Computing Conditional Probabilities in Markovian Models Efficiently*

Christel Baier, Joachim Klein, Sascha Klüppelholz, and Steffen Märcker

Institute for Theoretical Computer Science
Technische Universität Dresden, Germany

Abstract. The fundamentals of probabilistic model checking for Markovian models and temporal properties have been studied extensively in the past 20 years. Research on methods for computing conditional probabilities for temporal properties under temporal conditions is, however, comparably rare. For computing conditional probabilities or expected values under ω-regular conditions in Markov chains, we introduce a new transformation of Markov chains that incorporates the effect of the condition into the model. For Markov decision processes, we show that the task to compute maximal reachability probabilities under reachability conditions is solvable in polynomial time, while it was conjectured to be computationally hard. Using adaptions of known automata-based methods, our algorithm can be generalized for computing the maximal conditional probabilities for ω-regular events under ω-regular conditions. The feasibility of our algorithms is studied in two benchmark examples.

1 Introduction

Probabilistic model checking has become a prominent technique for the quantitative analysis of systems with stochastic phenomena. Tools like PRISM [20] or MRMC [18] provide powerful probabilistic model checking engines for Markovian models and temporal logics such as probabilistic computation tree logic (PCTL) for discrete models and its continuous-time counterpart CSL (continuous stochastic logic) or linear temporal logic (LTL) as formalism to specify complex path properties. The core task for the quantitative analysis is to compute the probability of some temporal path property or the expected value of some random variable. For finite-state Markovian model with discrete probabilities, this task is solvable by a combination of graph algorithms, matrix-vector operations and methods for solving linear equation systems or linear programming techniques [25,9,15,7]. Although probabilistic model checking is a very active research topic and many researchers have suggested sophisticated methods e.g. to tackle the state explosion problem or to provide algorithms for the analysis of infinite-state

* This work was in part funded by the DFG through the CRC 912 HAEC, the cluster of excellence cfAED, the project QuaOS, the DFG/NWO-project ROCKS, and by the ESF young researcher group IMData 100098198, and the EU-FP-7 grant 295261 (MEALS).

E. Ábrahám and K. Havelund (Eds.): TACAS 2014, LNCS 8413, pp. 515–530, 2014.

stochastic models or probabilistic games, there are important classes of properties that are not directly supported by existing probabilistic model checkers. Among these are *conditional probabilities* that are well-known in probability theory and statistics, but have been neglected by the probabilistic model checking community. Exceptions are [1,2] where PCTL has been extended by a conditional probability operator and recent approaches for discrete and continuous-time Markov chains and patterns of path properties with multiple time- and cost-bounds [13,17].

The usefulness of conditional probabilities for anonymization protocols has been illustrated in [1,2]. Let us provide here some more intuitive examples that motivate the study of conditional probabilities. For systems with unreliable components one might ask for the conditional probability to complete a task successfully within a given deadline, under the condition that no failure will occur that prevents the completion of the task. If multiple tasks $\theta_1, \ldots, \theta_k$ have to be completed, assertions on the conditional probability or the conditional costs to complete task θ_i, under the condition that some other task θ_j will be completed successfully might give important insights on how to schedule the tasks without violating some service-level agreements. For another example, the conditional expected energy requirements for completing a task, under the condition that a certain utility value can be guaranteed, can provide useful insights for the design of power management algorithms. Conditional probabilities can also be useful for assume-guarantee-style reasoning. In these cases, assumptions on the stimuli of the environment can be formalized by a path property ψ and one might then reason about the quantitative system behavior using the conditional probability measure under the condition that ψ holds.

Given a purely stochastic system model $\mathcal{M}$ (e.g. a Markov chain), the analysis under conditional probability measures can be carried out using standard methods for unconditional probabilities, as we can simply rely on the mathematical definition of the conditional probability for φ (called here the *objective*) under condition ψ:

$$\mathrm{Pr}^{\mathcal{M}}_{s}(\,\varphi\,|\,\psi\,) \quad = \quad \frac{\mathrm{Pr}^{\mathcal{M}}_{s}(\varphi \wedge \psi)}{\mathrm{Pr}^{\mathcal{M}}_{s}(\psi)}$$

where s is a state in $\mathcal{M}$ with $\mathrm{Pr}^{\mathcal{M}}_{s}(\psi) > 0$. If both the objective φ and the condition ψ are ω-regular path properties, e.g. specified by LTL formulas or some ω-automaton, then $\varphi \wedge \psi$ is again ω-regular, and the above quotient is computable with standard techniques. This approach has been taken by Andrés and van Rossum [1,2] for the case of discrete Markov chains and PCTL path formulas, where $\varphi \wedge \psi$ is not a PCTL formula, but a ω-regular property of some simple type if nested state formulas are viewed as atoms. Recently, an automata-based approach has been developed for continuous-time Markov chains and CSL path formulas built by cascades of the until-operator with time- and cost-bounds [13]. This approach has been adapted in [17] for discrete-time Markov chains and PCTL-like path formulas with multiple bounded until-operators.

For models that support both the representation of nondeterministic and probabilistic behaviors, such as Markov decision processes (MDPs), reasoning about (conditional) probabilities requires the resolution of the nondeterministic choices by means of *schedulers*. Typically, one is interested in guarantees that can be given even for worst-case scenarios. That is, we are interested in the maximal (or minimal) conditional probability for the objective φ under condition ψ when ranging over all schedulers. Unfortunately, there is no straightforward reduction to unconditional maximal (or minimal) probabilities, simply because extrema of quotients cannot be computed by the quotient of the extremal values of the numerator and the denominator. [1,2] present a model checking algorithm for MDP and PCTL extended by a conditional probability operator. The essential features are algorithms for computing the maximal or minimal conditional probabilities for the case where both the objective and the condition are given as PCTL path formulas. These algorithms rely on transformations of the given MDP into an acyclic one and the fact that for PCTL objectives and conditions optimal schedulers that are composed by two memoryless schedulers (so-called *semi history-independent* schedulers) exist. The rough idea is to consider all semi history-independent schedulers and compute the conditional probabilities for them directly. This method suffers from the combinatorial blow-up and leads to an exponential-time algorithm. [1,2] also present reduction and bounding heuristics to omit some semi history-independent schedulers, but these cannot avoid the exponential worst-case time complexity. We are not aware of an implementation of these methods.

