' = \{s_0\}$ and there exists no transition from s_0 to s_0 we can omit the first rule. Now consider all outgoing transitions from MECs. For \mathcal{M}_1 there exists a transition from $s_3 \xrightarrow{\alpha, 1} s_5$ in the underlying MA, where $s_3 \in S_1$ and $s_5 \in S_2$. It follows that $\mathbf{P}'(u_1, \alpha, u_2) = \mathbf{P}(s_3, \alpha, S_2) = 1$. Now consider all states in U and Q and add new transitions with $\mathbf{P}(u_i, \perp, q_i) = \mathbf{P}(q_i, \perp, q_i) = 1$ for $i = 1, 2$. Finally, consider all states $s \in S_{\mathbf{ssp}} \cap S$ with a transition into a MEC. Hence, $\mathbf{P}'(s_0, \perp, u_1) = \mathbf{P}(s_0, \perp, s_1) = 1$. The MDP of $\mathbf{ssp}_{lra}(\mathcal{M})$ is depicted in Figure 3(b).

Theorem 4.7. For MA \mathcal{M} , $LRA^{\min}(s_0, G)$ equals $cR^{\min}(s_0, \diamond Q)$ in SSP $\mathbf{ssp}_{lra}(\mathcal{M})$.

To summarise, computing the minimum long-run average fraction of time that is spent in some goal states in $G \subseteq S$ in an arbitrary MA \mathcal{M} starting in state s_0 equals the minimum expected cost of an SSP.

5. TIMED REACHABILITY OBJECTIVES

This section presents an algorithm that approximates time-bounded reachability probabilities in MAs. We start with a fixpoint characterisation, and then explain how these probabilities can be approximated using a discretisation technique.

5.1. Fixpoint characterisation. Our goal is to come up with a fixpoint characterisation for the maximum (or minimum) probability to reach a set of goal states in a time interval. Let \mathcal{I} and \mathcal{Q} be the set of all nonempty nonnegative real intervals with real and rational bounds, respectively. For interval $I \in \mathcal{I}$ and $t \in \mathbb{R}_{\geq 0}$, let $I \ominus t = \{x - t \mid x \in I \wedge x \geq t\}$. Given MA \mathcal{M} , $I \in \mathcal{I}$ and a set $G \subseteq S$ of goal states, the set of all paths that reach some goal states within interval I is denoted by $\diamond^I G$. Let $p_{\max}^{\mathcal{M}}(s, \diamond^I G)$ be the maximum probability of reaching G within interval I if starting in state s at time 0. Here, the maximum is taken over all possible generic measurable policies. The next lemma provides a characterisation of $p_{\max}^{\mathcal{M}}(s, \diamond^I G)$ as a fixpoint.

Lemma 5.1. *Let \mathcal{M} be an MA, $G \subseteq S$ and $I \in \mathcal{I}$ with $\inf I = a$ and $\sup I = b$. Then, $p_{\max}^{\mathcal{M}}(s, \diamond^I G)$ is the least fixpoint of the higher-order operator $\Omega: (S \times \mathcal{I} \rightarrow [0, 1]) \rightarrow (S \times \mathcal{I} \rightarrow [0, 1])$, which for $s \in MS$ is given by:*

$$\Omega(F)(s, I) = \begin{cases} \int_0^b E(s) e^{-E(s)t} \sum_{s' \in S} \mathbf{P}(s, s') F(s', I \ominus t) dt & s \notin G \\ e^{-E(s)a} + \int_0^a E(s) e^{-E(s)t} \sum_{s' \in S} \mathbf{P}(s, s') F(s', I \ominus t) dt & s \in G \end{cases}$$

and for $s \in PS$ is defined by:

$$\Omega(F)(s, I) = \begin{cases} 1 & s \in G \wedge 0 \in I \\ \max_{\alpha \in Act(s)} \sum_{s' \in S} \mu_{\alpha}^s(s') F(s', I) & \text{otherwise.} \end{cases}$$

The proof of Lemma 5.1 is a slight adaptation of the proof of [17, Thm. 4], where it has been also shown that $p_{\max}^{\mathcal{M}}(s, \diamond^I G)$ is Lipschitz continuous and thus measurable. The characterisation is a simple generalisation of that for IMCs [38], reflecting the fact that taking an action from a probabilistic state leads to a distribution over the states (rather than a single state). The above characterisation yields a Volterra integral equation system which is in general not directly tractable [3]. To tackle this problem, we approximate the fixpoint characterisation using discretisation, extending ideas developed in [38].

5.2. Discretisation. We split the time interval into equally-sized discretisation steps, each of length δ . The discretisation step is assumed to be small enough such that with high probability it carries at most one Markovian transition. This allows us to construct a discretised MA (dMA), a variant of a semi-MDP, obtained by summarising the behaviour of the MA at equidistant time points. Paths in a dMA can be seen as time-abstract paths in the corresponding MA, implicitly counting discretisation steps, and thus discrete time.

Definition 5.2. Given MA $\mathcal{M} = (S, A, \rightarrow, \Longrightarrow, s_0)$ and discretisation step $\delta \in \mathbb{R}_{>0}$, $\mathcal{M}_\delta = (S, A, \rightarrow, \Longrightarrow_\delta, s_0)$ is the dMA induced from \mathcal{M} with respect to δ , with $\Longrightarrow_\delta = \{(s, \mu^s) \mid s \in MS\}$, where

$$\mu^s(s') = \begin{cases} (1 - e^{-E(s)\delta})\mathbf{P}(s, s') & \text{if } s' \neq s \\ (1 - e^{-E(s)\delta})\mathbf{P}(s, s') + e^{-E(s)\delta} & \text{otherwise.} \end{cases}$$

Using the above fixpoint characterisation, it is now possible to relate reachability probabilities in the MA \mathcal{M} to reachability probabilities in its dMA \mathcal{M}_δ .

Theorem 5.3. *Given MA $\mathcal{M} = (S, A, \rightarrow, \Longrightarrow, s_0)$, $G \subseteq S$, interval $I = [0, b] \in \mathcal{Q}$ with $b \geq 0$ and $\lambda = \max_{s \in MS} E(s)$. Let $\delta \in \mathbb{R}_{>0}$ be such that $b = k_b \delta$ for some $k_b \in \mathbb{N}$. Then, for all $s \in S$ it holds that*

$$p_{\max}^{\mathcal{M}_\delta}(s, \diamond^{[0, k_b]} G) \leq p_{\max}^{\mathcal{M}}(s, \diamond^{[0, b]} G) \leq p_{\max}^{\mathcal{M}_\delta}(s, \diamond^{[0, k_b]} G) + 1 - e^{-\lambda b} (1 + \lambda \delta)^{k_b}.$$

This theorem can be extended to intervals with non-zero lower bounds; for the sake of brevity, the details are omitted here. The remaining problem is to compute $p_{\max}^{\mathcal{M}_\delta}(s, \diamond^{[0, k_b]} G)$, which is the maximum probability to reach some goal state in dMA \mathcal{M}_δ within the step bound k_b from initial state s . Let $\diamond^{[0, k_b]} G$ be the set of infinite (time-abstract) paths of \mathcal{M}_δ that reach some state in G within k_b steps; the objective is then formalised by $p_{\max}^{\mathcal{M}_\delta}(s, \diamond^{[0, k_b]} G) = \sup_{D \in TA} \Pr_{s, D}(\diamond^{[0, k_b]} G)$ where we recall that TA denotes the set of time-abstract policies. Our algorithm is now an adaptation (to dMA) of the well-known value iteration scheme for MDPs.

The algorithm proceeds by backward unfolding of the dMA in an iterative manner, starting from the goal states. Each iteration intertwines the analysis of Markovian states and of probabilistic states. The key idea is that a path from probabilistic states to G is split into two parts: reaching Markovian states from probabilistic states in zero time and reaching goal states from Markovian states in interval $[0, j]$, where j is the step count of the iteration. The former computation can be reduced to an unbounded reachability problem in the MDP induced by probabilistic states with rewards on Markovian states. For the latter, the algorithm operates on the previously computed reachability probabilities from all Markovian states up to step count j . We can generalise this recipe from step-bounded reachability to step interval-bounded reachability; details are described in [21].

6. TOOL CHAIN AND CASE STUDIES

This section describes the implementation of the algorithms discussed, together with the modelling features resulting in our MAMA tool chain. Also, we present two case studies that provide empirical evidence of the strengths and weaknesses of the MAMA tool chain.

6.1. Modelling. As argued in the introduction, MAs can be used as a semantical model for various modelling formalisms. We use the process-algebraic specification language MAPA (Markov Automata Process Algebra) [35, 34]. This language contains the usual process algebra operators, can treat data as first-class citizens, and supports several reduction techniques for MA specifications. In fact, it turns out to be beneficial to map a language (like GSPNs) to MAPA so as to profit from these reductions.

The MAPA language supports algebraic processes featuring data, nondeterministic choice, action prefix with probabilistic choice, rate prefix, conditional behaviour and process instantiation (allowing recursion). Using MAPA processes as basic building blocks, the language also supports the modular construction of large systems via top-level parallelism, encapsulation, hiding and renaming. The operational semantics of a MAPA specification yields an MA; for a detailed exposition of the syntax and semantics we refer to [35, 34].

