Don't Know in Probabilistic Systems

Harald Fecher[1], Martin Leucker[2], and Verena Wolf[3]

[1] Institute of Informatics, University of Kiel, Germany
[2] Institute of Informatics, TU Munich, Germany
[3] Institute of Informatics, University of Mannheim, Germany

Abstract. In this paper the abstraction-refinement paradigm based on 3-valued logics is extended to the setting of probabilistic systems. We define a notion of abstraction for Markov chains. To be able to relate the behavior of abstract and concrete systems, we equip the notion of abstraction with the concept of simulation. Furthermore, we present model checking for abstract probabilistic systems (abstract Markov chains) with respect to specifications in probabilistic temporal logics, interpreted over a 3-valued domain. More specifically, we introduce a 3-valued version of probabilistic computation-tree logic (PCTL) and give a model checking algorithm w.r.t. abstract Markov chains.

1 Introduction

Abstraction is one of the most successful techniques for fighting the state space explosion problem in model checking [4]. Abstractions hide some of the details of the verified system, thus resulting in a smaller model. In the seminal papers on abstraction-based model checking, *conservative* abstractions for *true* have been studied. In this setting, if a formula is true in the abstract model then it is also true in the concrete (precise) model of the system. However, if it is false in the abstract model then nothing can be deduced for the concrete one [3].

In the 3-valued setting, the goal is to define abstractions that are conservative for both true and false. Therefore, a third value *indefinite* (also called *don't know*), denoted by ?, is introduced that identifies when too much information is hidden to decide whether the formula evaluates to true or false in the concrete system. Thus, *indefinite* indicates that the abstract system has to be *refined*, meaning that less information should be concealed.

Kripke Modal Transition Systems (KMTS, [13]) have become a popular device to model abstractions of transition systems. In the abstraction process, states of the concrete system are grouped together in the abstract system. Transitions between sets of concrete states are then classified as *must* or *may* edges. Very roughly, may edges are a kind of over approximation while must edges are a kind of under approximation.

In this paper, we study abstractions for (labeled discrete-time) Markov chains (MCs). MCs are a typical underlying model for sequential probabilistic programs or probabilistic process algebras [19]. In simple words, MCs are transition systems where the transitions are enriched with transition probabilities. To get an

abstraction in the same spirit as the one for KMTS, one could again group states of the concrete system together to obtain an abstract system. Then we have to come up with a suitable notion of over and under approximation of transitions. We suggest to label transitions by intervals of probabilities, similar as in [15, 20]. The lower bound of an interval represents an under approximation while the upper bound is used for the over approximation.

This motivates to define the notion of *Abstract Markov Chains* (AMCs) as a kind of transition system where transitions are labeled with intervals of probabilities. To compare the behavior of a given AMC and a given MC, we introduce a simulation relation, called probabilistic simulation. We call an AMC M' coarser or an abstraction of AMC M if M' simulates M and vice versa M is called finer or a refinement of M'. We show that the abstractions obtained by the process mentioned above are in the simulation relation.

When AMCs are used in the context of abstraction, we motivate that only certain combinations of intervals are meaningful and call such AMCs *delimited. Cutting* arbitrary AMCs to delimited ones, also during the model checking process, will give more precise results at (nearly) no cost, as we will describe.

Our main motivation for abstraction is model checking. For probabilistic systems, Jonsson and Hansson introduced Probabilistic Computation Tree Logic (PCTL) [10] that allows formulation of statements involving the measure of certain paths. We give PCTL a 3-valued semantics over AMCs. The semantics is defined, as we show, in the right manner w.r.t. abstractions: If a formula evaluates to true or false in the abstract system, it does so in the concrete system. If the result is *indefinite*, nothing can be said about the concrete system.

We then present (two versions of) a model checking algorithm for AMCs and 3-valued PCTL. The gist of our algorithms is to use 3-valued combinations instead of boolean (as in the 2-valued case) for state formulas and to compute measures for each path property similar as in the setting of Markov decision processes [1, 7].

Recently, 3-valued-based model checking and refinement has gained a lot of interest. A framework for 3-valued abstraction is introduced in [13, 8]. In [17, 16, 2], model checking of 3-valued (or multi-valued) versions of CTL or CTL* have been studied. Game-based approaches allow an elegant treatment of refinement and have been presented in [18, 9] in the setting of CTL and respectively the μ-calculus.

General issues for abstractions of probabilistic systems are discussed in [12, 14] while we concentrate on a specific abstraction together with dedicated model checking algorithms. The closest works to ours are [6] and [11]. In [6], Markov decision processes are proposed for abstracting of MCs. However, they only consider reachability properties while we study PCTL model checking. More importantly, our notion of simulation is coarser—thus allowing for coarser and therefore smaller abstractions—while maintaining soundness w.r.t. 3-valued PCTL. In [11][1], criterias have been engineered that guarantee an abstraction to be optimal (in some sense). While, of course, such an optimal abstraction sounds preferable,

[1] We thank the author for providing us the as yet unpublished manuscript.

the approach loses some of its elegance since—in simple words—it requires storage of much information. Furthermore, it is not clear (to us) how to obtain this information without constructing the underlying MC.

We conclude our paper by discussing the pros and cons of the different approaches in detail and order them w.r.t. their precision (in a sense made precise below).

Outline AMCs are derived in the next section. Before, introducing 3-valued PCTL in Section 4, we discuss the relation of measures of paths in finer and coarser systems first for reachability properties, in Section 3. The model checking algorithms for PCTL is given in Section 5. We compare our framework with existing ones in Section 6.

2 Abstract Markov Chains

To introduce our notion of abstraction, let us consider the Markov chain shown in Figure 1(a). A Markov chain consists of states labeled with propositions. The states are connected by transitions that are labeled with probabilities for taking the corresponding transitions. Following the idea of Kripke Modal Transition Systems, states of the concrete system are grouped together in the abstract system. For example, s_5 and s_6 form the abstract state A_2 (Figure 1(b)). While in the case of transition systems, we obtain so-called *may*- and *must*-transitions denoting that there may be a transition from one state to the other, or, there is a transition for sure, we deal with lower and upper bounds on the transition probabilities here. For example, we say that we move from A_2 to A_1 with some probability in $[0, \frac{1}{4}]$ since we either cannot move to A_1 (when in s_6) or move to A_1 with probability $\frac{1}{4}$ (when in s_5). This motivates the definition of an *abstract* Markov chain. Let us first fix some notation:

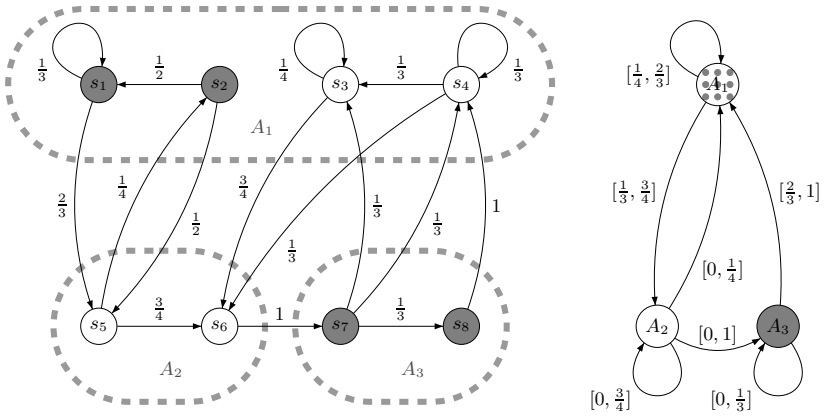
Let AP be a nonempty finite set of *propositions* and $\mathbb{B}_3 = \{\bot, ?, \top\}$ the three valued truth domain. Let X be a finite set. For $Y, Y' \subseteq X$ and a function $Q : X \times X \to \mathbb{R}$ let $Q(Y, Y') = \sum_{y \in Y} \sum_{y' \in Y'} Q(y, y')$. We omit brackets if Y or Y' is a singleton. The function $Q(x, \cdot)$ is given by $x' \mapsto Q(x, x')$ for all $x' \in X$. Furthermore let $psdistr(X) = \{f : X \to [0, 1]\}$ be the set of all *pseudo distribution functions* on X and $distr(X) = \{f \in psdistr(X) \mid \sum_{x \in X} f(x) = 1\}$ the set of *distributions* on X.