Contribution. The theoretical main contribution is twofold. First, for discrete Markov chains we present an alternative approach that relies on a transformation where we switch from the original Markov chain $\mathcal{M}$ to a modified Markov chain $\mathcal{M}_\psi$ such that the conditional probabilities in $\mathcal{M}$ agree with the (unconditional) probabilities in $\mathcal{M}_\psi$ for all measurable path properties φ. That is, $\mathcal{M}_\psi$ only depends on the condition ψ, but not on the objective φ. Second, for MDPs we provide a polynomial-time algorithm for computing maximal conditional probabilities when both the objective φ and the condition ψ are reachability properties. (This task was suspected to be computationally hard in [2].) Moreover, we show that adaptions of known automata-based approaches are applicable to extend this method for ω-regular objectives and conditions. In both cases, the time complexity of our methods is roughly the same as for computing (extremal) unconditional probabilities for properties of the form $\varphi \wedge \psi$.

Outline. Section 2 summarizes the relevant concepts of Markov chains and MDPs. The theoretical foundations of our approach will be presented for Markov chains in Sections 3 for MDPs in Section 4. Section 5 reports on experimental results. Section 6 contains some concluding remarks. Omitted proofs and other additional material can be found in the technical report [6].

2 Preliminaries

We briefly summarize our notations used for Markov chains and Markov decision processes. Further details can be found in textbooks on probability theory and Markovian models, see e.g. [24,19,16].

Markov Chains. A Markov chain is a pair $\mathcal{M} = (S, P)$ where S is a countable set of states and $p : s \times s \to [0, 1]$ a function, called the transition probability function, such that $\sum_{s' \in S} P(s, s') = 1$ for each state s. Paths in $\mathcal{M}$ are finite or infinite sequences $s_0\, s_1\, s_2 \dots$ of states built by transitions, i.e., $P(s_{i-1}, s_i) > 0$ for all $i \geq 1$. If $\pi = s_0\, s_1 \dots s_n$ is a finite path then $\mathit{first}(\pi) = s_0$ denotes the first state of π, and $\mathit{last}(\pi) = s_n$ the last state of π. The notation $\mathit{first}(\pi)$ will be used also for infinite paths. We refer to the value

$$\Pr(\pi) \;\; = \prod_{1 \leq i \leq n} P(s_{i-1}, s_i)$$

as the probability for π. The cylinder set $\mathit{Cyl}(\pi)$ is the set of all infinite paths ς where π is a prefix of ς. We write $\mathit{FPaths}(s)$ for the set of all finite paths π with $\mathit{first}(\pi) = s$. Similarly, $\mathit{Paths}(s)$ stands for the set of infinite paths starting in s.

Given a state s, the probability space induced by $\mathcal{M}$ and s is defined using classical measure-theoretic concepts. The underlying sigma-algebra is generated by the cylinder sets of finite paths. This sigma-algebra does not depend on s. We refer to the elements of this sigma-algebra as *(measurable) path events*. The probability measure $\Pr_s^{\mathcal{M}}$ is defined on the basis of standard measure extension theorems that yield the existence of a probability measure $\Pr_s^{\mathcal{M}}$ with $\Pr_s^{\mathcal{M}}\big(\mathit{Cyl}(\pi)\big) = \Pr(\pi)$ for all $\pi \in \mathit{FPaths}(s)$, while the cylinder sets of paths π with $\mathit{first}(\pi) \neq s$ have measure 0 under $\Pr_s^{\mathcal{M}}$.

Markov Decision Processes (MDPs). MDPs can be seen as a generalization of Markov chains where the operational behavior in a state s consists of a nondeterministic selection of an enabled action α, followed by a probabilistic choice of the successor state, given s and α. Formally, an MDP is a tuple $\mathcal{M} = (S, \mathit{Act}, P)$ where S is a finite set of states, Act a finite set of actions and $P : S \times \mathit{Act} \times S \to [0, 1]$ a function such that for all states $s \in S$ and $\alpha \in \mathit{Act}$:

$$\sum_{s' \in S} P(s, \alpha, s') \in \{0, 1\}$$

We write $\mathit{Act}(s)$ for the set of actions that are enabled in s, i.e., $P(s, \alpha, s') > 0$ for some $s' \in S$. For technical reasons, we require that $\mathit{Act}(s) \neq \varnothing$ for all states s. State s is said to be *probabilistic* if $\mathit{Act}(s) = \{\alpha\}$ is a singleton, in which case we also write $P(s, s')$ rather than $P(s, \alpha, s')$. A *trap* state is a probabilistic state s with $P(s, s) = 1$. *Paths* are finite or infinite sequences $s_0\, s_1\, s_2 \dots$ of states such that for all $i \geq 1$ there exists an action α_i with $P(s_{i-1}, \alpha_i, s_i) > 0$. (For our purposes, the actions are irrelevant in paths.) Several notations that have been introduced for Markov chains can now be adapted for Markov decision processes, such as $\mathit{first}(\pi)$, $\mathit{FPaths}(s)$, $\mathit{Paths}(s)$.