To enable state space reduction and generation, our tool chain uses a linearised normal form of MAPA referred to as MLPE (Markovian Linear Probabilistic process Equation). In this format, there is precisely one process which consists of a nondeterministic choice between a set of symbolic transitions, making MLPEs easy to translate to MAs. Every MAPA specification can be translated efficiently into an MLPE while preserving strong bisimulation [35].

Reduction techniques. On MLPEs, several reduction techniques have been defined. Some of them simplify the MLPE to improve readability and speed up state space generation, while others really modify it in such a way that the underlying MA gets smaller. Being defined on the specification, these reductions eliminate the need to ever generate the original unreduced state space. We briefly discuss six such techniques.

- *Maximal progress reduction* removes Markovian transitions from states also having τ -transitions (motivated by the maximal progress assumption).
- *Constant elimination* [25] replaces parameters that remain forever constant by their initial (and hence permanent) value.
- *Expression simplification* [25] evaluates functions for which all parameters are constants and applies basic laws from logic.
- *Summation elimination* [25] removes trivial nondeterministic choices often arising from synchronisations.
- *Dead-variable reduction* [37] detects parts of the specification in which the value of some variable is irrelevant: it will be overwritten before being used for all possible futures. When reaching such a part, the variable is reset to its initial value.
- *Confluence reduction* [36] detects spurious nondeterminism resulting from parallel composition. It denotes a subset of the probabilistic transitions of a MAPA specification as confluent, meaning that they can safely be given priority if enabled together with other transitions.

6.2. MAMA tool chain. Our tool chain consists of several tool components: SCOOP [33, 35], IMCA [18], and GEMMA [4], see Figure 4. The tool chain comprises about 8,000 LOC (without comments). SCOOP (written in Haskell) supports the generation of MAs from MAPA specifications by a translation into the MLPE format. It implements all the reduction techniques described above. The capabilities of the IMCA tool component (written in C++) have been lifted to expected time and long-run objectives for MAs, and extended with

timed reachability objectives. It also supports (untimed) reachability objectives which are not treated further here. A prototypical translator from GSPNs to MAs, in fact MAPA specifications, has been realised (the GEMMA component, written in Haskell). We connected the three components into a single tool chain, by making SCOOP export the (reduced) state space of an MLPE in the IMCA input language. Additionally, SCOOP has been extended to translate properties, based on the actions and parameters of a MAPA specification, to a set of goal states in the underlying MA. That way, in one easy process, systems and their properties can be modelled in MAPA, translated to an optimised MLPE by SCOOP, exported to the IMCA tool and then analysed.

6.3. Case studies. This section presents experiments with MAMA. All experiments were conducted on a 2.5 GHz Intel Core i5 processor with 4GB RAM, running Mac OS X 10.8.3.

Processor grid. First, we consider a model of a 2×2 concurrent processor architecture. Using GEMMA [4], we automatically derived the MA model from the GSPN model in [1, Fig. 11.7]. Previous analysis of this model required weights for all immediate transitions, which necessitates having complete knowledge of the mutual behaviour of all these transitions. We allow a weight assignment to just a (possibly empty) subset of the immediate transitions—reflecting the practical scenario of only knowing the mutual behaviour for a selection of the transitions. For this case study we indeed kept weights for only a few of the transitions, obtaining probabilistic behaviour for them and nondeterministic behaviour for the others.

Table 1 reports on the time-bounded and time-interval bounded probabilities for reaching a state such that the first processor has an empty task queue. We vary the degree of multitasking K , the error bound ϵ and the interval I . For each setting, we report the number of states $|S|$ and goal states $|G|$, and the generation time with SCOOP (both with and without the reductions from Section 6.1).

The runtime demands grow with both the upper and lower time bound, as well as with the required accuracy. The model size also affects the per-iteration cost and thus the overall complexity of reachability computation. Note that the reductions speed-up the analysis times by a factor between 1.8 and 2.5: even more than the reduction in state space size. This is due to the fact that these techniques significantly reduce the degree of nondeterminism.

Table 2 displays the results for expected time until an empty task queue, as well as the long-run average that a processor is active. In contrast to [1], which fixes all nondeterminism and obtains, for instance, an LRA of 0.903 for $K = 2$, we are now able to retain nondeterminism and provide the more informative interval $[0.8810, 0.9953]$. Again, SCOOP’s reduction techniques significantly improve runtimes.

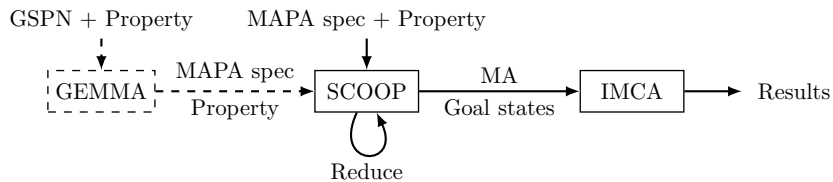


Figure 4: Analysing Markov automata using the MAMA tool chain.

K	unreduced			reduced			ϵ	I	$p^{\min}(s_0, \diamond G)$			$p^{\max}(s_0, \diamond G)$				
	$ S $	$ G $	time	$ S $	$ G $	time			time(unred)	time(red)	time(unred)	time(red)	time(unred)	time(red)		
2	2,508	1,398	0.6	1,789	1,122	0.8	10^{-2}	[0, 3]	0.91	58.5	31.0	0.95	54.9	21.7		
									10^{-2}	[0, 4]	0.96	103.0	54.7	0.98	97.3	38.8
									10^{-2}	[1, 4]	0.91	117.3	64.4	0.96	109.9	49.0
									10^{-3}	[0, 3]	0.910	580.1	309.4	0.950	544.3	218.4
3	10,852	4,504	3.1	7,201	3,613	3.5	10^{-2}	[0, 3]	0.18	361.5	202.8	0.23	382.8	161.1		
									10^{-2}	[0, 4]	0.23	643.1	360.0	0.30	681.4	286.0
									10^{-2}	[1, 4]	0.18	666.6	377.3	0.25	696.4	317.7
									10^{-3}	[0, 3]	0.176	3,619.5	2,032.1	0.231	3,837.3	1,611.9
4	31,832	10,424	9.8	20,021	8,357	10.5	10^{-2}	[0, 3]	0.01	1,156.8	614.9	0.03	1,196.5	486.4		

Table 1: Interval reachability probabilities for the grid. (Time in seconds.)

K	$eT^{\min}(s_0, \diamond G)$			$eT^{\max}(s_0, \diamond G)$			$LRA^{\min}(s_0, G)$			$LRA^{\max}(s_0, G)$		
	time(unred)	time(red)	time(red)	time(unred)	time(red)	time(red)	time(unred)	time(red)	time(unred)	time(red)	time(red)	
2	1.0000	0.3	0.1	1.2330	0.7	0.3	0.8110	1.3	0.7	0.9953	0.5	0.2
3	11.1168	18.3	7.7	15.2768	135.4	40.6	0.8173	36.1	16.1	0.9998	4.7	2.6
4	102.1921	527.1	209.9	287.8616	6,695.2	1,869.7	0.8181	505.1	222.3	1.0000	57.0	34.5

Table 2: Expected times and long-run averages for the grid. (Time in seconds.)

Q	N	unreduced			reduced			ϵ	I	$p^{\min}(s_0, \diamond G)$			$p^{\max}(s_0, \diamond G)$		
		$ S $	$ G $	time	$ S $	$ G $	time			time(unred)	time(red)	time(unred)	time(red)	time(unred)	time(red)
2	3	1,497	567	0.4	990	324	0.2	10^{-3}	[0, 1]	0.277	4.7	2.9	0.558	4.6	2.5
										10^{-3}	[1, 2]	0.486	22.1	14.9	0.917
2	4	4,811	2,304	1.0	3,047	1,280	0.6	10^{-3}	[0, 1]	0.201	25.1	14.4	0.558	24.0	13.5
										10^{-3}	[1, 2]	0.344	106.1	65.8	0.917
3	3	14,322	5,103	3.0	9,522	2,916	1.7	10^{-3}	[0, 1]	0.090	66.2	40.4	0.291	60.0	38.5
										10^{-3}	[1, 2]	0.249	248.1	180.9	0.811
3	4	79,307	36,864	51.6	50,407	20,480	19.1	10^{-3}	[0, 1]	0.054	541.6	303.6	0.291	578.2	311.0
										10^{-3}	[1, 2]	0.141	2,289.3	1,305.0	0.811
4	2	6,667	1,280	1.1	4,745	768	0.8	10^{-3}	[0, 1]	0.049	19.6	14.0	0.118	19.7	12.8
										10^{-3}	[1, 2]	0.240	83.2	58.7	0.651
4	3	131,529	45,927	85.2	87,606	26,244	30.8	10^{-3}	[0, 1]	0.025	835.3	479.0	0.118	800.7	466.1
										10^{-3}	[1, 2]	0.114	3,535.5	2,062.3	0.651

Table 3: Interval reachability probabilities for the polling system. (Time in seconds.)

Polling system. Second, we consider a polling system with two stations and one server, similar to the one depicted in Figure 2 and inspired by [32]. There are incoming requests of N possible types, each of them with a (possibly different) service rate. Additionally, the stations each store these in a local queue of size Q . We vary the values of Q and N , analysing a total of six different settings. Since—as for the previous case—analysis scales proportionally with the error bound, we keep this constant here.