Definition 1. *An* abstract Markov chain *(AMC) is a tuple* (S, P^l, P^u, L) *where:*

- S *is a finite set of states,*
- $P^l, P^u : S \times S \to [0, 1]$ *are matrices describing the lower and upper bounds for the transition probabilities between states such that for all* $s, s' \in S$, $P^l(s, \cdot)$ *and* $P^u(s, \cdot)$ *are pseudo distribution functions and*

$$P^l(s, s') \leq P^u(s, s') \text{ and } P^l(s, S) \leq 1 \leq P^u(s, S), \tag{1}$$

- $L : S \times AP \to \mathbb{B}_3$ *is a labeling function that assigns a truth value to each pair of state and proposition.*



(a) Markov Chain (b) AMC

Fig. 1. A Markov chain and its abstraction

Note that with condition (1) we do not consider states without any outgoing transition. We call an AMC $M = (S, P^l, P^u, L)$ a *Markov chain* (MC) if $P^l = P^u =: P$. Note that in this case $P(s, \cdot) \in \mathit{distr}(S)$, for all $s \in S$. Let X be a finite set. Let g^l, g^u be a pair of functions in $\mathit{psdistr}(X)$ with $g^l(x) \leq g^u(x)$ for all $x \in X$. We write $g(x)$ for the interval $[g^l(x), g^u(x)] \subseteq [0, 1]$ and $\mathit{distr}(g)$ for the set $\{f \in \mathit{distr}(X) \mid \forall x \in X : f(x) \in g(x)\}$. If $g^l = Q^l(x, \cdot)$ and $g^u = Q^u(x, \cdot)$ for some $Q^l, Q^u \in \mathit{psdistr}(X \times X)$ we put $\mathit{distr}(Q(x, \cdot)) = \mathit{distr}(g)$.

Let us now formalize the notion of abstraction:

Definition 2. *Let $M = (S, P^l, P^u, L)$ be an AMC and $\mathcal{A} = \{A_1, A_2, \ldots, A_n\} \subseteq 2^S$ a partition of S, i.e. $A_i \neq \emptyset$, $A_i \cap A_j = \emptyset$ for $i \neq j$, $1 \leq i, j \leq n$ and $\bigcup_{i=1}^n A_i = S$. Then the abstraction of M induced by $\mathcal{A}$ is the AMC $\mathit{abstract}(M, \mathcal{A}) = (\tilde{S}, \tilde{P}^l, \tilde{P}^u, \tilde{L})$ given by*

- $\tilde{S} = \mathcal{A}$,
- $\tilde{P}^l(A_i, A_j) = \min_{s \in A_i} P^l(s, A_j)$ *and* $\tilde{P}^u(A_i, A_j) = \max_{s \in A_i} P^u(s, A_j)$.
- *For $a \in AP$ the labeling of an abstract state (also called* macro state*) $A \in \mathcal{A}$ is given by*

$$\tilde{L}(A, a) = \begin{cases} \top, & \text{if } L(s, a) = \top \text{ for all } s \in A, \\ \bot, & \text{if } L(s, a) = \bot \text{ for all } s \in A, \\ ?, & \text{otherwise.} \end{cases}$$

Example 1. Figure 1 (a) illustrates a MC with 8 states. These states are grouped together, denoted by the dashed grey circles, to form the abstract system with three states.[2] The intervals of probabilities are obtained as described before. For

[2] Note that the question of how to partition the state space usually depends on where MCs are used and is beyond the scope of this paper.

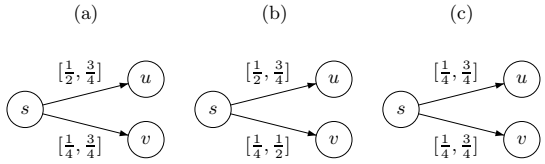


Fig. 2. Sharpening and widening the intervals

simplicity we consider a single proposition $a \in AP$ that holds exactly in all grey shaded states. Thus, in the abstract system, we get $\tilde{L}(A_1, a) =?$, $\tilde{L}(A_2, a) = \bot$ and $\tilde{L}(A_3, a) = \top$.

Scheduler In the setting of AMCs, in every state s, there is a choice for the distribution yielding the probabilities to reach successor states. This non-determinism can be resolved by means of a scheduler: A (history-dependent) *scheduler* for a state s_0 is a function $\eta : s_0 S^* \to \mathit{distr}(S \times S)$ that maps each sequence of states $s_0 \dots s$ to a distribution in $\mathit{distr}(P(s, \cdot))$. The set of all schedulers for an AMC M starting in state s_0 is denoted by $\mathcal{S}(M, s_0)$. We write $\mathcal{S}(s_0)$ if M is clear from the context.

Delimited AMCs Since a scheduler is defined to select only distributions (rather than pseudo distributions), we can sharpen the definition of AMCs, motivated as follows (see also [20]):

Consider the AMC M in Figure 2, (a)[3]. Assume that one chooses the value $\frac{3}{4}$, i.e. $P^u(s, v) = P^l(s, v) = \frac{3}{4}$, from the interval $[\frac{1}{4}, \frac{3}{4}]$ labeling the transition from state s to v. But then, the transition probability from state s to u is $P(s, u) = 1 - P(s, v) = \frac{1}{4} \notin [\frac{1}{2}, \frac{3}{4}]$. This problem does not occur in case (b) and (c) of Figure 2. The AMC in case (b) is "finer" than (a) since $[\frac{1}{4}, \frac{1}{2}] \subset [\frac{1}{4}, \frac{3}{4}]$, whereas case (c) is more abstract than (a). In the following we will "cut" AMCs so that cases with "non-constructive" information do not occur and give a transformation that refines an AMC such that the conditions are fulfilled. Thus, for the example, we change from case (a) to the finer model of (b) rather than to (c).

Definition 3. *For a finite set X let $g^l, g^u \in \mathit{psdistr}(X)$ with $g^l(x) \leq g^u(x)$ for all $x \in X$. The* cut *of g^l and g^u is the pair $\mathit{cut}(g^l, g^u) = (f^l, f^u)$ given by*

$$f^l(x) = \min\{h(x) \mid h \in \mathit{distr}(g)\} \qquad f^u(x) = \max\{h(x) \mid h \in \mathit{distr}(g)\}$$

We call an AMC $M = (S, P^l, P^u, L)$ delimited *iff for all $s \in S$ it holds that*

$$\mathit{cut}(P^l(s, \cdot), P^u(s, \cdot)) = (P^l(s, \cdot), P^u(s, \cdot)).$$

[3] We sometimes omit outgoing transitions in examples from now on.

Summing up, *cut* deletes values that cannot be completed to a distribution, so no scheduler of an AMC gets lost:

Lemma 1. *Let $M = (S, P^l, P^u, L)$ be an AMC and $M' = (S, cut(P^l, P^u), L)$ the* delimited version *of M. Then for all $s \in S$,*

$$\mathcal{S}(M, s) = \mathcal{S}(M', s)$$

Note that the *cut* operator is easy to calculate, e.g., the lower bound of the transition probability for s to s' will be $\max\{P^l(s, s'), 1 - \sum_{s'' \neq s'} P^u(s, s'')\}$ in the delimited version. If a lower bound is adapted no upper bound has to be adapted and vice versa.