Reasoning about probabilities for path properties in MDPs requires the selection of an initial state and the resolution of the nondeterministic choices between the possible transitions. The latter is formalized via *schedulers*, often also called policies or adversaries, which take as input a finite path and select an action to be executed. For the purposes of this paper, it suffices to consider deterministic, possibly history-dependent schedulers, i.e., partial functions $\mathfrak{S} : \mathit{FPaths} \to \mathit{Act}$ such that $\mathfrak{S}(\pi) \in \mathit{Act}\big(\mathit{last}(\pi)\big)$ for all finite paths π. Given a scheduler $\mathfrak{S}$, an $\mathfrak{S}$-*path* is any path that might arise when the nondeterministic choices in $\mathcal{M}$ are resolved using $\mathfrak{S}$. Thus, $\pi = s_0\, s_1 \ldots s_n$ is an $\mathfrak{S}$-path iff $P\big(s_{k-1}, \mathfrak{S}(s_0\, s_1 \ldots s_{k-1}), s_k\big) > 0$ for all $1 \leq k \leq n$. In this case, $\mathfrak{S}[\pi]$ denotes the scheduler "$\mathfrak{S}$ after π" given by $\mathfrak{S}[\pi](t_0\, t_1 \ldots t_k) = \mathfrak{S}(s_0\, s_1 \ldots s_n\, t_1 \ldots t_k)$ if $s_n = t_0$. The behavior of $\mathfrak{S}[\pi]$ for paths not starting in s_n is irrelevant. The probability of π under $\mathfrak{S}$ is the product of the probabilities of its transitions:

$$\Pr^{\mathfrak{S}}(\pi) \quad = \quad \prod_{i=0}^{n-1} P\big(s_{k-1}, \mathfrak{S}(s_0\, s_1 \ldots s_{k-1}), s_k\big)$$

Infinite $\mathfrak{S}$-paths are defined accordingly.

For a pointed MDP $(\mathcal{M}, s_{\mathit{init}})$, i.e. an MDP as before with some distinguished initial state $s_{\mathit{init}} \in S$, the behavior of $(\mathcal{M}, s_{\mathit{init}})$ under $\mathfrak{S}$ is purely probabilistic and can be formalized by an infinite tree-like Markov chain $\mathcal{M}_s^{\mathfrak{S}}$ where the states are the finite $\mathfrak{S}$-paths starting in s. The probability measure $\Pr^{\mathfrak{S}}_{\mathcal{M},s}$ for measurable sets of the infinite paths in the Markov chain $\mathcal{M}_s^{\mathfrak{S}}$, can be transferred to infinite $\mathfrak{S}$-paths in $\mathcal{M}$ starting in s. Thus, if $\varPhi$ is a path event then $\Pr^{\mathfrak{S}}_{\mathcal{M},s}(\varPhi)$ denotes its probability under scheduler $\mathfrak{S}$ for starting state s. For a worst-case analysis of a system modeled by an MDP $\mathcal{M}$, one ranges over all initial states and all schedulers (i.e., all possible resolutions of the nondeterminism) and considers the maximal or minimal probabilities for $\varPhi$. If $\varPhi$ represents a desired path property, then $\Pr^{\min}_{\mathcal{M},s}(\varPhi) = \inf_{\mathfrak{S}} \Pr^{\mathfrak{S}}_{\mathcal{M},s}(\varPhi)$ is the probability for $\varPhi$ in $\mathcal{M}$ that can be guaranteed even for the worst-case scenarios. Similarly, if $\varPhi$ stands for a bad (undesired) path event, then $\Pr^{\max}_{\mathcal{M},s}(\varPhi) = \sup_{\mathfrak{S}} \Pr^{\mathfrak{S}}_{\mathcal{M},s}(\varPhi)$ is the least upper bound that can be guaranteed for the likelihood of $\varPhi$ in $\mathcal{M}$.

Temporal-Logic Notations, Path Properties. Throughout the paper, we suppose that the reader is familiar with ω-automata and temporal logics. See e.g. [8,14,5]. We often use LTL- and CTL-like notations and identify LTL-formulas with the set of infinite words over the alphabet 2^{AP} that are models for the formulas, where AP denotes the underlying set of atomic propositions. For the Markov chain or MDP $\mathcal{M}$ under consideration we suppose then that they are extended by a labeling function $L : S \to 2^{\mathsf{AP}}$, with the intuitive meaning that precisely the atomic propositions in $L(s)$ hold for state s. At several places, we will use temporal state and path formulas where single states or sets of states in $\mathcal{M}$ are used as atomic propositions with the obvious meaning. Similarly, if $\mathcal{M}$ arises by some product construction, (sets of) local states will be treated as atomic propositions. For the interpretation of LTL- or CTL-like formulas in $\mathcal{M}$, the probability annotations (as well as the action labels in case of an MDP) are ignored and $\mathcal{M}$ is viewed as an ordinary Kripke structure.

By a *path property* we mean any language consisting of infinite words over 2^{AP}. Having in mind temporal logical specifications, we use the logical operators $\vee$, $\wedge$, $\neg$ for union, intersection and complementation of path properties. A path property $\varPhi$ is said to be measurable if the set of infinite paths π in $\mathcal{M}$ satisfying $\varPhi$ is a path event, i.e., an element of the induced sigma-algebra. Indeed, all

ω-regular path properties are measurable [25]. We abuse notations and identify measurable path properties and the induced path event. Thus,

$$\mathrm{Pr}^{\mathfrak{S}}_{\mathcal{M},s}(\varphi) \quad = \quad \mathrm{Pr}^{\mathfrak{S}}_{\mathcal{M},s}\left(\left\{ \pi \in \mathit{Paths}(s) \,:\, \pi \models \varphi \right\}\right)$$

denotes the probability for φ under scheduler $\mathfrak{S}$ and starting state s.

Assumptions. For the methods proposed in the following sections, we suppose that the state space of the given Markov chain and the MDP is finite and that all transition probabilities are rational.

3 Conditional Probabilities in Markov Chains

In what follows, let $\mathcal{M} = (S, P)$ be a finite Markov chain as in Section 2 and ψ an ω-regular condition. We present a transformation $\mathcal{M} \rightsquigarrow \mathcal{M}_{\psi}$ such that the conditional probabilities $\mathrm{Pr}^{\mathcal{M}}_{s}(\,\varphi \,|\, \psi\,)$ agree with the (unconditional) probabilities $\mathrm{Pr}^{\mathcal{M}_\psi}_{s_\psi}(\,\varphi\,)$ for all ω-regular objectives φ. Here, s is a state in $\mathcal{M}$ with $\mathrm{Pr}^{\mathcal{M}}_{s}(\psi) > 0$ and s_ψ the "corresponding" state in $\mathcal{M}_\psi$. We first treat the case where ψ is a reachability condition and then explain a generalization for ω-regular conditions.