Table 3 reports results for time-bounded and time-interval bounded properties, and Table 4 displays probabilities and runtime results for expected times and long-run averages. For all analyses, the goal set consists of all states for which both station queues are full.

Q	N	$eT^{\min}(s_0, \diamond G)$			$eT^{\max}(s_0, \diamond G)$			$LRA^{\min}(s_0, G)$			$LRA^{\max}(s_0, G)$		
		time(timed)	time(red)	time(timed)	time(red)	time(timed)	time(red)	time(timed)	time(red)	time(timed)	time(red)	time(timed)	time(red)
2	3	1.0478	0.2	0.1	2.2489	0.3	0.2	0.1230	0.8	0.5	0.6596	0.2	0.1
2	4	1.0478	0.2	0.1	3.2053	2.0	1.0	0.0635	9.0	5.2	0.6596	1.3	0.6
3	3	1.4425	1.0	0.6	4.6685	8.4	5.0	0.0689	177.9	123.6	0.6600	26.2	13.0
3	4	1.4425	9.7	4.6	8.0294	117.4	67.2	0.0277	7,696.7	5,959.5	0.6600	1,537.2	862.4
4	2	1.8226	0.4	0.3	4.6032	2.4	1.6	0.1312	45.6	32.5	0.6601	5.6	3.9
4	3	1.8226	29.8	14.2	9.0300	232.8	130.8	– timeout (18 hours) –			0.6601	5,339.8	3,099.0

Table 4: Expected times and long-run averages for the polling system. (Time in seconds.)

7. CONCLUSION

This paper presented new algorithms for the quantitative analysis of Markov automata (MAs) and proved their correctness. Three objectives have been considered: expected time, long-run average, and timed reachability. The MAMA tool chain supports the modelling and reduction of MAs, and can analyse these three objectives. It is also equipped with a prototypical tool to map GSPNs onto MAs. The MAMA tool is accessible via its easy-to-use web interface that can be found at <http://fmt.cs.utwente.nl/~timmer/mama>. Experimental results on a processor grid and a polling system give insight into the accuracy and scalability of the presented algorithms. Future work will focus on efficiency improvements and reward extensions [20].

ACKNOWLEDGEMENTS

This work is funded by the EU FP7-projects SENSATION and MEALS, the STW project ArRangeer (grant 12238), the NWO project SYRUP (grant 612.063.817), and the DFG Sonderforschungsbereich AVACS.

REFERENCES

- [1] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets*. John Wiley & Sons, 1995.
- [2] M. Ajmone Marsan, G. Conte, and G. Balbo. A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems. *ACM Transactions on Computer Systems*, 2(2):93–122, 1984.
- [3] C. Baier, B. R. Haverkort, H. Hermanns, and J.-P. Katoen. Model-checking algorithms for continuous-time Markov chains. *IEEE Transactions on Software Engineering*, 29(6):524–541, 2003.
- [4] R. Bamberg. Non-deterministic generalised stochastic Petri nets modelling and analysis. Master’s thesis, University of Twente, 2012.
- [5] D. P. Bertsekas and J. N. Tsitsiklis. An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, 16(3):580–595, 1991.
- [6] H. Boudali, P. Crouzen, and M. I. A. Stoelinga. A rigorous, compositional, and extensible framework for dynamic fault tree analysis. *IEEE Transactions on Dependable and Secure Computing*, 7(2):128–143, 2010.
- [7] 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.
- [8] K. Chatterjee and M. Henzinger. Faster and dynamic algorithms for maximal end-component decomposition and related graph problems in probabilistic verification. In *SODA*, pages 1318–1336. SIAM, 2011.

- [9] N. Coste, H. Hermanns, E. Lantreibecq, and W. Serwe. Towards performance prediction of compositional models in industrial GALS designs. In *CAV*, volume 5643 of *LNCS*, pages 204–218. Springer, 2009.
- [10] L. de Alfaro. *Formal Verification of Probabilistic Systems*. PhD thesis, Stanford University, 1997.
- [11] L. de Alfaro. How to specify and verify the long-run average behavior of probabilistic systems. In *LICS*, pages 454–465. IEEE, 1998.
- [12] L. de Alfaro. Computing minimum and maximum reachability times in probabilistic systems. In *CONCUR*, volume 1664 of *LNCS*, pages 66–81. Springer, 1999.
- [13] Y. Deng and M. Hennessy. On the semantics of Markov automata. *Information and Computation*, 222:139–168, 2013.
- [14] C. Eisentraut, H. Hermanns, J.-P. Katoen, and L. Zhang. A semantics for every GSPN. In *Petri Nets*, volume 7927 of *LNCS*, pages 90–109. Springer, 2013.
- [15] C. Eisentraut, H. Hermanns, and L. Zhang. Concurrency and composition in a stochastic world. In *CONCUR*, volume 6269 of *LNCS*, pages 21–39. Springer, 2010.
- [16] C. Eisentraut, H. Hermanns, and L. Zhang. On probabilistic automata in continuous time. In *LICS*, pages 342–351. IEEE, 2010.
- [17] H. Fu. Maximal cost-bounded reachability probability on continuous-time Markov decision processes. *CoRR*, abs/1310.2514, 2013.
- [18] D. Guck, T. Han, J.-P. Katoen, and M. R. Neuhäüßer. Quantitative timed analysis of interactive Markov chains. In *NFM*, volume 7226 of *LNCS*, pages 8–23. Springer, 2012.
- [19] D. Guck, H. Hatefi, H. Hermanns, J.-P. Katoen, and M. Timmer. Modelling, reduction and analysis of Markov automata. In *QEST*, volume 8054 of *LNCS*, pages 55–71. Springer, 2013.
- [20] D. Guck, M. Timmer, H. Hatefi, E. J. J. Ruijters, and M. I. A. Stoelinga. Modelling and analysis of Markov reward automata. In *ATVA*, to appear in *LNCS*. Springer, 2014.
- [21] H. Hatefi and H. Hermanns. Model checking algorithms for Markov automata. In *ECEASST (AVoCS proceedings)*, volume 53, 2012.
- [22] 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 *DSN*, pages 419–424. IEEE, 2010.
- [23] H. Hermanns. *Interactive Markov Chains: The Quest for Quantified Quality*, volume 2428 of *LNCS*. Springer, 2002.
- [24] J.-P. Katoen. GSPNs revisited: Simple semantics and new analysis algorithms. In *ACSD*, pages 6–11. IEEE, 2012.
- [25] J.-P. Katoen, J. C. van de Pol, M. I. A. Stoelinga, and M. Timmer. A linear process-algebraic format with data for probabilistic automata. *Theoretical Computer Science*, 413(1):36–57, 2012.
- [26] G. López, H. Hermanns, and J.-P. Katoen. Beyond memoryless distributions: Model checking semi-Markov chains. In *PAPM-PROBMIV*, number 2165 in *LNCS*, pages 57–70. Springer, 2001.
- [27] J. F. Meyer, A. Movaghar, and W. H. Sanders. Stochastic activity networks: Structure, behavior, and application. In *PNPM*, pages 106–115. IEEE, 1985.
- [28] M. R. Neuhäüßer, M. I. A. Stoelinga, and J.-P. Katoen. Delayed nondeterminism in continuous-time Markov decision processes. In *FOSSACS*, volume 5504 of *LNCS*, pages 364–379. Springer, 2009.
- [29] J. Norris. *Markov Chains*. Cambridge University Press, 1997.
- [30] M. L. Puterman. *Markov Decision Processes*. Wiley, 1994.
- [31] R. Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems*. PhD thesis, Massachusetts Institute of Technology, 1995.
- [32] M. M. Srinivasan. Nondeterministic polling systems. *Management Science*, 37(6):667–681, 1991.
- [33] M. Timmer. SCOOP: A tool for symbolic optimisations of probabilistic processes. In *QEST*, pages 149–150. IEEE, 2011.
- [34] M. Timmer. *Efficient Modelling, Generation and Analysis of Markov Automata*. PhD thesis, University of Twente, 2013.
- [35] M. Timmer, J.-P. Katoen, J. C. van de Pol, and M. I. A. Stoelinga. Efficient modelling and generation of Markov automata. In *CONCUR*, volume 7454 of *LNCS*, pages 364–379. Springer, 2012.
- [36] M. Timmer, M. I. A. Stoelinga, and J. C. van de Pol. Confluence reduction for Markov automata. In *FORMATS*, volume 8053 of *LNCS*, pages 243–257, 2013.
- [37] J. C. van de Pol and M. Timmer. State space reduction of linear processes using control flow reconstruction. In *ATVA*, volume 5799 of *LNCS*, pages 54–68. Springer, 2009.

- [38] L. Zhang and M. R. Neuhäüßer. Model checking interactive Markov chains. In *TACAS*, volume 6015 of *LNCS*, pages 53–68. Springer, 2010.