If we construct $abstract(M, \mathcal{A})$ we always receive a delimited AMC if M is a MC. This is not necessarily the case if M is an AMC, even if M is delimited, as Figure 3 shows.

Remark 1. In the following we assume that w.l.o.g. all considered AMCs are delimited unless otherwise stated.

Extreme Distributions As will become apparent in the following, distributions taking up values on the borders of intervals, called extreme distributions, are of special interest. Let $g^l, g^u \in psdistr(X)$ with $g^l(x) \leq g^u(x)$ for all $x \in X$ and $cut(g^l, g^u) = (g^l, g^u)$. For $X' \subseteq X$ let $ex_{min}(g^l, g^u, X')$ be the set of distributions $h \in distr(g)$ such that $X' = \emptyset$ implies $h = g^l = g^u$ and $X' \neq \emptyset$ implies

$$\exists x \in X'.h(x) = g^l(x) \wedge h \in ex_{min}(cut(g^l, g^u[x \mapsto g^l(x)]), X' \setminus \{x\}),$$

where $f[s \mapsto n]$ denotes the function that agrees everywhere with f except at s where it is equal to n. Dually, let $ex_{max}(g^l, g^u, X')$ be the set of distributions $h \in distr(g)$ such that $X' = \emptyset$ implies $h = g^l = g^u$ and $X' \neq \emptyset$ implies

$$\exists x \in X'.h(x) = g^u(x) \wedge h \in ex_{max}(cut(g^l[x \mapsto g^u(x)], g^u), X' \setminus \{x\}).$$

Definition 4. *We say that a distribution $h \in distr(g)$* min-extreme *if $h \in ex_{min}(g^l, g^u, X)$ and* max-extreme *if $h \in ex_{max}(g^l, g^u, X)$. h is called* extreme *if it is min-extreme or max-extreme.*

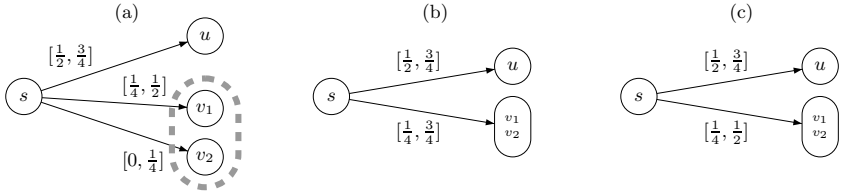


Fig. 3. Cutting abstraction: (a) abstracted to (b) delimited to (c)

Simulation To compare the behavior described by two AMCs, we introduce the notion of probabilistic simulation that is an extension of probabilistic simulation for MCs [15].

Definition 5. *Let $M = (S, P^l, P^u, L)$ be an AMC. We call $\mathcal{R} \subseteq S \times S$ a probabilistic simulation iff $s\mathcal{R}s'$ implies:*

1. *$\forall a \in AP : (L(s', a) \neq ?) \Longrightarrow L(s', a) = L(s, a)$,*
2. *for each $h \in distr(P(s, \cdot))$ there exists $h' \in distr(P(s', \cdot))$ and $\delta \in distr(S \times S)$ such that for all $u, v \in S$*

$$(i)\ \delta(u, v) > 0 \Longrightarrow u\mathcal{R}v, \quad (ii)\ \delta(u, S) = h(u), \quad (iii)\ \delta(S, v) = h'(v).$$

We write $s \preceq s'$ iff there exists a probabilistic simulation $\mathcal{R}$ with $s\mathcal{R}s'$. For AMC $M_i = (S_i, P^l{}_i, P^u{}_i, L_i)$, $s_i \in S_i$, $i = 1, 2$ we write $s_1 \preceq s_2$ iff there exists a probabilistic simulation $\mathcal{R}$ on $S_1 \cup S_2$ with $s_1\mathcal{R}s_2$ in the composed AMC of M_1 and M_2 (which is constructed in the obvious way, assuming $S_1 \cap S_2 = \emptyset$).

Note that if $s \preceq s'$ then all possible distributions h of s are matched by a distribution h' of s'. The opposite does not hold, i.e., the set $distr(P(s', \cdot))$ may contain distributions that can not be simulated by a distribution of s.

The previously defined abstraction operator induces a simulation:

Theorem 1. *Let $M = (S, P^l, P^u, L)$ be an AMC and $abstract(M, \mathcal{A})$ an abstraction of M induced by a partition $\mathcal{A}$ of S. Then s is simulated by its macro state, i.e. for all $s \in S, A \in \mathcal{A}$*

$$s \in A \implies s \preceq A.$$

Example 2. Consider the MC M of Example 1, Figure 1. We have $s_4 \preceq A_1$, for instance: Let $\mathcal{R} = \{(s_i, A_1) \mid 1 \leq i \leq 4\} \cup \{(s_5, A_2), (s_6, A_2)\} \cup \{(s_7, A_3), (s_8, A_3)\}$. Since $\tilde{L}(a, A_1) = ?$ condition (1) of Definition 5 is trivially fulfilled. Checking condition (2) for (s_4, A_1) yields: $\delta(s_3, A_1) = \delta(s_4, A_1) = \delta(s_6, A_2) = \frac{1}{3}$ and 0 for all remaining pairs. Then $h' \in distr(\mathcal{A})$ with $h'(A_1) = \delta(s_3, A_1) + \delta(s_4, A_1) = \frac{2}{3}$, $h'(A_2) = \frac{1}{3}$, and $h'(A_3) = 0$ is an element of $distr(\tilde{P}(A_1, \cdot))$ such that condition (2) is fulfilled.

3 Measures and Simulation

Let us define a notion of measure for AMCs and discuss how measures are related w.r.t. simulation. Here, we study reachability properties. In the next section, we extend our study to a three-valued version of Probabilistic Computation Tree Logic (PCTL).

A nonempty set Ω of possible outcomes of an experiment of chance is called *sample space*. A set $\mathfrak{B} \subseteq 2^\Omega$ is called *Borel field* (or σ-algebra) over Ω if it contains Ω, $\Omega \setminus E$ for each $E \in \mathfrak{B}$, and the union of any countable sequence

of sets from $\mathfrak{B}$. The subsets of Ω that are elements of $\mathfrak{B}$ are called *measurable* (w.r.t. $\mathfrak{B}$). A Borel field $\mathfrak{B}$ is *generated* by an at most countable set $\mathcal{E}$, denoted by $\mathfrak{B} = \langle \mathcal{E} \rangle$, if $\mathfrak{B}$ is the closure of $\mathcal{E}$'s elements under complement and countable union.

A *probability space* is a triple $\mathcal{PS} = (\Omega, \mathfrak{B}, \mathrm{Prob})$ where Ω is a sample space, $\mathfrak{B}$ is a Borel field over Ω, and Prob is a mapping $\mathfrak{B} \to [0,1]$ such that $\mathrm{Prob}(\Omega) = 1$ and $\mathrm{Prob}(\bigcup_{i=1}^{\infty} E_i) = \sum_{i=1}^{\infty} \mathrm{Prob}(E_i)$ for any sequence $E_1, E_2, \ldots$ of pairwise disjoint sets from $\mathfrak{B}$. We call Prob a *probability measure*.