Reachability Condition. Let $G \subseteq S$ be a set of goal states and $\psi = \Diamond G$. Intuitively, the Markov chain $\mathcal{M}_\psi$ arises from $\mathcal{M}$ by a monitoring technique that runs in parallel to $\mathcal{M}$ and operates in two modes. In the initial mode "before or at G", briefly called *before mode*, the attempt is to reach G by avoiding all states s with $s \not\models \exists\Diamond G$. The transition probabilities for the states in before mode are modified accordingly. As soon as G has been reached, $\mathcal{M}_\psi$ switches to the *normal mode* where $\mathcal{M}_\psi$ behaves as $\mathcal{M}$. In what follows, we write s^{bef} and s^{nor} for the copies of state s in the before and normal mode, respectively. For $V \subseteq S$, let $V^{\mathit{bef}} = \left\{ s^{\mathit{bef}} : s \in V, s \models \exists\Diamond G \right\}$ the set of V-states where $\mathrm{Pr}^{\mathcal{M}}_{s}(\Diamond G)$ is positive and $V^{\mathit{nor}} = \left\{ s^{\mathit{nor}} : s \in V \right\}$. The Markov chain $\mathcal{M}_\psi = (S_\psi, P_\psi)$ is defined as follows. The state space of $\mathcal{M}_\psi$ is $S_\psi = S^{\mathit{bef}} \cup S^{\mathit{nor}}$. For $s \in S \setminus G$ and $v \in S$ with $s \models \exists\Diamond G$ and $v \models \exists\Diamond G$:

$$P_\psi(s^{\mathit{bef}}, v^{\mathit{bef}}) \quad = \quad P(s,v) \cdot \frac{\mathrm{Pr}^{\mathcal{M}}_{v}(\Diamond G)}{\mathrm{Pr}^{\mathcal{M}}_{s}(\Diamond G)}$$

For $s \in G$, we define $P_\psi(s^{\mathit{bef}}, v^{\mathit{nor}}) = P(s,v)$, modeling the switch from before to normal mode. For the states in normal mode, the transition probabilities are given by $P_\psi(s^{\mathit{nor}}, v^{\mathit{nor}}) = P(s,v)$. In all other cases, $P_\psi(\cdot) = 0$. For the labeling with atomic propositions, we suppose that each state s in $\mathcal{M}$ and its copies s^{bef} and s^{nor} in $\mathcal{M}_\psi$ satisfy the same atomic propositions.

By applying standard arguments for finite Markov chains we obtain that $\mathrm{Pr}^{\mathcal{M}_\psi}_{s^{\mathit{bef}}}(\Diamond G^{\mathit{bef}}) = 1$ for all states s in $\mathcal{M}$ with $s \models \exists\Diamond G$. (This is a simple consequence of the fact that all states in S^{bef} can reach G^{bef}.) Thus, up to the switch from G to G^{bef}, the condition $\Diamond G$ (which we impose for $\mathcal{M}$) holds almost surely for $\mathcal{M}_\psi$. For each path property φ, there is a one-to-one correspondence between the infinite paths π in $\mathcal{M}$ with $\pi \models \varphi \wedge \Diamond G$ and the infinite paths π_ψ in $\mathcal{M}_\psi$ with $\pi_\psi \models \varphi$. More precisely, each path π_ψ in $\mathcal{M}_\psi$ induces a path $\pi_\psi|_{\mathcal{M}}$ in $\mathcal{M}$ by dropping the mode annotations. Vice versa, each path π in $\mathcal{M}$ can be augmented with mode annotations to obtain a path π_ψ in $\mathcal{M}_\psi$ with $\pi_\psi|_{\mathcal{M}} = \pi$, provided that π either contains some G-state or consists of states s with $s \models \exists\Diamond G$. This yields a one-to-one correspondence between the cylinder sets in $\mathcal{M}_\psi$ and the cylinder sets spanned by finite paths of $\mathcal{M}$ that never enter some state s with $s \not\models \exists\Diamond G$ without having visited G before.

Theorem 1 (Soundness of the transformation). *If Φ is a path event for $\mathcal{M}$ (i.e., a measurable set of infinite paths) then $\Phi|_{\mathcal{M}} = \big\{\pi_\psi : \pi \in \Phi, \pi \models \Diamond G \vee \Box\exists\Diamond G\big\}$ is measurable in $\mathcal{M}_\psi$. Moreover, for each s of $\mathcal{M}$ with $s \models \exists\Diamond G$:*

$$\mathrm{Pr}^{\mathcal{M}}_{s}\big(\Phi \,|\, \Diamond G\big) \;=\; \mathrm{Pr}^{\mathcal{M}_\psi}_{s^{bef}}\big(\Phi|_{\mathcal{M}}\big)$$

Hence, $\mathrm{Pr}^{\mathcal{M}}_{s}\big(\varphi \,|\, \Diamond G\big) = \mathrm{Pr}^{\mathcal{M}_\psi}_{s^{bef}}\big(\varphi\big)$ for all measurable path properties φ.

Thus, once $\mathcal{M}_\psi$ has been constructed, conditional probabilities for arbitrary path properties in $\mathcal{M}$ can be computed by standard methods for computing unconditional probabilities in $\mathcal{M}_\psi$, with the same asymptotic costs. (The size of $\mathcal{M}_\psi$ is linear in the size of $\mathcal{M}$.) $\mathcal{M}_\psi$ can be constructed in time polynomial in the size of $\mathcal{M}$ as the costs are dominated by the computation of the reachability probabilities $\mathrm{Pr}^{\mathcal{M}}_{s}(\Diamond G)$. $\mathcal{M}_\psi$ can also be used to reason about the conditional expected value of a random function f on infinite paths in $\mathcal{M}$, as we have:

$$\mathsf{E}^{\mathcal{M}}\big(\, f \,|\, \Diamond G\big) \;=\; \mathsf{E}^{\mathcal{M}_\psi}\big(\, f_\psi \,|\, \Diamond(G^{bef} \cup G^{nor})\big)$$

where $f_\psi(\pi') = f\big(\pi'\big|_{\mathcal{M}}\big)$ and $\mathsf{E}^{\mathcal{N}}(\cdot)$ denotes the expected-value operator in $\mathcal{N}$. An important instance is expected accumulated rewards to reach a certain set of states. See [6].