APPENDIX A. PROOF OF THEOREM 3.1

Recall that the minimal expected time to reach G from $s \in S$ is defined by

$$eT^{\min}(s, \diamond G) = \inf_{D \in GM} \mathbb{E}_{s,D}(V_G) = \inf_{D \in GM} \int_{Paths} V_G(\pi) \Pr_{s,D}(d\pi)$$

where D is a generic measurable policy on \mathcal{M} . eT^{\min} is a function indexed by G . Further, $V_G: Paths \rightarrow \mathbb{R}_{\geq 0}^{\infty}$ is the elapsed time before visiting some state in G for the first time, i.e., $V_G(\pi) = \min \{t \in \mathbb{R}_{\geq 0} \mid G \cap \pi@t \neq \emptyset\}$ where $\min(\emptyset) = \infty$. Let $\Delta(\pi, k) = \sum_{i=0}^{k-1} t_i$ be the elapsed time on infinite path $\pi = s_0 \xrightarrow{\sigma_0, \mu_0, t_0} s_1 \xrightarrow{\sigma_1, \mu_1, t_1} \dots$ after k steps. V_G can be therefore rewritten as

$$V_G(\pi) = \begin{cases} \Delta(\pi, j) & \text{if } \pi[j] \in G \wedge \forall i < j. \pi[i] \notin G \\ \Delta(\pi, \infty) & \text{if } \forall i. \pi[i] \notin G \end{cases} \quad (\text{A.1})$$

Theorem 3.1. *The function eT^{\min} is a fixpoint of the Bellman operator*

$$[L(v)](s) = \begin{cases} \frac{1}{E(s)} + \sum_{s' \in S} \mathbf{P}(s, s') \cdot v(s') & \text{if } s \in MS \setminus G \\ \min_{\alpha \in Act(s)} \sum_{s' \in S} \mu_{\alpha}^s(s') \cdot v(s') & \text{if } s \in PS \setminus G \\ 0 & \text{if } s \in G, \end{cases}$$

where $Act(s) = \{\tau_i \mid s \xrightarrow{\tau_i} \mu\}$ and $\mu_{\alpha}^s \in \text{Distr}(S)$ is as formerly defined.

Proof. We show that $L(eT^{\min}(s, \diamond G)) = eT^{\min}(s, \diamond G)$, for all $s \in S$. Therefore, we will distinguish three cases: $s \in MS \setminus G$, $s \in PS \setminus G$, and $s \in G$. Note that $D \in GM$.

(i) if $s \in MS \setminus G$, we derive

$$\begin{aligned}
eT^{\min}(s, \diamond G) &= \inf_D \mathbb{E}_{s,D}(V_G) = \inf_D \int_{Paths} V_G(\pi) \cdot \Pr_{s,D}(d\pi) = \inf_D \int_{Paths} \Delta(\pi, k) \cdot \Pr_{s,D}(d\pi) \\
&= \inf_D \int_{Paths} \sum_{i=0}^{k-1} t_i \cdot \Pr_{s,D}(d\pi) = \inf_D \int_{Paths} (t_0 + \sum_{i=1}^{k-1} t_i) \cdot \Pr_{s,D}(d\pi) \\
&= \inf_D \int_0^\infty t \cdot E(s) e^{-E(s)t} + \sum_{s' \in S} \mathbf{P}(s, s') \cdot \mathbb{E}_{s', D[s \xrightarrow{\perp, \mathbf{P}(s, \cdot), t} s']}(V_G) dt \\
&= \inf_D \left(\int_0^\infty t \cdot E(s) e^{-E(s)t} dt + \int_0^\infty \sum_{s' \in S} \mathbf{P}(s, s') \cdot \mathbb{E}_{s', D[s \xrightarrow{\perp, \mathbf{P}(s, \cdot), t} s']}(V_G) dt \right) \\
&= \inf_D \left(\frac{1}{E(s)} + \sum_{s' \in S} \mathbf{P}(s, s') \cdot \int_0^\infty \mathbb{E}_{s', D[s \xrightarrow{\perp, \mathbf{P}(s, \cdot), t} s']}(V_G) dt \right) \\
&= \frac{1}{E(s)} + \inf_D \sum_{s' \in S} \mathbf{P}(s, s') \cdot \int_0^\infty \mathbb{E}_{s', D[s \xrightarrow{\perp, \mathbf{P}(s, \cdot), t} s']}(V_G) dt \\
&= \frac{1}{E(s)} + \inf_D \sum_{s' \in S} \mathbf{P}(s, s') \cdot \mathbb{E}_{s', D}(V_G) \\
&= \frac{1}{E(s)} + \sum_{s' \in S} \mathbf{P}(s, s') \cdot \inf_D \mathbb{E}_{s', D}(V_G) \\
&= \frac{1}{E(s)} + \sum_{s' \in S} \mathbf{P}(s, s') \cdot eT^{\min}(s', \diamond G) \\
&= L(eT^{\min}(s, \diamond G)).
\end{aligned}$$

where $D[s \xrightarrow{\perp, \mathbf{P}(s, \cdot), t} s']$ is the policy that resolves nondeterminism for path π' starting from s' as D does it for $s \xrightarrow{\perp, \mathbf{P}(s, \cdot), t} \pi'$, i.e. $D(s \xrightarrow{\perp, \mathbf{P}(s, \cdot), t} \pi') = D[s \xrightarrow{\perp, \mathbf{P}(s, \cdot), t} s'](\pi')$.

(ii) if $s \in PS \setminus G$, we derive

$$\begin{aligned}
eT^{\min}(s, \diamond G) &= \inf_D \mathbb{E}_{s,D}(V_G) = \inf_D \int_{Paths} V_G(\pi) \Pr_{s,D}(d\pi) \\
&= \inf_D \sum_{s \xrightarrow{\alpha, \mu, 0} s'} D(s)(\alpha) \cdot \mathbb{E}_{s', D[s \xrightarrow{\alpha, \mu, 0} s']}(V_G).
\end{aligned}$$

where $D[s \xrightarrow{\alpha, \mu, 0} s']$ is the policy that resolves nondeterminism for path π' starting from s' as D does it for $s \xrightarrow{\alpha, \mu, 0} \pi'$, i.e. $D(s \xrightarrow{\alpha, \mu, 0} \pi') = D[s \xrightarrow{\alpha, \mu, 0} s'](\pi')$. Each action $\alpha \in Act(s)$ uniquely determines a distribution μ_α^s , such that the successor state s' , with $s \xrightarrow{\alpha, \mu_\alpha^s, 0} s'$, satisfies $\mu_\alpha^s(s') > 0$:

$$\alpha^\star = \arg \min \left\{ \inf_D \sum_{s' \in S} \mu_\alpha^s(s') \cdot \mathbb{E}_{s', D}(V_G) \mid \alpha \in Act(s) \right\}$$

Hence, all optimal policies choose α^* with probability 1, i.e. $D(s)(\alpha^*) = 1$ and $D(s)(\beta) = 0$ for all $\beta \neq \alpha^*$. Thus, we obtain

$$\begin{aligned}
eT^{\min}(s, \diamond G) &= \inf_D \min_{s \xrightarrow{\alpha} \mu_\alpha^s} \sum_{s' \in S} \mu_\alpha^s(s') \cdot \mathbb{E}_{s', D[s \xrightarrow{\alpha, \mu_\alpha^s, 0} s']} (V_G) \\
&= \min_{s \xrightarrow{\alpha} \mu_\alpha^s} \inf_D \sum_{s' \in S} \mu_\alpha^s(s') \cdot \mathbb{E}_{s', D[s \xrightarrow{\alpha, \mu_\alpha^s, 0} s']} (V_G) \\
&= \min_{s \xrightarrow{\alpha} \mu_\alpha^s} \inf_D \sum_{s' \in S} \mu_\alpha^s(s') \cdot \mathbb{E}_{s', D} (V_G) \\
&= \min_{s \xrightarrow{\alpha} \mu_\alpha^s} \sum_{s' \in S} \mu_\alpha^s(s') \cdot eT^{\min}(s', \diamond G) \\
&= \min_{\alpha \in \text{Act}(s)} \sum_{s' \in S} \mu_\alpha^s(s') \cdot eT^{\min}(s', \diamond G) \\
&= L(eT^{\min}(s, \diamond G)).
\end{aligned}$$

(iii) if $s \in G$, we derive

$$eT^{\min}(s, \diamond G) = \inf_D \int_{\text{Paths}} V_G(\pi) \Pr_{s, D}(d\pi) = 0 = L(eT^{\min}(s, \diamond G)). \quad \square$$

APPENDIX B. PROOF OF THEOREM 3.3

Theorem 3.3. *Given an MA \mathcal{M} , $eT^{\min}(s, \diamond G)$ equals $cR^{\min}(s, \diamond G)$ in $\mathbf{ssp}_{et}(\mathcal{M})$.*

Proof. As shown in [5, 10], $cR^{\min}(s, \diamond G)$ is the unique fixpoint of the Bellman operator L' defined as

$$[L'(v)](s) = \min_{\alpha \in \text{Act}(s)} c(s, \alpha) + \sum_{s' \in S \setminus G} \mathbf{P}(s, \alpha, s') \cdot v(s') + \sum_{s' \in G} \mathbf{P}(s, \alpha, s') \cdot g(s').$$

We show that the Bellman operator L for \mathcal{M} defined in Theorem 3.1 equals L' for $\mathbf{ssp}_{et}(\mathcal{M})$. Note that by definition $g(s) = 0$ for all $s \in G$. Thus

$$[L'(v)](s) = \min_{\alpha \in \text{Act}(s)} c(s, \alpha) + \sum_{s' \in S \setminus G} \mathbf{P}(s, \alpha, s') \cdot v(s').$$

We distinguish three cases, $s \in MS \setminus G$, $s \in PS \setminus G$, and $s \in G$.