For an AMC $M = (S, P^l, P^u, L)$, let $\Omega = S^\omega$ be the set of *trajectories* (also called *paths*) of M. Let $\mathfrak{B}$ be the Borel field generated by $\{\mathcal{C}(\pi) \mid \pi \in S^*\}$, where $\mathcal{C}(\pi) = \{\pi' \in \Omega \mid \pi \text{ is a prefix of } \pi'\}$ is the *basic cylinder set* of π. A scheduler $\eta \in \mathcal{S}(M, s_0)$ induces a probability space $\mathcal{PS}^\eta = (\Omega, \mathfrak{B}, \mathrm{Prob}^\eta)$ as follows: Prob^η is uniquely given by $\mathrm{Prob}^\eta(\Omega) = 1$ and, for $n \geq 1$, $\mathrm{Prob}^\eta(\mathcal{C}(s_0 s_1 \ldots s_n)) = h_1(s_1) \ldots h_n(s_n)$, where $h_i = \eta(s_0 \ldots s_{i-1})$, for $i \in \{1, \ldots, n\}$, is the probability distribution selected by η. We set $\mathrm{Prob}^\eta(\mathcal{C}(s'_0 s'_1 \ldots s'_n)) = 0$ if $s'_0 \neq s_0$. Furthermore, we put $\pi(s) = \mathcal{C}(s)$ and for $n = 0, 1, 2, \ldots$ let $\pi[n]$ denote the n-th state of π.

When interested in the infimum of probabilities of measurable sets w.r.t. all schedulers, it suffices to consider only extreme distributions, which take values only at boundaries of intervals. A scheduler is called *extreme* iff it only chooses extreme distributions. The set of all extreme schedulers for state s is denoted by $\mathcal{ES}(M, s)$ and $\mathcal{ES}(s)$ if M is known.

Theorem 2. *For state s in an AMC, we have for every measurable set Q of the induced probability space that*

$$\inf_{\eta \in \mathcal{ES}(s)} \mathit{Prob}^\eta(Q) = \inf_{\eta \in \mathcal{S}(s)} \mathit{Prob}^\eta(Q)$$

The previous theorem can easily be shown as follows: Take a scheduler η and show that the measure is reduced (or stays the same) when changing η to an extreme distribution.

Note that while there are typically infinitely many distributions leading from one state to the other in an AMC, there are only finitely many extreme distributions.

Let us compare the notion of AMCs with the one of Markov decision processes (MDPs) in the three-valued setting: A *Markov decision process* (MDP) is a tuple $M = (S, \Sigma, \mathit{Prob}, L)$, where S is a finite set of states, Σ is a non-empty finite set of letters, $\mathit{Prob} : S \times \Sigma \rightharpoonup \mathit{distr}(S)$ is a partial function that yields for a state s and a given letter σ a distribution function for successor states. $L : S \times AP \to \mathbb{B}_3$ is a labeling function that assigns a truth value to each pair of state and proposition.

The MDP $M' = \mathit{MDP}(M)$ induced by an AMC $M = (S, P^l, P^u, L)$ is given as $M' = (S, \Sigma, \mathit{Prob}, L)$ where $\Sigma = \{\sigma_h \mid h \in \mathit{distr}(P(s, \cdot))$ for some $s \in S$ and h is extreme$\}$, Prob is such that $\mathit{Prob}(s, \sigma_h) = h$ if $h \in \mathit{distr}(P(s, \cdot))$ and h is extreme and $\mathit{Prob}(s, \sigma_h)$ is undefined otherwise.

Thus, $\mathit{MDP}(M)$ defines a Markov decision process with the same state space as M but with (finitely-many) extreme distributions. The notion of schedulers

carries over in the expected manner, i.e. $\mathcal{ES}(M,s) = \mathcal{ES}(MDP(M),s)$ for all states s. More importantly, the infimum of the measure of some measurable set with respect to some scheduler class obviously coincides, due to Theorem 2.

For the remainder of this section, let us now concentrate on reachability properties. More specifically, for an AMC M and s one of its state, a proposition $a \in AP$, $\alpha \in \mathbb{B}_3$ and $n = 0, 1, 2, \ldots$, let, $Reach(s,a,\alpha,n) := \{\pi \in \pi(s) \mid L(\pi[n],a) = \alpha$ and for all $k < n, L(\pi[k],a) \neq \alpha\}$ and

$$Reach(s,a,\alpha) = \bigcup_{n \geq 0} Reach(s,a,\alpha,n)$$

For reachability properties, it was shown in the setting of Markov decision processes (MDPs), that the infimum with respect to all schedulers agrees with the one when only so-called simple schedulers are considered [7]. A scheduler $\eta \in \mathcal{S}(M,s)$ is called *simple* iff for all $\pi, \pi' \in S^*$, $s' \in S$, we have $\eta(s\pi s') = \eta(s\pi' s')$, meaning that the choice does not depend on the history π. Thus, a similar result holds for AMCs as well. The set of simple schedulers that choose only extreme distributions is denoted by $\mathcal{SES}(M,s)$ for AMC or MDP M. Since there are only finitely many simple extreme schedulers, the infimum is indeed a minimum. Thus, we get

Lemma 2. *For state s, $a \in AP$, and $\alpha \in \mathbb{B}_3$ it holds that*

$$\begin{aligned} &\inf_{\eta \in \mathcal{S}(s)} Prob^{\eta}(Reach(s,a,\alpha)) \\ =\ &\inf_{\eta \in \mathcal{SES}(s)} Prob^{\eta}(Reach(s,a,\alpha)) \\ =\ &\min_{\eta \in \mathcal{SES}(s)} Prob^{\eta}(Reach(s,a,\alpha)) \end{aligned}$$

Let us now compare the behavior of two AMCs w.r.t. abstraction, i.e., simulation. We give the intuition of the following Lemma first. Let $s_0 \preceq s_0'$. When scheduler $\eta \in \mathcal{S}(s_0)$ chooses some distribution h_0, there is, according to the definition of simulation, a corresponding $h_0' \in distr(P(s_0', \cdot))$. This implies that for every state s_1 reachable by h_0 with positive probability, there is a set of states $s_{1_1}', \ldots, s_{1_{k_1}}'$ reachable by h_0' with positive probability, each simulating s_1. Now, for $s_0 s_1$, we can argue in the same fashion: For $\eta(s_0 s_1) = h_1$ there is a corresponding h_{1_i}' for each $s_0' s_{1_i}'$, and so on...

Let us be more precise: For a scheduler $\eta \in \mathcal{S}(s_0)$ we define a scheduler $\eta' \in \mathcal{S}(s_0')$ inductively as follows: For $h = \eta(s_0)$ define $\eta'(s_0') = h'$, where h' is as in the definition of the simulation relation. Similarly, let $s_0 \ldots s_n$ be a sequence of states such that $\mathrm{Prob}^{\eta}(\mathcal{C}(s_0 \ldots s_n)) > 0$ and $h = \eta(s_0 \ldots s_n)$. By induction, there is a set of states $s_{n_1}', \ldots s_{n_k}'$ each simulating s_n. For each $s_{n'}' \in \{s_{n_1}', \ldots s_{n_k}'\}$, define $\eta'(s_0' \ldots s_{n'}') = h'$, where h' is as in the definition of the simulation relation.

Lemma 3. *For $\alpha \in \{\top, \bot\}$, $a \in AP$ it holds that $s \preceq s'$ implies*

$$\inf_{\eta \in \mathcal{S}(s)} Prob^{\eta}(Reach(s,a,\alpha)) \geq \inf_{\eta' \in \mathcal{S}(s')} Prob^{\eta'}(Reach(s',a,\alpha))$$

The previous lemma can be shown by induction on n, where n is the position where the proposition a has value α for the first time. Induction hypothesis is that

$$\begin{aligned} Prob^{\eta}(Reach(s,a,\alpha,n)) &= Prob^{\eta'}(Reach(s',a,\alpha,n)) \\ &\geq \inf_{\eta'' \in \mathcal{S}(s)} Prob^{\eta''}(Reach(s',a,\alpha,n)) \end{aligned}$$

where η' is the scheduler constructed for η as described above and η may be the one for which the infimum is taken.