$\boldsymbol{\omega}$-regular conditions. Suppose now that the condition ψ is given by a deterministic ω-automaton $\mathcal{A}$ with, e.g., Rabin or Streett acceptance. To construct a Markov chain that incorporates the probabilities in $\mathcal{M}$ under the condition ψ, we rely on the standard techniques for the quantitative analysis of Markov chains against automata-specifications [26,5]. The details are straightforward, we just give a brief outline. First, we build the standard product $\mathcal{M} \otimes \mathcal{A}$ of $\mathcal{M}$ and $\mathcal{A}$, which is again a Markov chain. Let G be the union of the bottom strongly connected components C of $\mathcal{N} = \mathcal{M} \otimes \mathcal{A}$ that meet the acceptance condition of $\mathcal{A}$. Then, the probability $\mathrm{Pr}^{\mathcal{M}}_{s}(\psi)$ equals $\mathrm{Pr}^{\mathcal{N}}_{\langle s,q_s\rangle}(\Diamond G)$, where $\langle s, q_s\rangle$ is the state in $\mathcal{M} \otimes \mathcal{A}$ that "corresponds" to s. We then apply the transformation for $\mathcal{N} \rightsquigarrow \mathcal{N}_\psi$ as explained above and obtain that for all measurable path properties φ:

$$\mathrm{Pr}^{\mathcal{M}}_{s}\big(\varphi \,|\, \psi\big) \;=\; \mathrm{Pr}^{\mathcal{N}}_{\langle s,q_s\rangle}\big(\varphi \,|\, \Diamond G\big) \;=\; \mathrm{Pr}^{\mathcal{N}_\psi}_{\langle s,q_s\rangle}(\varphi)$$

for all states s in $\mathcal{M}$ where $\mathrm{Pr}^{\mathcal{M}}_{s}(\psi)$ is positive. This shows that the task to compute conditional probabilities for ω-regular conditions is solvable by algorithms for computing (unconditional) probabilities for ω-regular path properties.

4 Conditional Probabilities in Markov Decision Processes

We now consider the task to compute maximal conditional probabilities in MDPs. We start with the "core problem" where the objective and the condition are reachability properties. The general case of ω-regular objectives and conditions will be treated in Section 4.2.

4.1 Conditional Reachability Probabilities in MDPs

Let $(\mathcal{M}, s_{\mathit{init}})$ be a pointed MDP where $\mathcal{M} = (S, \mathit{Act}, P)$ and let F, $G \subseteq S$ such that $s_{\mathit{init}} \models \exists\Diamond G$, in which case $\Pr^{\max}_{\mathcal{M},s_{\mathit{init}}}(\Diamond G) > 0$. The task is to compute

$$\max_{\mathfrak{S}} \Pr^{\mathfrak{S}}_{\mathcal{M},s_{\mathit{init}}}\big(\Diamond F \,|\, \Diamond G\big) \quad = \quad \max_{\mathfrak{S}} \frac{\Pr^{\mathfrak{S}}_{\mathcal{M},s_{\mathit{init}}}\big(\Diamond F \wedge \Diamond G\big)}{\Pr^{\mathfrak{S}}_{\mathcal{M},s_{\mathit{init}}}\big(\Diamond G\big)}$$

where $\mathfrak{S}$ ranges over all schedulers for $\mathcal{M}$ such that $\Pr^{\mathfrak{S}}_{\mathcal{M},s_{\mathit{init}}}(\Diamond G) > 0$. By the results of [1,2], there exists a scheduler $\mathfrak{S}$ maximizing the conditional probability for $\Diamond F$, given $\Diamond G$. (This justifies the use of max rather than sup.)

Only for simplicity, we assume that $F \cap G = \varnothing$. Thus, there are just two cases for the event $\Diamond F \wedge \Diamond G$: "either F before G, or G before F". We also suppose $s_{\mathit{init}} \notin F \cup G$ and that all states $s \in S$ are accessible from s_{init}.

Step 1: Normal form Transformation

We first present a transformation $\mathcal{M} \rightsquigarrow \mathcal{M}'$ such that the maximal conditional probability for "$\Diamond F$, given $\Diamond G$" agrees with the maximal conditional probability for "$\Diamond F'$, given $\Diamond G'$" in $\mathcal{M}'$ where F' and G' consist of trap states. This can be seen as some kind of normal form for maximal conditional reachability probabilities and relies on the following observation.

Lemma 1 (Scheduler improvement). *For each scheduler $\mathfrak{S}$ there is a scheduler $\mathfrak{T}$ such that for all states s with $\Pr^{\mathfrak{S}}_{\mathcal{M},s}(\Diamond G) > 0$:*

(1) $\Pr^{\mathfrak{S}}_{\mathcal{M},s}\big(\Diamond F \,|\, \Diamond G\big) \;\leq\; \Pr^{\mathfrak{T}}_{\mathcal{M},s}\big(\Diamond F \,|\, \Diamond G\big)$

(2) $\Pr^{\mathfrak{T}[\pi]}_{\mathcal{M},t}(\Diamond G) = \Pr^{\max}_{\mathcal{M},t}(\Diamond G)$ *for all $t \in F$ and $\pi \in \mathit{\Pi}_{s\ldots t}$*

(3) $\Pr^{\mathfrak{T}[\pi]}_{\mathcal{M},u}(\Diamond F) = \Pr^{\max}_{\mathcal{M},u}(\Diamond F)$ *for all $u \in G$ and finite paths $\pi \in \mathit{\Pi}_{s\ldots u}$*

where $\mathit{\Pi}_{s\ldots u}$ denotes the set consisting of all finite paths $s_0\, s_1 \ldots s_n$ in $\mathcal{M}$ with $s_0 = s$, $s_n = u$ and $\{s_0, s_1, \ldots, s_{n-1}\} \cap (F \cup G) = \varnothing$.