(i) If $s \in MS \setminus G$, then $\text{Act}(s) = \{\perp\}$ and therefore $\min_{\alpha \in \text{Act}(s)} c(s, \alpha) = c(s, \perp)$. Further $c(s, \perp) = \frac{1}{E(s)}$ and for all $s' \in S$, $\mathbf{P}(s, \perp, s') = \frac{\mathbf{R}(s, s')}{E(s)}$. Thus

$$[L'(v)](s) = \frac{1}{E(s)} + \sum_{s' \in S} \frac{\mathbf{R}(s, s')}{E(s)} \cdot v(s') = [L(v)](s).$$

(ii) If $s \in PS \setminus G$, for each action $\alpha \in \text{Act}(s)$ and successor state s' , with $\mathbf{P}(s, \alpha, s') > 0$ it follows that $\mathbf{P}(s, \alpha, s') = \mu_\alpha^s(s')$. Further, $c(s, \alpha) = 0$ for all $\alpha \in \text{Act}$. Thus

$$[L'(v)](s) = \min_{\alpha \in \text{Act}(s)} \sum_{s' \in S} \mathbf{P}(s, \alpha, s') \cdot v(s') = \min_{\alpha \in \text{Act}(s)} \sum_{s' \in S} \mu_\alpha^s(s') \cdot v(s') = [L(v)](s).$$

- (iii) If $s \in G$, then by definition $|Act(s)| = 1$ with $Act(s) = \{\perp\}$ and $\mathbf{P}(s, \perp, s) = 1$ and $c(s, \perp) = 0$. Thus

$$[L'(v)](s) = \sum_{s' \in S} \mathbf{P}(s, \alpha, s') \cdot v(s') = 0 = [L(v)](s). \quad \square$$

APPENDIX C. PROOF OF THEOREM 4.2

First we recall the definition of weak bisimulation for MAs [16]. Therefore, we have to introduce some additional notation. A *sub-distribution* μ over a set S is a function $\mu: S \rightarrow [0, 1]$ with $\sum_{s \in S} \mu(s) \leq 1$. We define $\text{supp}(\mu) = \{s \in S \mid \mu(s) > 0\}$ as the *support* of μ and the probability of $S' \subseteq S$ with respect to μ as $\mu(S') = \sum_{s \in S'} \mu(s)$. Let $|\mu| := \mu(S)$ denote the size of the sub-distribution μ . If $|\mu| = 1$ then μ is a full distribution. Let $\text{Distr}(S)$ and $\text{Subdistr}(S)$ denote the set of distributions and sub-distributions over S , respectively. We write $\mathbb{1}_s$ for the *Dirac distribution* for s , determined by $\mathbb{1}_s(s) = 1$. Let μ and μ' be two sub-distributions, then $\mu'' := \mu \oplus \mu'$ is defined by $\mu''(s) = \mu(s) + \mu'(s)$, if $|\mu''| \leq 1$. Further, μ'' can be split back into μ and μ' , where (μ, μ') is defined as the splitting of μ'' .

Next we introduce the tree notation for weak transitions. For $\sigma, \sigma' \in \mathbb{N}_{>0}^*$, let $\sigma \leq \sigma'$ if there exists a (possibly empty) $\Phi \in \mathbb{N}_{>0}^*$ such that $\sigma\Phi = \sigma'$. Moreover, let $\sigma < \sigma'$ if $\sigma \leq \sigma'$ and $\sigma \neq \sigma'$. A partial function $\mathcal{T}: \mathbb{N}_{>0}^* \rightarrow L$, which satisfies

- if $\sigma \leq \sigma'$ and $\sigma' \in \text{dom}(\mathcal{T})$ then $\sigma \in \text{dom}(\mathcal{T})$
- if $\sigma i \in \text{dom}(\mathcal{T})$ for $i > 1$, then also $\sigma(i-1) \in \text{dom}(\mathcal{T})$
- $\epsilon \in \text{dom}(\mathcal{T})$

is called an (infinite) *L-labelled tree*. The root of the tree \mathcal{T} is called ϵ and $\sigma \in \text{dom}(\mathcal{T})$ is a node of \mathcal{T} . A node σ is called a leaf of \mathcal{T} if there is no $\sigma' \in \text{dom}(\mathcal{T})$ such that $\sigma < \sigma'$. We denote the set of all leaves of \mathcal{T} by $\text{Leaf}_{\mathcal{T}}$ and the set of all inner nodes of \mathcal{T} by $\text{Inner}_{\mathcal{T}}$. Let $L = S \times \mathbb{R}_{\geq 0}$. A node in an *L-labelled tree* \mathcal{T} is labelled by a state and the probability of reaching this node from the root node of the tree. For a node σ we write $\text{Sta}_{\mathcal{T}}(\sigma)$ for the first component of $\mathcal{T}(\sigma)$ and $\text{Prob}_{\mathcal{T}}(\sigma)$ for the second component of $\mathcal{T}(\sigma)$.

Definition C.1 (Weak transition tree). Let $\mathcal{M} = (S, Act, \rightarrow, \Longrightarrow, s_0)$ be an MA. A weak transition tree \mathcal{T} is a $S \times \mathbb{R}_{\geq 0}$ -labelled tree that satisfies the following condition

- (i) $\text{Prob}_{\mathcal{T}}(\epsilon) = 1$,
- (ii) $\forall \sigma \in \text{Inner}_{\mathcal{T}} \setminus \text{Leaf}_{\mathcal{T}} : \exists \mu : \text{Sta}_{\mathcal{T}}(\sigma) \rightarrow \mu$ and
 $\text{Prob}_{\mathcal{T}}(\sigma) \cdot \mu = \llbracket (\text{Sta}_{\mathcal{T}}(\sigma'), \text{Prob}_{\mathcal{T}}(\sigma')) \mid \sigma' \in \text{Children}_{\mathcal{T}}(\sigma) \rrbracket$
- (iii) $\sum_{\sigma \in \text{Leaf}_{\mathcal{T}}} \text{Prob}(\sigma) = 1$.

A weak transition tree \mathcal{T} corresponds to a probabilistic execution fragment. It starts from $\text{Sta}_{\mathcal{T}}(\epsilon)$, and resolves the nondeterministic choices at every inner node of the tree, which represents the state in the MA it is labelled with. $\text{Prob}_{\mathcal{T}}(\sigma)$ is the probability of reaching a state $\text{Sta}_{\mathcal{T}}(\sigma)$ via immediate transitions in the MA, starting from state $\text{Sta}_{\mathcal{T}}(\epsilon)$. The distribution associated with \mathcal{T} , denoted $\mu_{\mathcal{T}}$, is defined as

$$\mu_{\mathcal{T}} \stackrel{\text{def}}{=} \bigoplus_{\sigma \in \text{Leaf}_{\mathcal{T}}} \llbracket (\text{Sta}_{\mathcal{T}}(\sigma), \text{Prob}_{\mathcal{T}}(\sigma)) \rrbracket.$$

Now we can define a *weak transition*: For $s \in S$ and $\mu \in \text{Distr}(S)$, let $s \rightsquigarrow \mu$ if μ is induced by some internal weak transition tree \mathcal{T} with $\text{Sta}_{\mathcal{T}}(\epsilon) = s$. Let $\mu \in \text{Distr}(S)$. If for every state $s_i \in \text{supp}(\mu)$, $s_i \rightsquigarrow \mu'_i$ for some μ'_i , then we write $\mu \rightsquigarrow \bigoplus_{s_i \in \text{supp}(\mu)} \mu(s_i) \mu'_i$.

Now a convex combination of weak transitions can be defined. Let $\mu \rightsquigarrow_C \gamma$ if there exists a finite index set I , and weak transitions $\mu \rightsquigarrow \gamma_i$ and a factor $c_i \in (0, 1]$ for every $i \in I$, with $\sum_{i \in I} c_i = 1$ and $\gamma = \bigoplus_{i \in I} c_i \gamma_i$. Let the set of splittings of immediate successor sub-distributions be defined as $\text{split}(\mu) = \{(\mu_1, \mu_2) \mid \exists \mu' : \mu \rightsquigarrow_C \mu' \wedge \mu' = \mu_1 \oplus \mu_2\}$.

Definition C.2 (Weak bisimulation). A symmetric relation \mathcal{R} on sub-distributions over S is called a weak bisimulation if and only if whenever $\mu_1 \mathcal{R} \mu_2$ then for all $\alpha \in \mathbb{R} \cup \{\epsilon\} : |\mu_1| = |\mu_2|$ and for all $s \in \text{supp}(\mu_1)$ there exists $\mu_2^{\rightarrow}, \mu_2^{\Delta} \in \text{split}(\mu_2)$ and

- (i) $\mu_1(s) \mathbf{1}_s \mathcal{R} \mu_2^{\rightarrow}$ and $(\mu_1 \ominus s) \mathcal{R} \mu_2^{\Delta}$
- (ii) whenever $s \xrightarrow{a} \mu'_1$ for some μ'_1 then $\mu_2^{\rightarrow} \rightsquigarrow_C^a \mu''$ and $(\mu_1(s) \cdot \mu'_1) \mathcal{R} \mu''$

Two sub-distributions μ and γ are weak bisimilar, denoted $\mu \approx \gamma$, if the pair (μ, γ) is contained in some weak bisimulation.