Note that for the supremum, the corresponding result only holds when adding the paths that reach a state for which a evaluates to ?:

Lemma 4. *For $\alpha \in \{\top, \bot\}$, $a \in AP$ we have that $s \preceq s'$ implies*

$$\sup_{\eta \in \mathcal{S}(s)} Prob^{\eta}(Reach(s,a,\alpha)) \leq \sup_{\eta' \in \mathcal{S}(s')} Prob^{\eta'}(Reach(s',a,\alpha) \cup Reach(s',\eta',a,?))$$

Thus, Lemma 2 and Lemma 3 yield that the lower bound for some reachability property in the coarser system is less or equal than in the finer system.

Theorem 3. *Let s, s' be states in an AMC with $s \preceq s'$ and $a \in AP$, and $\alpha \in \{\top, \bot\}$. Then*

$$\min_{\eta \in \mathcal{S}(s)} Prob^{\eta}(Reach(s,a,\alpha)) \geq \min_{\eta' \in \mathcal{SES}(s')} Prob^{\eta'}(Reach(s',a,\alpha))$$

In simple words, the previous theorem says that when the minimum of a reachability property is at least p in the coarser system, it is so in the finer system as well.

4 3-valued PCTL

Recall that AP denotes a nonempty finite set of *propositions*. The set of *Probabilistic Computation-Tree Logic (PCTL)* [10, 5] formulas over AP, denoted by PCTL, is the set of *state-formulas* φ inductively defined as follows:

$$\varphi ::= \text{true} \mid a \mid \varphi \wedge \varphi \mid \neg\varphi \mid [\Phi]_{\bowtie p} \qquad \Phi ::= X\varphi \mid \varphi \,\mathcal{U}\, \varphi$$

where $\bowtie \in \{\leq, <, \geq, >\}$, $p \in [0,1]$ and $a \in AP$. The formulas defined by Φ are called *path-formulas*[4].

In the setting of AMCs, a state might no longer just satisfy or refuse a formula, but a third value ? (don't know) is appropriate. Consequently, we define the satisfaction of a formula w.r.t. a state as a function into $\mathbb{B}_3$, which forms a complete lattice ordering the elements as $\bot < ? < \top$. Joins and meets in this lattice are denoted by $\sqcup$ and $\sqcap$, respectively. Complementation is denoted by $^-$, where $\top$ and $\bot$ are complementary to each other while $\overline{?} = ?$.

[4] To simplify the presentation, we omit the bounded until operator given in [10], which could easily be added.

$$
\begin{array}{lll}
[\![s, \text{true}]\!] & = \top & [\![s, \text{false}]\!] = \bot \\
[\![s, a]\!] & = L(s,a) & \\
[\![s, \varphi_1 \wedge \varphi_2]\!] & = [\![s,\varphi_1]\!] \sqcap [\![s,\varphi_2]\!] & [\![s, \neg\varphi_1]\!] = \overline{[\![s,\varphi_1]\!]} \\
[\![s, [\Phi]_{\geq p}]\!] & = \begin{cases} \top & \text{if } Pr^l(s,\Phi,\top) \geq p \\ \bot & \text{if } Pr^l(s,\Phi,\bot) > 1-p \\ ? & \text{otherwise} \end{cases} & [\![s, [\Phi]_{\leq p}]\!] = \begin{cases} \top & \text{if } Pr^l(s,\Phi,\bot) \geq 1-p \\ \bot & \text{if } Pr^l(s,\Phi,\top) > p \\ ? & \text{otherwise} \end{cases} \\
[\![\pi, X\varphi_1]\!] & = [\![\pi[1], \varphi_1]\!] & \\
[\![\pi, \varphi_1 \,\mathcal{U}\, \varphi_2]\!] & = \begin{cases} \top & \text{if } \exists i.([\![\pi[i],\varphi_2]\!] = \top \text{ and } \forall 0 \leq j < i.[\![\pi[j],\varphi_1]\!] = \top) \\ \bot & \text{if } \forall i.([\![\pi[i],\varphi_2]\!] \neq \bot \implies \exists 0 \leq j < i.[\![\pi[j],\varphi_1]\!] = \bot) \\ ? & \text{otherwise,} \end{cases} &
\end{array}
$$

Fig. 4. Semantics of PCTL formulas

When a formula evaluates in a state to $\top$ or $\bot$, we sometimes say that the result is *definite*. Otherwise, we say that it is *indefinite*. Similarly, we say the result holds for sure or is violated for sure if it evaluates to $\top$ respectively $\bot$. We say it may be true or may be false if it evaluates to ?.

Given an AMC $M = (S, P^l, P^u, L)$ and a PCTL formula φ we define the satisfaction function $[\![s, \varphi]\!]$ for state $s \in S$ and $[\![\pi, \Phi]\!]$ for trajectory $\pi \in S^\omega$ inductively as shown in Figure 4, where $Pr^l(s, \Phi, \alpha) = \inf_{\eta \in \mathcal{SES}(s)} Prob^\eta(\{\pi \in \pi(s) \mid [\![\pi, \Phi]\!] = \alpha\})$ for $\alpha \in \mathbb{B}_3$. For the cases $\bowtie = <$ and $\bowtie = >$ the value of $[\![s, [\Phi]_{\bowtie p}]\!]$ is similar to the cases $\leq$ and $\geq$, respectively, but we exchange $\leq$ by $<$ and vice versa.

To understand why the above semantics is sound with respect to the notion of simulation in Definition 5 we discuss each operator in the following and state the soundness result later in Theorem 4.

Case $\text{true}, \text{false}, a, \wedge, \neg$*:* The semantics is defined as expected for the base and boolean cases.

Case X *and* $\mathcal{U}$*:* The truth value of $[\![\pi, X\varphi_1]\!]$ equals the result of φ_1 in state $\pi[1]$. A trajectory π satisfies the formula $\varphi_1 \,\mathcal{U}\, \varphi_2$ for sure, if φ_1 holds for sure until φ_2 holds for sure. It is violated, if either φ_2 is always wrong for sure, or otherwise φ_1 is violated before.

Case $[\Phi]_{\geq p}$*:* For $[\Phi]_{\geq p}$, the situation is slightly more involved. First, we remark that Lemma 2 holds also for PCTL path properties, i.e. that is suffices to consider simple extreme schedulers instead of arbitrary ones.

Lemma 5. *Let M be an AMC, s one of its states, Φ a path property of PCTL, $\alpha \in \mathbb{B}_3$, and $Q = \{\pi \in \pi(s) \mid [\![\pi, \Phi]\!] = \alpha\}$. Then*

$$\inf_{\eta \in \mathcal{S}(s)} Prob^\eta(Q) = \inf_{\eta \in \mathcal{SES}(s)} Prob^\eta(Q) = \min_{\eta \in \mathcal{SES}(s)} Prob^\eta(Q)$$

In view of the simulation relation we can show that coarser systems yield even lower bounds than finer systems.

Lemma 6. *For states s, s' in an AMC with $s \preceq s'$ and Φ a path property of PCTL, $\alpha \in \{\top, \bot\}$, $Q = \{\pi \in \pi(s) \mid [\![\pi, \Phi]\!] = \alpha\}$, $Q' = \{\pi \in \pi(s') \mid [\![\pi, \Phi]\!] = \alpha\}$*

we have

$$\min_{\eta \in \mathcal{SES}(s)} Prob^{\eta}(Q) \geq \min_{\eta' \in \mathcal{SES}(s')} Prob^{\eta'}(Q')$$

The previous lemmas can easily be shown as their counterparts for reachability properties listed in the previous section.