Recall that $\mathfrak{S}[\pi]$ denotes the scheduler "$\mathfrak{S}$ after π". The idea is that $\mathfrak{T}$ behaves as $\mathfrak{S}$ as long as neither F nor G has been reached. As soon as a G-state (resp. F-state) has been entered, $\mathfrak{T}$ mimics some scheduler that maximizes the probability to reach F (resp. G). This scheduler satisfies (2) and (3) by construction. Item (1) follows after some calculations (see [6]).

As a consequence of Lemma 1, for studying the maximal conditional probability for $\Diamond F$ given $\Diamond G$, it suffices to consider schedulers $\mathfrak{T}$ satisfying conditions (2) and (3). Let $\mathcal{M}'$ be the MDP that behaves as $\mathcal{M}$ as long as no state in F or G has been visited. After visiting an F-state t, $\mathcal{M}'$ moves probabilistically to a fresh goal state with probability $\mathrm{Pr}^{\max}_{\mathcal{M},t}(\Diamond G)$ or to a fail state with the remaining probability. Similarly, after visiting a G-state u, $\mathcal{M}'$ moves probabilistically to the goal state or to a new state *stop*. Formally, $\mathcal{M}' = (S', \mathit{Act}, P')$ where the state space of $\mathcal{M}'$ is $S' = S \cup T$ and

$$T \;=\; \bigl\{ \mathit{goal}, \mathit{stop}, \mathit{fail} \bigr\}.$$

The transition probabilities in $\mathcal{M}'$ for the states in $S \setminus (F \cup G)$ agree with those in $\mathcal{M}$, i.e., $P'(s,\alpha,s') = P(s,\alpha,s')$ for all $s \in S \setminus (F \cup G)$, $s' \in S$ and $\alpha \in \mathit{Act}$. The states $t \in F$ and $u \in G$ are probabilistic in $\mathcal{M}'$ with the transition probabilities:

$$\begin{array}{ll} P'(t,\mathit{goal}) = \mathrm{Pr}^{\max}_{\mathcal{M},t}(\Diamond G) & \qquad P'(u,\mathit{goal}) = \mathrm{Pr}^{\max}_{\mathcal{M},u}(\Diamond F) \\ P'(t,\mathit{fail}) \;= 1 - \mathrm{Pr}^{\max}_{\mathcal{M},t}(\Diamond G) & \qquad P'(u,\mathit{stop}) = 1 - \mathrm{Pr}^{\max}_{\mathcal{M},u}(\Diamond F) \end{array}$$

The three fresh states *goal*, *fail* and *stop* are trap states. Then, by Lemma 1:

Corollary 1 (Soundness of the normal form transformation). *For all states s in $\mathcal{M}$ with $s \models \exists \Diamond G$:*

$$\mathrm{Pr}^{\max}_{\mathcal{M},s}\bigl(\Diamond F \,|\, \Diamond G\bigr) \;= \mathrm{Pr}^{\max}_{\mathcal{M}',s}\bigl(\,\Diamond \mathit{goal}\,\big|\,\Diamond(\mathit{goal} \vee \mathit{stop})\,\bigr)$$

Optional simplification of $\mathcal{M}'$. Let W be the set of states w in $\mathcal{M}'$ such that for some scheduler $\mathfrak{S}$, the goal-state is reachable from w via some $\mathfrak{S}$-path, while the trap state *stop* will not be reached along $\mathfrak{S}$-paths from w. Then, all states in W can be made probabilistic with successors *goal* and *fail* and efficiently computable transition probabilities. This transformation of $\mathcal{M}'$ might yield a reduction of the reachable states, while preserving the maximal conditional probabilities for $\Diamond \mathit{goal}$, given $\Diamond(\mathit{goal} \vee \mathit{stop})$. For details, see [6].

Step 2: Reduction to Ordinary Maximal Reachability Probabilities

We now apply a further transformation $\mathcal{M}' \rightsquigarrow \mathcal{M}_{\varphi|\psi}$ such that the maximal conditional probability for $\varphi = \Diamond \mathit{goal}$, given $\psi = \Diamond(\mathit{goal} \vee \mathit{stop})$, in $\mathcal{M}'$ agrees with the maximal (unconditional) probability for $\Diamond \mathit{goal}$ in $\mathcal{M}_{\varphi|\psi}$.

Let us first sketch the ratio of this transformation. Infinite paths in $\mathcal{M}'$ that violate the condition ψ do not "contribute" to the conditional probability for $\Diamond \mathit{goal}$. The idea is now to "redistribute" their probabilities to all the paths satisfying the condition ψ. Speaking roughly, we aim to mimic a stochastic process that generates a sequence $\pi_0, \pi_1, \pi_2 \ldots$ of sample paths in $\mathcal{M}'$ starting in s_{init} until a path π_i is obtained where the condition ψ holds. To formalize this "redistribution procedure" by switching from $\mathcal{M}'$ to some new MDP $\mathcal{M}_{\varphi|\psi}$ we need some *restart mechanism* to discard generated prefixes of paths π_j violating ψ by returning to the initial state s_{init}, from which the next sample run π_{j+1} will be generated. Note that by discarding paths that do not satisfy ψ, the proportion of the paths satisfying $\varphi \wedge \psi$ and the paths satisfying ψ is not affected and almost surely a path satisfying ψ will be generated. Thus, the conditional probability for $\varphi \wedge \psi$ given ψ under some scheduler of the original MDP agrees with the (unconditional) probability for φ under the corresponding scheduler of the new MDP $\mathcal{M}_{\varphi|\psi}$.