MA $\mathcal{M}_1, \mathcal{M}_2$ are weak bisimilar, denoted $\mathcal{M}_1 \approx \mathcal{M}_2$, if their initial (Dirac) distributions are bisimilar in the direct sum.

Lemma C.3. *For every unichain MA and stationary deterministic policy D , the induced stochastic process \mathcal{M}_D is weak bisimilar to an ergodic CTMC C .*

Proof. Let \mathcal{M}_D be the stochastic process induced by a unichain MA \mathcal{M} and stationary deterministic policy D . As \mathcal{M} is unichain it directly follows that \mathcal{M}_D is strongly connected. The proof that \mathcal{M}_D is weakly bisimilar to a CTMC C goes along the same lines as in [14] where it has been shown that the MA semantics of well-defined GSPNs is weakly bisimilar to their CTMC semantics. As the stochastic process \mathcal{M}_D can be considered as a 1-safe GSPN that by D is well-defined, the result follows. \square

Theorem 4.2. *For unichain MA \mathcal{M} , $LRA^{\min}(s, G)$ equals $R^{\min}(s)$ in $\text{mdp}(\mathcal{M})$.*

Proof. Let \mathcal{M} be a unichain MA with state space S and $G \subseteq S$ a set of goal states. We consider a stationary deterministic policy D on \mathcal{M} . It follows that there exists an ergodic CTMC C such that $\mathcal{M}_D \approx C$. Note that $G \subseteq MS$; thus G can be represented by the union of zero or more equivalence classes under \approx .

The long-run average for state $s \in S$ and $G \subseteq S$ is given by

$$LRA^D(s, G) = \mathbb{E}_{s,D}(A_G) = \mathbb{E}_{s,D} \left(\lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t \mathbf{1}_G(\mathcal{X}_u) du \right)$$

where \mathcal{X}_u is the random variable, denoting $\pi @ u$. With the ergodic theorem from [29] we obtain that almost surely

$$\frac{1}{t} \int_0^t \mathbf{1}_{\{s_i \in \mathcal{X}_u\}} du \rightarrow \frac{1}{m_i E(s_i)} \text{ as } t \rightarrow \infty$$

holds, where m_i is the expected return time to state s_i . Therefore, in our induced ergodic CTMC, almost surely

$$\mathbb{E}_{s_i} \left(\lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t \mathbf{1}_{\{s_i\}}(\mathcal{X}_u) du \right) = \frac{1}{m_i \cdot E(s_i)}. \quad (\text{C.1})$$

Thus, almost surely the fraction of time to stay in s_i in the long-run is $\frac{1}{m_i \cdot E(s_i)}$. Let μ_i be the probability to stay in s_i in the long-run in the embedded DTMC of C where

$\mathbf{P}(s, s') = \frac{\mathbf{R}(s, s')}{E(s)}$. Thus $\mu \cdot \mathbf{P} = \mu$ where μ is the vector containing μ_i for all states $s_i \in S$. Given the probability of μ_i of staying in state s_i the expected return time is given by

$$m_i = \frac{\sum_{s_j \in S} \mu_j \cdot E(s_j)^{-1}}{\mu_i}. \quad (\text{C.2})$$

Gathering these results yields:

$$\begin{aligned} LRA^D(s, G) &= \mathbb{E}_{s,D} \left(\lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t \mathbf{1}_G(\mathcal{X}_u) \, du \right) = \mathbb{E}_{s,D} \left(\lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t \sum_{s_i \in G} \mathbf{1}_{\{s_i\}}(\mathcal{X}_u) \, du \right) \\ &= \sum_{s_i \in G} \mathbb{E}_{s,D} \left(\lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t \mathbf{1}_{\{s_i\}}(\mathcal{X}_u) \, du \right) \stackrel{(\text{C.1})}{=} \sum_{s_i \in G} \frac{1}{m_i \cdot E(s_i)} \\ &\stackrel{(\text{C.2})}{=} \sum_{s_i \in G} \frac{\mu_i}{\sum_{s_j \in S} \mu_j \cdot E(s_j)^{-1}} \cdot \frac{1}{E(s_i)} = \frac{\sum_{s_i \in G} \mu_i \cdot E(s_i)^{-1}}{\sum_{s_j \in S} \mu_j \cdot E(s_j)^{-1}} \\ &= \frac{\sum_{s_i \in S} \mathbf{1}_G(s_i) \cdot \mu_i E(s_i)^{-1}}{\sum_{s_j \in S} \mu_j \cdot E(s_j)^{-1}} = \frac{\sum_{s_i \in S} \mu_i \cdot (\mathbf{1}_G(s_i) \cdot E(s_i)^{-1})}{\sum_{s_j \in S} \mu_j \cdot E(s_j)^{-1}} \\ &= \frac{\sum_{s_i \in S} \mu_i \cdot c_1(s_i, D(s_i))}{\sum_{s_j \in S} \mu_j \cdot c_2(s_j, D(s_j))} \stackrel{[11]}{=} \mathbb{E}_{s,D}(\mathcal{R}) \end{aligned}$$

Thus, by definition there exists a one-to-one correspondence between the policy D of \mathcal{M} and its corresponding MDP $\mathbf{mdp}(\mathcal{M})$. With the results from above this yields that $LRA^{\min}(s, G) = \inf_D LRA^D(s, G)$ in MA \mathcal{M} equals $R^{\min}(s, \diamond G) = \inf_D \mathbb{E}_{s,D}(\mathcal{R})$ in $\mathbf{mdp}(\mathcal{M})$. \square

APPENDIX D. PROOF OF LEMMA 4.3

Lemma 4.3. Let \mathcal{M} be a maximal end component and D a stationary deterministic policy inducing a multichain on \mathcal{M} . Then there exists a stationary deterministic policy D' inducing a unichain on \mathcal{M} such that the long-run ratio is at least as good as for D .

Proof. (sketch). By the limit in the long-run ratio definition of \mathcal{R} it follows that for every $i \geq 0$ the prefix of π up to i does not matter. Thus, $\mathcal{R}(\pi) = \mathcal{R}(\pi_i)$ where π_i denotes the path π from the i -th position onwards. Therefore, given a policy D , inducing a multichain on maximal end component \mathcal{M} , we can construct a unichain policy D' as follows: Let D' fixes the recurrent class S' of \mathcal{M} with the minimal value induced by D (in case of the maximal long-run ratio, the maximal value respectively). For states outside of S' , D' is a policy that reaches S' with probability 1. \square

APPENDIX E. PROOF OF THEOREM 4.4

Theorem 4.4. For MA $\mathcal{M} = (S, A, \rightarrow, \Longrightarrow, s_0)$ with MECs $\{\mathcal{M}_1, \dots, \mathcal{M}_k\}$ with state spaces $S_1, \dots, S_k \subseteq S$, and set of goal states $G \subseteq S$:

$$LRA^{\min}(s_0, G) = \inf_{D \in GM} \sum_{j=1}^k LRA_j^{\min}(G) \cdot \Pr_{s_0, D}(\diamond \square S_j),$$

where $\Pr_{s_0, D}(\diamond \square S_j)$ is the probability to eventually reach and continuously stay in some states in S_j from s_0 under policy D and $LRA_j^{\min}(G)$ is the LRA of $G \cap S_j$ in unichain MA \mathcal{M}_j .

Proof. (sketch). Let \mathcal{M} be a finite MA with maximal end components $\{\mathcal{M}_1, \dots, \mathcal{M}_k\}$, $G \subseteq S$ a set of goal states, and $\pi \in Paths(\mathcal{M})$ an infinite path in \mathcal{M} . For all policies D , each path π can be partitioned into finite and infinite paths of the form

$$\begin{aligned} \pi_{s_0 s} &= s_0 \xrightarrow{\alpha_0, \mu_0, t_0} s_1 \xrightarrow{\alpha_1, \mu_1, t_1} \dots \xrightarrow{\alpha_n, \mu_n, t_n} s, \text{ and} \\ \pi_s^\omega &= s \xrightarrow{\alpha_s, \mu_s, t_s} \dots \xrightarrow{\alpha_i, \mu_i, t_i} s \dots \end{aligned}$$

where $\pi_{s_0 s}$ is the path starting in initial state s_0 and ends in $s \in \mathcal{M}_i$ for some $0 < i \leq k$. Further, all states on path π_s^ω belong to maximal end component \mathcal{M}_i . Note, that a state on path $\pi_{s_0 s}$ can be part of another maximal end component \mathcal{M}_j (as in Example 4.6). Hence, it is not sufficient to only check if eventually a MEC is reached, as done in the corresponding theorem for IMCs in [18]. Thus, the minimal LRA will be obtained when the LRA in each MEC \mathcal{M}_i is minimal and the combined LRA of all MECs is minimal according to their persistence under policy D . \square

APPENDIX F. PROOF OF THEOREM 4.7

Theorem 4.7. For MA \mathcal{M} , $LRA^{\min}(s_0, G)$ equals $cR^{\min}(s_0, \diamond Q)$ in SSP $\mathbf{ssp}_{lra}(\mathcal{M})$.