For $[\Phi]_{\geq p}$, we measure the paths starting in s for which Φ evaluates to $\top$ and check whether the lower bound of this measure is greater or equal to p. If so, the result is $\top$ and for a finer state s' with $s' \preceq s$ this measure is also greater than p.

For scheduler $\eta \in \mathcal{SES}(M, s')$ we set $p^{\eta}_{\alpha} = Prob^{\eta}(\{\pi \in \pi(s') \mid [\![\pi, \Phi]\!] = \alpha\})$ and observe that $\sum_{\alpha \in \mathbb{B}_3} p^{\eta}_{\alpha} = 1$. If the measure of the paths starting in s for which Φ evaluates to $\bot$ is greater than $1 - p$, then this is also the case for s', i.e. $p^{\eta}_{\bot} > 1 - p$. Therefore, this leaves less than $1 - (1 - p) = p$ for $p^{\eta}_{\top} + p^{\eta}_{?}$. In other words, even if $p^{\eta}_{?}$ is added to $p^{\eta}_{\top}$, the constraint $\geq p$ cannot be met. Therefore, we decide for $\bot$.

Case $[\Phi]_{\leq p}$*:* For $[\Phi]_{\leq p}$, we consider the measure of paths starting in s for which Φ evaluates to $\top$. If the lower bound is already bigger than p, it is so especially so for s' and we decide for $[\Phi]_{\leq p}$ as $\bot$. Similarly, if for enough paths Φ evaluates to $\bot$, we can be sure that the measure of paths satisfying Φ is small. If $Pr^{l}(s, \Phi, \bot) \geq 1 - p$ then in the finer system for all $\eta \in \mathcal{SES}(M, s')$ we get $p^{\eta}_{\bot} \geq 1 - p$. But then $p^{\eta}_{?} + p^{\eta}_{\top} \leq 1 - (1 - p) = p$. In other words, even if $p^{\eta}_{?}$ is added to $p^{\eta}_{\top}$, the constraint $\leq p$ is fulfilled and we go for $\top$.

The following theorem states that our framework developed so far can indeed be used for abstraction based model checking and follows easily from Lemma 6 and the discussion above. In simple words, it says that the result of checking a formula in the abstract system agrees with the one for the finer system, unless it was indefinite.

Theorem 4. *Let s and s' be two states of an AMC M with $s \preceq s'$. Then for all $\varphi \in PCTL$:*

$$[\![s', \varphi]\!] \neq ? \text{ implies } [\![s, \varphi]\!] = [\![s', \varphi]\!].$$

Observe that the 3-valued PCTL semantics of an MC understood as an AMC coincides with the usual 2-valued PCTL semantics for Markov chains.

5 Model Checking 3-valued PCTL

In this section, we discuss two model checking algorithms for 3-valued PCTL. As for CTL, both model checking algorithms work bottom-up the parse tree of φ. Hence, it suffices to describe their steps inductively on the structure of φ. Each state s is labeled with a function t_s assigning to each subformula its truth value. t_s is defined directly for true, false, a, $\varphi_1 \wedge \varphi_2$, and $\neg\varphi_1$ according to the definition of their semantics. For $[\Phi]_{\bowtie p}$, t_s can easily be determined, provided the lower bound of a measure of paths for some path property (denoted by Pr^{l} in Figure 4) can be computed. Therefore, it remains to show how to compute

the lower bound of the measure of paths for which an until or next-step formula evaluates to $\top$, $\bot$, and ?. Let us discuss $\Phi := \varphi_1 \,\mathcal{U}\, \varphi_2$. The treatment of the next-step operator is similar but easier and is omitted here. Thanks to Theorem 2, computing the measure for an until property becomes (technically) easy, since only extreme distributions have to be considered.

Reduction to MDP model checking The first idea is to convert an AMC M to an MDP $MDP(M)$ and reuse existing methods for computing path properties on MDPs. Before translating M, we can assume that for every state, we know the truth values of φ_1 and φ_2. We annotate $MDP(M)$ with (two-valued) propositions corresponding to the values of φ_1 and φ_2 in M. More specifically, label a state s of $MDP(M)$ by the (new) propositions a_{φ_1} and a_{φ_2}, if φ_1 respectively φ_2 evaluates to $\top$ in s. Label s by propositions $\bar{a}_{\varphi_1}$ and $\bar{a}_{\varphi_2}$, if φ_1 respectively φ_2 are $\bot$. Now, considering the semantics of the until operation as shown in Figure 4, it is easy to see that $\varphi_1 \,\mathcal{U}\, \varphi_2$ on a path of M evaluates to $\top$ iff $a_{\varphi_1} \,\mathcal{U}\, a_{\varphi_2}$ on the same path (of $MDP(M)$) evaluates to true. Similarly, it is easy to see that $\varphi_1 \,\mathcal{U}\, \varphi_2$ on a path of M evaluates to $\bot$ iff $\neg(\neg\bar{a}_{\varphi_1} \,\mathcal{U}\, \neg\bar{a}_{\varphi_2})$ evaluates to true.

Using the reduction to an MDP model checking problem for until properties, we have completed the first algorithm that is mainly used to give an upper bound on the complexity of the model checking problem.

Complexity Computing the semantics for an AMC M and a formula $\varphi \in$ PCTL bottom-up for every state can be done in linear time, provided the measures for path properties are given. For every state s with k outgoing transitions, one can obtain, in the worst case, $k!$ extreme distributions. Thus, the size of $MDP(M)$ is at most exponential in the size of M, where, as expected, the size of M, denoted by $|M|$ is the number of states plus the number of transitions, i.e., pairs (s, s') for which $P^l(s, s') > 0$. Computing the measure for a path property in an MDP M' is polynomial with respect to the size of M' (states plus non-zero transitions) [7]. Thus, overall, we get:

Theorem 5. *Given an AMC $M = (S, P^l, P^u, L)$ and a PCTL formula φ, then the algorithm outlined in this section labels every state $s \in S$ with $t_s(\psi) = [\![s, \psi]\!]$ for each subformula ψ of φ in time polynomial w.r.t. $O(2^{|M| \log |M|})$ and linear w.r.t. the size of φ, where $|M|$ denotes the size of M.*

Fixpoint computation The reduction to an MDP for computing path properties suffers from the effort spent for computing all extreme distributions. Therefore, we have implemented a version of the algorithm that is based on fixpoint iteration. This algorithm, while (only) approximating the minimal result in question, chooses (and computes) extreme distributions in an on-the-fly fashion, leading to huge space gains.

Our approach is inspired by the treatment in [1, 5] done for MDPs. Let us define the sets:

$$
\begin{aligned}
W^+_\top &= \{s \mid t_s(\varphi_2) = \top\}\\
W^-_\top &= \{s \mid t_s(\varphi_2) \neq \top \text{ and } t_s(\varphi_1) \neq \top\}\\
W^+_\bot &= \{s \mid t_s(\varphi_2) = \bot \text{ and } t_s(\varphi_1) = \bot\}\\
W^-_\bot &= \{s \mid t_s(\varphi_2) \neq \bot\}
\end{aligned}
$$

To simplify our presentation, we say that Φ evaluates in a state to some value in $\mathbb{B}_3$ if it evaluates to that value on all paths starting in this state.

Φ holds in $W^+_\top$ for sure and is violated for sure in $W^-_\top$. However, the result is $\bot$ in $W^+_\bot$ since φ_1 as well as φ_2 is $\bot$. In $W^-_\bot$ the formula is not necessarily violated.