The restart policy is obvious for finite paths that enter the trap state *fail*. Instead of staying in *fail*, we simply restart the computation by returning to the initial state s_{init}. The second possibility to violate ψ are paths that never enter one of the three trap states in T. To treat such paths we rely on well-known results for finite-state MDPs stating that for all schedulers $\mathfrak{S}$ almost all $\mathfrak{S}$-paths eventually enter an end component (i.e., a strongly connected sub-MDP), stay there forever and visit all its states infinitely often [11,12]. The idea is that we equip all states s that belong to some end component without any T-state with the restart-option, i.e., we add the nondeterministic alternative to return to the initial state s_{init}. To enforce that such end components will be left eventually by taking the restart-transition, one might impose strong fairness conditions for the schedulers in $\mathcal{M}_{\varphi|\psi}$. Such fairness assumptions are, however, irrelevant for maximal reachability conditions [3,4].

Let B be the set of (bad) states v such that there exists a scheduler $\mathfrak{S}$ that never visits one of the three trap states *goal*, *stop* or *fail* when starting in v:

$$v \in B \quad \text{iff} \quad \begin{cases} \text{there exists a scheduler } \mathfrak{S} \\ \text{such that } \Pr^{\mathfrak{S}}_{\mathcal{M}',v}\big(\Diamond T\big) \;=\; 0 \end{cases}$$

The MDP $\mathcal{M}_{\varphi|\psi} = (S', \mathit{Act} \cup \{\tau\}, P_{\varphi|\psi})$ has the same state space as the normal form MDP $\mathcal{M}'$. Its action set extends the action set of $\mathcal{M}'$ by a fresh action symbol τ for the restart-transitions. For the states $s \in S' \setminus B$ with $s \neq \mathit{fail}$, the new MDP $\mathcal{M}_{\varphi|\psi}$ behaves as $\mathcal{M}'$, i.e., $P_{\varphi|\psi}(s, \alpha, s') = P'(s, \alpha, s')$ for all $s \in S' \setminus (B \cup \{\mathit{fail}\})$, $\alpha \in \mathit{Act}$ and $s' \in S'$. The fresh action τ is not enabled in the states $s \in S' \setminus (B \cup \{\mathit{fail}\})$. For the fail-state, $\mathcal{M}_{\varphi|\psi}$ returns to the initial state, i.e., $P_{\varphi|\psi}(\mathit{fail}, \tau, s_{\mathit{init}}) = 1$ and $P_{\varphi|\psi}(\mathit{fail}, \tau, s') = 0$ for all states $s' \in S' \setminus \{s_{\mathit{init}}\}$. No other action than τ is enabled in *fail*. For the states $v \in B$, $\mathcal{M}_{\varphi|\psi}$ decides nondeterministically to behave as $\mathcal{M}$ or to return to the initial state s_{init}. That is, if $v \in B$, $\alpha \in \mathit{Act}$, $s' \in S'$ then $P_{\varphi|\psi}(v, \alpha, s') = P'(v, \alpha, s')$ and $P_{\varphi|\psi}(v, \tau, s_{\mathit{init}}) = 1$. In all remaining cases, we have $P_{\varphi|\psi}(v, \tau, \cdot) = 0$.

Paths in $\mathcal{M}'$ that satisfy $\Box B$ or that end up in *fail*, do not "contribute" to the conditional probability for $\Diamond \mathit{goal}$, given $\Diamond(\mathit{goal} \vee \mathit{stop})$. Instead the probability for the infinite paths π with $\pi \in \Box B$ or $\pi \models \Diamond \mathit{fail}$ in $\mathcal{M}'$ are "distributed" to the probabilities for $\Diamond \mathit{goal}$ and $\Diamond \mathit{stop}$ when switching to conditional probabilities. This is mimicked by the restart-transitions to s_{init} in $\mathcal{M}_{\varphi|\psi}$.

Theorem 2 (Soundness of step 2). *For the initial state $s = s_{\mathit{init}}$, we have:*

$$\Pr^{\max}_{\mathcal{M}',s}\big(\Diamond \mathit{goal} \,\big|\, \Diamond(\mathit{goal} \vee \mathit{stop})\big) = \Pr^{\max}_{\mathcal{M}_{\varphi|\psi},s}\big(\Diamond \mathit{goal}\big)$$

Algorithm and Complexity. As an immediate consequence of Theorem 2, the task to compute maximal conditional reachability probabilities in MDPs is reducible to the task to compute maximal ordinary (unconditional) reachability probabilities, which is solvable using linear programming techniques [24,7]. The size of the constructed MDP is linear in the size of $\mathcal{M}'$, which again is linear in the size of $\mathcal{M}$. The construction of $\mathcal{M}'$ and $\mathcal{M}_{\varphi|\psi}$ is straightforward. For $\mathcal{M}'$ we need to compute ordinary maximal reachability probabilities in $\mathcal{M}$. Using standard algorithms for the qualitative fragment of PCTL, the set B of bad states is computable by a graph analysis in polynomial time (see e.g. [5]). Thus, maximal conditional probabilities for reachability objectives and conditions can be computed in time polynomial in the size of $\mathcal{M}$.

4.2 Conditional Probabilities in MDPs for Other Events

Using standard automata-based techniques, our method can be generalized to deal with ω-regular properties for both the objective and the condition.

ω-regular objectives under reachability conditions. Using a standard automata-based approach, the suggested technique is also applicable to compute maximal conditional probabilities $\Pr^{\max}_{\mathcal{M},s}\big(\varphi \,\big|\, \Diamond G\big)$. Here, we deal with a deterministic ω-automaton $\mathcal{A}$ for φ and then compute the maximal conditional probabilities $\Pr^{\max}_{\mathcal{N},\langle s,q_s\rangle}\big(\Diamond F \,\big|\, \Diamond G\big)$ in the product-MDP $\mathcal{N} = \mathcal{M} \otimes \mathcal{A}$ where F is the union of all end components in $\mathcal{M} \otimes \mathcal{A}$ satisfying the acceptance condition of $\mathcal{A}$. Here, $\langle s, q_s\rangle$ denotes the state in $\mathcal{M} \otimes \mathcal{A}$ that “corresponds” to s.