Proof. Let π be an infinite path in the MDP $\mathbf{ssp}_{lra}(\mathcal{M})$ such that $\pi[i_Q]$ is the first visit of a state in Q along π , i.e., for all $j < i_Q$, $\pi[j] \notin Q$ and $\pi[i_Q] \in Q$. Similarly, we define i_q for a single state q . We define random variable $C_Q : Paths \rightarrow \mathbb{R}_{\geq 0}$ by $C_Q(\pi) = g(\pi[i_Q])$. Note that $D \in GM$.

$$\begin{aligned} cR^{\min}(s_0, \diamond Q) &= \inf_D \mathbb{E}_{s_0, D}(C_Q) \\ &= \inf_D \sum_{\pi \in Paths} C_Q(\pi) \cdot \Pr_{s_0, D}(\pi) = \inf_D \sum_{\pi \in Paths} \sum_{i=1}^k C_{\{q_i\}}(\pi) \cdot \Pr_{s_0, D}(\pi) \\ &= \inf_D \sum_{i=1}^k \sum_{\pi \in Paths} C_{\{q_i\}}(\pi) \cdot \Pr_{s_0, D}(\pi) = \inf_D \sum_{i=1}^k LRA_i^{\min}(G) \cdot \Pr_{s_0, D}(\diamond \{q_i\}) \\ &\stackrel{(*)}{=} \inf_D \sum_{i=1}^k LRA_i^{\min}(G) \cdot \Pr_{s_0, D}(\diamond \square S_i) = LRA^{\min}(s_0, G). \end{aligned}$$

Observe that in step (*) we use the transformation from Definition 4.5 in reverse. Hence, if $\Pr_{s, D}(\diamond q_i) > 0$, we eventually reach the maximal end component \mathcal{M}_i and always stay in it. Otherwise $\Pr_{s, D}(\diamond q_i) = 0$ and policy D chooses an action such that we leave \mathcal{M}_i or never even visit \mathcal{M}_i . \square

APPENDIX G. PROOF OF THEOREM 5.3

Let MA $\mathcal{M} = (S, A, \rightarrow, \Longrightarrow, s_0)$, $G \subseteq S$ and time interval $I = [0, b] \in \mathcal{Q}$ with $b \geq 0$. Let $\lambda = \max_{s \in MS} E(s)$ be the largest exit rate of any Markovian state and $\delta \in \mathbb{R}_{>0}$ be the discretisation step, chosen such that $b = k_b \delta$ for some $k_b \in \mathbb{N}$. We recall the definition of $\diamond^I G$ as the set of all paths that reach some goal state in G within interval I . Let random variable $\#_J: Paths \rightarrow \mathbb{N}$, where $J \in \mathcal{Q}$ is a time interval. Intuitively $\#_J$ counts the number of Markovian jumps happened inside interval J . For example $\#[0, \delta] = 1$ denotes the set of paths having exactly one Markovian transition in the first δ time units. Random vector $\#^{I, \delta}: Paths \rightarrow \mathbb{N}^{k_b}$ with I, δ and k_b as explained before, is defined as the vector of k_b elements, each counting Markovian jumps occurred in the corresponding chunk of length δ , i.e. $\#^{I, \delta} = (\#[0, \delta], \dots, \#[(k_b-2)\delta, (k_b-1)\delta], \#[(k_b-1)\delta, b])^T$. Moreover, let $\|\cdot\|_\infty$ denote the maximum norm, which takes the maximum over the absolute value of the elements of the given vector.

Lemma G.1. *Let \mathcal{M}_δ be the dMA induced by \mathcal{M} with respect to discretisation constant δ . Then for all $s \in S$:*

$$p_{\max}^{\mathcal{M}_\delta}(s, \diamond^{[0, k_b]} G) = \sup_{D \in GM} \Pr_{s, D}(\diamond^I G \mid \|\#^{I, \delta}\|_\infty < 2).$$

Proof. As discussed in Section 5, paths of \mathcal{M}_δ are essentially the paths from \mathcal{M} that carry only zero or one Markovian transitions in each discretisation step δ . Hence, for computing reachability probabilities in step interval $[0, k_b]$ in \mathcal{M}_δ , it is enough to consider paths in \mathcal{M} with at most one Markovian jumps in each δ time units. This set is described by $\|\#^{I, \delta}\|_\infty < 2$. \square

Lemma G.2. *For all $s \in S$ and $D \in GM$ in \mathcal{M} : $\Pr_{s, D}(\diamond^I G \mid \#[0, \delta] < 2) \leq \Pr_{s, D}(\diamond^I G)$.*

Proof. We assume $b > 0$, since for $b = 0$, $\Pr_{s, D}(\diamond^I G \mid \#[0, \delta] < 2) = \Pr_{s, D}(\diamond^I G)$. We have

$$\begin{aligned} \Pr_{s, D}(\diamond^I G) &= \Pr_{s, D}(\diamond^I G \cap \#[0, \delta] > 0) + \Pr_{s, D}(\diamond^I G \cap \#[0, \delta] = 0) \\ &= \Pr_{s, D}(\diamond^I G \cap \#[0, \delta] > 0) + \Pr_{s, D}(\diamond^I G \mid \#[0, \delta] = 0) \cdot \Pr_{s, D}(\#[0, \delta] = 0). \end{aligned} \quad (\text{G.1})$$

On the other hand we have

$$\begin{aligned} \Pr_{s, D}(\diamond^I G \mid \#[0, \delta] < 2) &= \Pr_{s, D}(\diamond^I G \mid \#[0, \delta] < 2, \#[0, \delta] = 1) \cdot \Pr_{s, D}(\#[0, \delta] = 1 \mid \#[0, \delta] < 2) \\ &\quad + \Pr_{s, D}(\diamond^I G \mid \#[0, \delta] < 2, \#[0, \delta] = 0) \cdot \Pr_{s, D}(\#[0, \delta] = 0 \mid \#[0, \delta] < 2). \end{aligned} \quad (\text{G.2})$$

We distinguish between two cases:

(i) $s \in MS \setminus G$: In this case, Eq. (G.1) gives

$$\Pr_{s, D}(\diamond^I G) = \int_0^\delta E(s) e^{-E(s)t} \sum_{s' \in S} \mathbf{P}(s, s') \Pr_{s', D}(\diamond^{I \ominus t} G) dt + \Pr_{s, D}(\diamond^{I \ominus \delta} G) e^{-E(s)\delta}. \quad (\text{G.3})$$

and for Eq. (G.2) we have

$$\begin{aligned} \Pr_{s, D}(\diamond^I G \mid \#[0, \delta] < 2) &= \int_0^\delta E(s) e^{-E(s)t} \sum_{s' \in S} \mathbf{P}(s, s') \Pr_{s', D}(\diamond^{I \ominus \delta} G) dt \\ &\quad + \Pr_{s, D}(\diamond^{I \ominus \delta} G) e^{-E(s)\delta}. \end{aligned} \quad (\text{G.4})$$

Since $\Pr_{s,D}(\diamond^{I \ominus t} G)$ is monotonically decreasing in t , we have $\Pr_{s,D}(\diamond^{I \ominus \delta} G) \leq \Pr_{s,D}(\diamond^{I \ominus t} G)$, $t \leq \delta$. Putting this in Eq. (G.3) and (G.4) leads to

$$\Pr_{s,D}(\diamond^I G \mid \#_{[0,\delta]} < 2) \leq \Pr_{s,D}(\diamond^I G).$$

- (ii) $s \in PS \setminus G$: From the law of total probability, we split time bounded reachability into two parts. First we compute the probability to reach the set of Markovian states from s by only taking probabilistic transitions in zero time, and then we quantify the probability to reach some goal state in G from Markovian states inside interval I . Therefore:

$$\begin{aligned} \Pr_{s,D}(\diamond^I G) &= \sum_{s' \in MS} \Pr_{s,D}(\diamond^{[0,0]} \{s'\}) \Pr_{s',D}(\diamond^I G) \\ &\stackrel{(*)}{\geq} \sum_{s' \in MS} \Pr_{s,D}(\diamond^{[0,0]} \{s'\}) \Pr_{s',D}(\diamond^I G \mid \#_{[0,\delta]} < 2) \\ &= \Pr_{s,D}(\diamond^I G \mid \#_{[0,\delta]} < 2), \end{aligned}$$

where $(*)$ follows from case (i) above. \square

Lemma G.3. For all $s \in S \setminus G$ and $D \in GM$ in \mathcal{M} :

$$\Pr_{s,D}(\diamond^I G \mid \|\#^{I,\delta}\|_\infty < 2) \leq \Pr_{s,D}(\diamond^I G \mid \#_{[0,\delta]} < 2).$$

Proof. The lemma holds for $b = 0$, since in this case, $\Pr_{s,D}(\diamond^I G \mid \|\#^{I,\delta}\|_\infty < 2) = \Pr_{s,D}(\diamond^I G \mid \#_{[0,\delta]} < 2)$. For $b > 0$, we decompose $\Pr_{s,D}(\diamond^I G \mid \|\#^{I,\delta}\|_\infty < 2)$ as Eq. (G.2) into:

$$\begin{aligned} &\Pr_{s,D}(\diamond^I G \mid \|\#^{I,\delta}\|_\infty < 2, \#_{[0,\delta]} = 1) \cdot \Pr_{s,D}(\#_{[0,\delta]} = 1 \mid \|\#^{I,\delta}\|_\infty < 2) \\ &+ \Pr_{s,D}(\diamond^I G \mid \|\#^{I,\delta}\|_\infty < 2, \#_{[0,\delta]} = 0) \cdot \Pr_{s,D}(\#_{[0,\delta]} = 0 \mid \|\#^{I,\delta}\|_\infty < 2). \end{aligned} \quad (\text{G.5})$$

Now we prove the lemma by induction over k_b .