Let p^{min}_α abbreviate $(Pr^l(s, \Phi, \alpha))_{s\in S}$. We obtain p^{min}_α as least fixpoint of the iteration described in the following:

First, let Del be the set of all pairs of delimited pseudo distribution functions on S and $b \in \{l, u\}$. Consider the minimization/maximization function $\xi^b : 2^S \times \text{Del} \times (S \to [0,1]) \to [0,1]$ that is given by $\xi^b(\emptyset, (g^l, g^u), x) = 0$ and for $S' \neq \emptyset$

$$
\begin{aligned}
\xi^l(S', (g^l, g^u), x) \;&=\; g^l(s^l) \cdot x(s^l) + \xi^l(S' \setminus \{s^l\}, cut(g^l, g^u[s^l \mapsto g^l(s^l)]), x)\\
&\qquad\qquad \text{if } \; x(s^l) = \min_{s'\in S'} x(s'),\\
\xi^u(S', (g^l, g^u), x) \;&=\; g^u(s^u) \cdot x(s^u) + \xi^u(S' \setminus \{s^u\}, cut(g^l[s^u \mapsto g^u(s^u)], g^u), x)\\
&\qquad\qquad \text{if } \; x(s^u) = \max_{s'\in S'} x(s').
\end{aligned}
$$

Note that $\xi^b(S, (g^l, g^u), x)$ sorts the states $s \in S'$ according to their values in x and chooses $h \in \mathit{distr}(g)$ that minimizes/ maximizes the value $\sum_{s\in S} h(s) \cdot x(s)$.[5]

Let $S^+, S^- \subseteq S$. We use ξ^b to define the function $\mathrm{F}^b_{(S^-,S^+)} : (S \to [0,1]) \to (S \to [0,1])$ that determines the next iteration step by

$$
\mathrm{F}^b_{(S^-,S^+)}(x)_{(s)} = \begin{cases} 1 & \text{if } s \in S^+,\\ 0 & \text{if } s \in S^-,\\ \xi^b(S, (P^l(s,\cdot), P^u(s,\cdot)), x) & \text{otherwise.}\end{cases}
$$

Furthermore, let x_0 denote the function that maps everything to 0.

Theorem 6. *The least fixpoint (w.r.t. point wise extension of the order of the real numbers) of the function $\mathrm{F}^b_{(S^-,S^+)}$ can be used to calculate the values p^{min}_α:*

$$
\begin{aligned}
p^{min}_\top(s) \;&=\; (\sqcup_{n\in\mathbb{N}} {\mathrm{F}^l_{(W^-_\top, W^+_\top)}}^{(n)}(x_0))_{(s)}\\
p^{min}_\bot(s) \;&=\; 1 - (\sqcup_{n\in\mathbb{N}} {\mathrm{F}^u_{(W^+_\bot, W^-_\bot)}}^{(n)}(x_0))_{(s)}.
\end{aligned}
$$

[5] The function ξ^b is well defined, i.e., the same value is obtained if another maximal/minimal state s' is considered. This follows from the fact that $\min\{\sum_{s\in S'} h(s) \mid h \in \mathit{distr}(g)\} = \min\{\sum_{s\in S'} h(s) \mid h \in \mathit{distr}(cut(g^l, g^u[s' \mapsto g^l(s')]))\}$ for all $S' \subseteq S$ with $s' \in S'$.

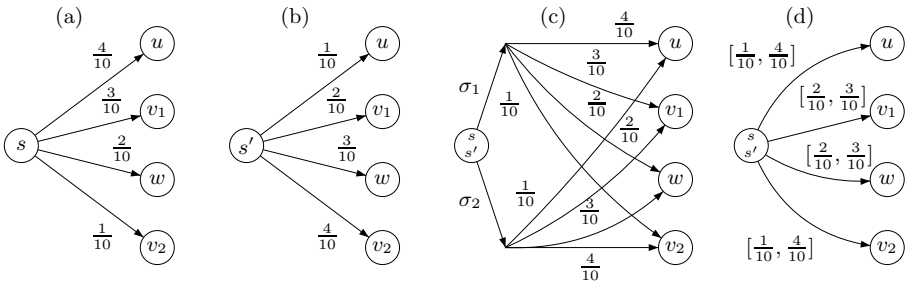


Fig. 5. Abstraction by MDPs vs. AMCs

The proof goes along the lines of the proof for MDPs (see [1, Chapter 3] for details).

Let us give an example showing that the *cut* in the definition of the fixpoint operator in Theorem 6 to calculate the probabilities for $[\Phi]_{\bowtie p}$ is indeed important:

Example 3. Let us consider the AMC shown in Figure 3 (a) (page 6) and $\Phi = \varphi_1 \ \mathcal{U} \ \varphi_2$. Assume that $t_s(\varphi_1) = t_{v_2}(\varphi_1) = t_u(\varphi_2) = \top$ and all remaining truth values for φ_1 and φ_2 are $\bot$. Furthermore, assume that there is a $[1,1]$-transition from v_2 back to itself. Then we get: $W^+_\top = \{u\}$ and $W^-_\top = \{v_1\}$. For example, the maximization function ξ^u chooses $P^l(s,u) = P^u(s,u) = \frac{3}{4}$ since $1 = \max\{1,0,0\} = \max\{x(u), x(v_1), x(v_2)\}$ after the first iteration step and due to the *cut* operation $P(s,v_1) = [\frac{1}{4}, \frac{1}{4}]$ and $P(s,v_2) = [0,0]$. Hence, $Pr^u(s,\Phi,\top) = \frac{3}{4} \cdot 1 + \frac{1}{4} \cdot 0 + 0 \cdot 0 = \frac{3}{4}$. $W^+_\bot = \{v_1, v_2\}$ and $W^-_\bot = \{u\}$. Altogether we get $Pr(s,\Phi,\top) = [\frac{1}{2}, \frac{3}{4}]$, $Pr(s,\Phi,\bot) = [\frac{1}{4}, \frac{1}{2}]$, $Pr(s,\Phi,?) = [0,0]$. Note that we get the intervals shown in Figure 3 (c). Thus, the subsequent cut in the definition of the fixpoint operator is necessary since the values in Figure 3 (b) yield less precise results.

6 Alternatives to AMCs

Let us discuss alternative approaches for abstraction of Markov chains. For reasons of space limitation, we keep the discussion informal.

Generally, *Markov Decision Processes* (MDPs) are considered to be abstractions for Markov chains. MDPs extend the model of MCs by allowing several distribution functions in each state (see Figure 5 (c)).

Thus, when merging states to obtain an abstraction, one could define the corresponding distribution functions, as indicated in Figure 5 (a)–(c). Hence, the result would be an MDP. Now, one might be tempted to use existing model checking theory for PCTL and MDPs to reason about the underlying Markov chain. However, this is not possible since, as far as we know, there is no 3-valued notion of PCTL for MDPs (not to mention, we need one that suits the role in the abstraction defined here).

When interested in reachability properties, the approach is possible and was pursued in [6]. Let us call the approach *AMDP*. Actually, the model checking algorithms presented in the previous section considers the AMC as an MDP with extreme distributions, but only when computing the minimal probabilities of path properties.

Of course, one could have developed such a 3-valued version of PCTL for MDPs as opposed for AMCs, as done here. But actually, the 3-valued PCTL semantics given in Section 4 can easily be taken over for such a 3-valued PCTL semantics for MDPs.

However, there is an intrinsic difference in the approach using AMCs and the one based on MDPs. An MDP can easily be abstracted to an AMC. For example, for the MDP shown in Figure 5 (c), we would get the AMC shown in Figure 5 (d). But using intervals, one *reduces more* information.