- $k_b = 1$: This case holds because interval $I = [0, \delta]$ contains one discretisation step and then $\Pr_{s,D}(\diamond^I G \mid \|\#^{I,\delta}\|_\infty < 2) = \Pr_{s,D}(\diamond^I G \mid \#_{[0,\delta]} < 2)$.
- $k_b - 1 \rightsquigarrow k_b$: Let I be $[0, b]$ and assume the lemma holds for interval $[0, (k_b - 1)\delta]$ (i.e. $I \ominus \delta$):

$$\Pr_{s,D}(\diamond^{I \ominus \delta} G \mid \|\#^{I \ominus \delta, \delta}\|_\infty < 2) \leq \Pr_{s,D}(\diamond^{I \ominus \delta} G \mid \#_{[0,\delta]} < 2). \quad (\text{G.6})$$

In order to show that the lemma holds for I , we distinguish between two cases:

- (i) $s \in MS \setminus G$: From Eq. (G.4) we have:

$$\Pr_{s,D}(\diamond^I G \mid \#_{[0,\delta]} < 2) = \sum_{s' \in S} \mathbf{P}(s, s') \Pr_{s',D}(\diamond^{I \ominus \delta} G) (1 - e^{-E(s)\delta}) + \Pr_{s,D}(\diamond^{I \ominus \delta} G) e^{-E(s)\delta}. \quad (\text{G.7})$$

Similarly from Eq. (G.5) we have:

$$\begin{aligned}
\Pr_{s,D}(\diamond^I G \mid \|\#^{I,\delta}\|_\infty < 2) &= \sum_{s' \in S} \mathbf{P}(s, s') \Pr_{s',D}(\diamond^{I \ominus \delta} G \mid \|\#^{I \ominus \delta, \delta}\|_\infty < 2) (1 - e^{-E(s)\delta}) \\
&\quad + \Pr_{s,D}(\diamond^{I \ominus \delta} G \mid \|\#^{I \ominus \delta, \delta}\|_\infty < 2) e^{-E(s)\delta} \\
&\stackrel{(G.6)}{\leq} \sum_{s' \in S} \mathbf{P}(s, s') \Pr_{s',D}(\diamond^{I \ominus \delta} G) (1 - e^{-E(s)\delta}) \\
&\quad + \Pr_{s,D}(\diamond^{I \ominus \delta} G) e^{-E(s)\delta} \\
&\stackrel{(G.7)}{=} \Pr_{s,D}(\diamond^I G \mid \#[0,\delta] < 2)
\end{aligned}$$

- (ii) $s \in PS \setminus G$: This case utilises the previously discussed idea of splitting paths using the law of total probabilities into two parts. The first part contains the set of paths that reach Markovian states from s in zero time using probabilistic transitions, while the second includes paths reaching some state in G from Markovian states. Hence:

$$\begin{aligned}
\Pr_{s,D}(\diamond^I G \mid \|\#^{I,\delta}\|_\infty < 2) &= \sum_{s' \in MS} \Pr_{s,D}(\diamond^{[0,0]} \{s'\}) \Pr_{s',D}(\diamond^I G \mid \|\#^{I,\delta}\|_\infty < 2) \\
&\stackrel{(*)}{\leq} \sum_{s' \in MS} \Pr_{s,D}(\diamond^{[0,0]} \{s'\}) \Pr_{s',D}(\diamond^I G \mid \#[0,\delta] < 2) \\
&= \Pr_{s,D}(\diamond^I G \mid \#[0,\delta] < 2),
\end{aligned}$$

where (*) follows from case (i) above. □

Lemma G.4. For all $s \in S \setminus G$: $p_{\max}^{\mathcal{M}_\delta}(s, \diamond^{[0,k_b]} G) \leq p_{\max}^{\mathcal{M}}(s, \diamond^I G)$.

Proof.

$$p_{\max}^{\mathcal{M}_\delta}(s, \diamond^{[0,k_b]} G) = \sup_{D \in GM} \Pr_{s,D}(\diamond^I G \mid \|\#^{I,\delta}\|_\infty < 2) \quad (\text{Lemma G.1})$$

$$\leq \sup_{D \in GM} \Pr_{s,D}(\diamond^I G \mid \#[0,\delta] < 2) \quad (\text{Lemma G.3})$$

$$\leq \sup_{D \in GM} \Pr_{s,D}(\diamond^I G) = p_{\max}^{\mathcal{M}}(s, \diamond^I G). \quad (\text{Lemma G.2})$$

□

Lemma G.5. For all $s \in S \setminus G$:

$$p_{\max}^{\mathcal{M}}(s, \diamond^I G) \leq p_{\max}^{\mathcal{M}_\delta}(s, \diamond^{[0,k_b]} G) + 1 - e^{-\lambda b} (1 + \lambda \delta)^{k_b}.$$

Proof.

$$\begin{aligned}
p_{\max}^{\mathcal{M}}(s, \diamond^I G) &= \sup_{D \in GM} \Pr_{s,D}(\diamond^I G) \\
&= \sup_{D \in GM} \left(\Pr_{s,D}(\diamond^I G \cap \|\#^{I,\delta}\|_{\infty} < 2) + \Pr_{s,D}(\diamond^I G \cap \|\#^{I,\delta}\|_{\infty} \geq 2) \right) \\
&\leq \sup_{D \in GM} \Pr_{s,D}(\diamond^I G \cap \|\#^{I,\delta}\|_{\infty} < 2) + \sup_{D \in GM} \Pr_{s,D}(\diamond^I G \cap \|\#^{I,\delta}\|_{\infty} \geq 2) \\
&\leq \sup_{D \in GM} \Pr_{s,D}(\diamond^I G \mid \|\#^{I,\delta}\|_{\infty} < 2) + \sup_{D \in GM} \Pr_{s,D}(\diamond^I G \cap \|\#^{I,\delta}\|_{\infty} \geq 2) \\
&\stackrel{(\dagger)}{=} p_{\max}^{\mathcal{M}_{\delta}}(s, \diamond^{[0,k_b]} G) + \sup_{D \in GM} \Pr_{s,D}(\diamond^I G \cap \|\#^{I,\delta}\|_{\infty} \geq 2) \\
&\leq p_{\max}^{\mathcal{M}_{\delta}}(s, \diamond^{[0,k_b]} G) + \sup_{D \in GM} \Pr_{s,D}(\|\#^{I,\delta}\|_{\infty} \geq 2),
\end{aligned}$$

where (\dagger) follows from Lemma G.1.

It remains to find an upper bound for $\sup_{D \in GM} \Pr_{s,D}(\|\#^{I,\delta}\|_{\infty} \geq 2)$ which is the maximum probability to have more than one Markovian jump in at least one time step among k_b time step(s) of length δ . Due to the independence of the number of Markovian jumps in discretisation steps, this probability can be upper bounded by k_b independent Poisson processes, all parametrised with the maximum exit rate exhibited in \mathcal{M} . In each Poisson process the probability of at most one Markovian jump in one discretisation step is $e^{-\lambda\delta}(1 + \lambda\delta)$, therefore the probability of a violation of this assumption in at least one discretisation step is $1 - e^{-\lambda b}(1 + \lambda\delta)^{k_b}$. Hence

$$\begin{aligned}
p_{\max}^{\mathcal{M}}(s, \diamond^I G) &\leq p_{\max}^{\mathcal{M}_{\delta}}(s, \diamond^{[0,k_b]} G) + \sup_{D \in GM} \Pr_{s,D}(\|\#^{I,\delta}\|_{\infty} \geq 2) \\
&\leq p_{\max}^{\mathcal{M}_{\delta}}(s, \diamond^{[0,k_b]} G) + 1 - e^{-\lambda b}(1 + \lambda\delta)^{k_b}. \quad \square
\end{aligned}$$

Theorem 5.3. *Given MA $\mathcal{M} = (S, A, \rightarrow, \Longrightarrow, s_0)$, $G \subseteq S$, interval $I = [0, b] \in \mathcal{Q}$ with $b \geq 0$ and $\lambda = \max_{s \in MS} E(s)$. Let $\delta \in \mathbb{R}_{>0}$ be such that $b = k_b \delta$ for some $k_b \in \mathbb{N}$. Then, for all $s \in S$ it holds that*

$$p_{\max}^{\mathcal{M}_{\delta}}(s, \diamond^{[0,k_b]} G) \leq p_{\max}^{\mathcal{M}}(s, \diamond^{[0,b]} G) \leq p_{\max}^{\mathcal{M}_{\delta}}(s, \diamond^{[0,k_b]} G) + 1 - e^{-\lambda b}(1 + \lambda\delta)^{k_b}.$$

Proof. For $s \in G$ we have that $p_{\max}^{\mathcal{M}_{\delta}}(s, \diamond^{[0,k_b]} G) = p_{\max}^{\mathcal{M}}(s, \diamond^{[0,b]} G) = 1$. For $s \in S \setminus G$, the theorem follows from Lemma G.4 and G.5. \square