This has two implications, one theoretical and one practical. Our semantics for PCTL path properties compares only extreme distributions. Probabilities that are not the bound of some transition probability interval are not considered. However, we might consider *all* extreme distributions. For example, one extreme distribution for the AMC in Figure 5 (d) is $(u \mapsto \frac{4}{10}, v_1 \mapsto \frac{2}{10}, w \mapsto \frac{3}{10}, v_2 \mapsto \frac{1}{10})$, which is not present in Figure 5 (c). Now, consider $\varphi = [X(a_u \vee a_w)]_{\leq \frac{6}{10}}$, where proposition a_u (a_w) is $\top$ in state u (respectively w) and $\bot$ in all other states. Then the macro state in Figure 5 (c) provides $\top$ for φ but for the AMC in Figure 5 (d) the result is ?. Thus, our results might sometimes be less precise. From the practical side, using MDPs, one reduces the number of states but basically all distributions are kept. But storing all distributions causes no memory savings and it is questionable whether such an abstraction does indeed satisfy practical needs. In our approach, on the other hand, if, for example, a third distribution denoted by σ_3 with $(u \mapsto \frac{2}{10}, v_1 \mapsto \frac{2}{10}, w \mapsto \frac{3}{10}, v_2 \mapsto \frac{3}{10})$ would be present in Figure 5 (c), we obtain the same AMC, thus, reducing the memory requirments.

A different approach was taken in [11]. There, criterias have been engineered that guarantee an abstraction to be optimal (in some sense). Let us call this approach *O*. While, of course, such an optimal abstraction sounds preferable, it turns out that neither AMCs nor MDPs carry enough information to be optimal. In simple words, the approach loses some of its elegance since it requires to store much information. Furthermore, it is not clear (to us) how to obtain this information without constructing the underlying Markov chain. The author of [11] therefore suggests as well a more simple approximation of the optimal abstraction, which we call *S*. In simple words, *S* is similar to AMCs but does not use the cut operator to optimize the information present in AMCs.

Let us discuss Example 15 of [11]: Consider Figure 6 and $\Phi = [X\neg a_{u_1}]_{>0}$ where a_{u_1} is true in u_1 and false in all other states. For the approach *S* the result in u_0 is ? because the sum of the two zero values of the lower bounds to the direct successors u_d and u_0 are added which yields $0 + 0 = 0$ (see [11, Example 15]). In our setting after the first chosen zero value, say $P^l(u_0, u_d) = P^u(u_0, u_d) = 0$ through the *cut* the next choice is $P^l(u_0, u_0) = 1 - 0.01 = 0.99$. The resulting extrem distribution is $(u_d \mapsto 0, u_0 \mapsto 0.99, u_1 \mapsto 0.01)$ which leads to $[\![u_0, \Phi]\!] = \top$.

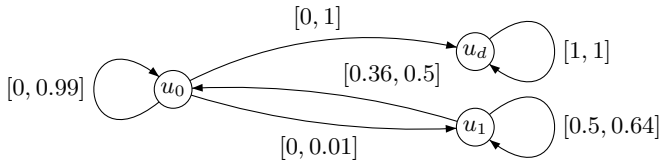


Fig. 6.

Thus, results based on S are less precise than the results obtained with our method. In terms of memory, S and AMC are comparable, provided the fixpoint computation method is used. Note that [11] does not address the question of model checking.

Summarizing, with abstraction one loses information usually by reducing space requirements. All approaches have in common, that states are grouped together to form an abstract system. They differ in the information that is kept for transitions. By means of precision, we can order the approaches $S < AMC < AMDP < O$, where $a < b$ means that a is less precise than b, when for some concrete system, the same states are grouped together. In terms of memory usage, we can order the approaches as $S = AMC < AMDP < O$, where $a < b$ means that a consumes less memory than b, when for some concrete system, the same states are grouped together.

7 Conclusion

In this paper, we have extended the abstraction-refinement paradigm based on three-valued logics to the setting of probabilistic systems. We have given a notion of abstraction for Markov chains. In simple words, abstract Markov chains are transition systems where the edges are labeled with intervals of probabilities. We equipped the notion with the concept of simulation to be able to relate the behavior of abstract and concrete systems.

We have presented model checking for abstract probabilistic systems (i.e. abstract Markov chains) with respect to specifications in a probabilistic temporal logic, interpreted over a 3-valued domain. More specifically, we studied a 3-valued version of PCTL. The model checking algorithm turns out to be quite similar to the ones developed in the setting of checking PCTL specifications of Markov decision processes. Thus, using the intuitive concept of intervals allows to refrain from saving all probability distributions present in concrete systems but storing only boundaries, while allowing to adapt existing theory of model checking probabilistic systems.

Our work can be extended into several directions. First, further insight in which and how to split states is desirable, when the model checking result is indefinite. It would also be interesting to extend our setting towards the more expressive logic PCTL* or to the setting of continuous-time Markov chains.

References

1. Christel Baier. *On the algorithmic verification of probabilistic systems*. Universität Mannheim, 1998. Habilitation Thesis.
2. M. Chechik, B. Devereux, S. Easterbrook, and A. Gurfinkel. Multi-valued symbolic model-checking. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 12:371–408, 2003.
3. E. Clarke, O. Grumberg, and D. Long. Model Checking and Abstraction. In *Proc. of POPL*, pages 342–354, New York, January 1992. ACM.
4. E.M. Clarke, O. Grumberg, and D.A. Peled. *Model Checking*. MIT press, December 1999.
5. C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, 42(4):857–907, July 1995.
6. P. D'Argenio, B. Jeannet, H. Jensen, and K. Larsen. Reduction and refinement strategies for probabilistic analysis. In *PAPM-PROBMIV*, pages 57–76, 2002.
7. Luca de Alfaro. *Formal Verification of Probabilistic Systems*. PhD thesis, Stanford University, 1997. Technical report STAN-CS-TR-98-1601.
8. P. Godefroid and R. Jagadeesan. On the expressiveness of 3-valued models. In *Verification, Model Checking and Abstract Interpretation (VMCAI)*, volume 2575 of *LNCS*, pages 206–222, 2003.
9. O. Grumberg, M. Lange, M. Leucker, and S. Shoham. *Don't know* in the μ-calculus. In *Proc. VMCAI'05*, volume 3385 of *LNCS*. Springer, 2005.
10. H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6:512–535, 1994.
11. M. Huth. On finite-state approximants for probabilistic computation tree logic. *Theoretical Computer Science*. to appear.
12. M. Huth. An abstraction framework for mixed non-deterministic and probabilistic systems. In *Validation of Stochastic Systems*, pages 419–444, 2004.
13. M. Huth, R. Jagadeesan, and D. Schmidt. Modal transition systems: A foundation for three-valued program analysis. In *European Symposium on Programming (ESOP)*, volume 2028, pages 155–169, 2001.
14. Michael Huth. Abstraction and probabilities for hybrid logics. In *Qualitative Aspects of Programming Languages*, 2004.
15. B. Jonsson and K. Larsen. Specification and refinement of probabilistic processes. In *Proc. 6th IEEE Int. Symp. on Logic in Computer Science*, 1991.
16. B. Konikowska and W. Penczek. Model checking for multi-valued computation tree logics. In *Beyond two: theory and applications of multiple-valued logic*, pages 193–210. Physica-Verlag GmbH, 2003.
17. B. Konikowska and W. Penczek. On designated values in multi-valued CTL* model checking. *Fundamenta Informaticae*, 60(1–4):221–224, 2004.
18. S. Shoham and O. Grumberg. A game-based framework for CTL counterexamplesand 3-valued abstraction-refinemnet. In *Computer Aided Verification (CAV)*, volume 2725 of *LNCS*, pages 275–287, 2003.
19. R. van Glabbeek, S. Smolka, B. Steffen, and C. Tofts. Reactive, generative, and stratified models of probabilistic processes. In *Logic in Computer Science*, pages 130–141, 1990.
20. W. Yi. Reasoning about uncertain information compositionally. In *Proc. of the 3rd International School and Symposium on Real-Time and Fault-Tolerant Systems*, volume 863 of *LNCS*. Springer, 1994.