(co-)safety conditions. If ψ is *regular co-safety condition* then we can use a representation of ψ by a deterministic finite automaton (DFA) $\mathcal{B}$, switch from $\mathcal{M}$ to the product-MDP $\mathcal{M} \otimes \mathcal{B}$ with the reachability condition stating that some final state of $\mathcal{B}$ should be visited. With slight modifications, an analogous technique is applicable for *regular safety conditions*, in which case we use a DFA for the bad prefixes of ψ. See [6].This approach is also applicable for MDPs with positive state rewards and if ψ is a reward-bounded reachability condition $\Diamond^{\leq r} a$.

ω-regular conditions. If the condition ψ and the objective φ are ω-regular then the task to compute $\Pr^{\max}_{\mathcal{M},s}\big(\varphi \,|\, \psi\big)$ is reducible to the task of computing maximal conditional probabilities for reachability objectives and some strong fairness condition ψ'. The idea is to simply use deterministic Streett automata $\mathcal{A}$ and $\mathcal{B}$ for φ and ψ and then to switch from $\mathcal{M}$ to the product-MDP $\mathcal{M}\otimes\mathcal{A}\otimes\mathcal{B}$. The condition ψ can then be replaced by $\mathcal{B}$'s acceptance condition. The goal set F of the objective $\Diamond F$ arises by the union of all end components in $\mathcal{M} \otimes \mathcal{A} \otimes \mathcal{B}$ where the acceptance conditions of both $\mathcal{A}$ and $\mathcal{B}$ hold.

It remains to explain how to compute $\Pr^{\max}_{\mathcal{M},s}\big(\varphi \,|\, \psi\big)$ where $\varphi = \Diamond F$ is a reachability objective and ψ is a strong fairness (i.e., Streett) condition, say:

$$\psi \quad = \quad \bigwedge_{1 \leq i \leq k} (\Box\Diamond R_i \rightarrow \Box\Diamond G_i)$$

We can rely on very similar ideas as for reachability conditions (see Section 4.1). The construction of a normal form MDP $\mathcal{M}'$ (step 1) is roughly the same except that we deal only with two fresh trap states: *goal* and *fail*. The restart mechanism in step 2 can be realized by switching from $\mathcal{M}'$ to a new MDP $\mathcal{M}_{\varphi|\psi}$

that is defined in the same way as in Section 4.1, except that restart-transitions are only added to those states v where $v \in R_i$ for some $i \in \{1, \ldots, k\}$, and v is contained in some end component that does not contain *goal* and does not contain any G_i-state. For further details we refer to the extended version [6]).

5 PRISM Implementation and Experiments

We have implemented most of the algorithms proposed in this paper in the popular model checker PRISM [21], extending the functionality of version 4.1. Our implementation is based on the "explicit" engine of PRISM, i.e., the analysis is carried out using an explicit representation of the reachable states and transitions. We have extended the explicit engine to handle LTL path properties for Markov chains using deterministic Rabin automata and PRISM's infrastructure.

For Markov chains, we implemented the presented transformation $\mathcal{M} \rightsquigarrow \mathcal{M}_\psi$ where ψ and φ are given as LTL formulas. The presented method for reachability conditions $\psi = \Diamond G$ has been adapted in our implementation for the more general case of constrained reachability conditions $\psi = H \,\mathrm{U}\, G$. Our implementation also supports a special treatment of conditions ψ consisting of a single step-bounded modality $\Diamond^{\leq n}$, $\mathrm{U}^{\leq n}$ or $\Box^{\leq n}$. Besides the computation of conditional probabilities, our implementation also provides the option to compute conditional expected rewards under (constrained) reachability or ω-regular conditions. We used the three types of expected rewards supported by PRISM: the expected accumulated reward until a target set F is reached or within the next $n \in \mathbb{N}$ steps, and the expected instantaneous reward obtained in the n-th step. For MDPs, our current implementation only supports the computation of maximal conditional probabilities for reachability objectives and reachability conditions based on the algorithm presented in Section 4.1.

Experiments with Markov Chains. To evaluate our transformation-based approach for Markov chains we carried out a series of experiments with the Markov chain model for the bounded retransmissions protocol presented in [10]. The model specifications are from the PRISM benchmark suite [22] (see `http://www.prismmodelchecker.org/casestudies/brp.php`). A sender has to transmit N fragments of a file using a simple protocol over lossy channels, where the probability of losing a message is 0.01, while the probability of losing an acknowledgment is 0.02. A parameter M specifies the maximum number of retries for each fragment. We applied our method to compute:

(B1) $\Pr_s^{\mathcal{M}}\big(\Diamond \text{"second retry for fragment"} \,\big|\, \Box \neg \text{"finish with error"}\big)$

(B2) $\Pr_s^{\mathcal{M}}\big(\Diamond \text{"finish with success"} \,\big|\, \Diamond \text{"2 fragments transmitted"}\big)$

(B3) $\Pr_s^{\mathcal{M}}\big(\Box \neg \text{"retry"} \,\big|\, \Diamond \text{"finish with success"} \wedge \Box \text{"retries} \leq 2\text{"}\big)$

All calculations for this paper were carried out on a computer with 2 Intel E5-2680 8-core CPUs at 2.70 GHz with 384Gb of RAM. Table 1 lists results for the calculation of the conditional probabilities (B1)–(B3), with $N = 128$ fragments and $M = 10$ retries. We report the number of states and the time for building the

Table 1. Statistics for the computation of (B1), (B2), (B3) for $N = 128$, $M = 10$

	model $\mathcal{M}$		$\Pr_s^{\mathcal{M}}(\varphi \mid \psi)$ via transformation				via quotient
	states	build	st. $\mathcal{M}_\psi$	$\mathcal{M} \rightsquigarrow \mathcal{M}_\psi$	calc in $\mathcal{M}_\psi$	total time	total time
(B1)	18,701	0.5 s	17,805	19.2 s	5.5 s	24.7 s	58.7 s
(B2)	18,701	0.5 s	18,679	1.7 s	17.0 s	18.7 s	39.2 s
(B3)	18,701	0.5 s	3,976	10.5 s	1.2 s	11.7 s	14.9 s

model and statistics for the calculation of $\Pr_s^{\mathcal{M}}(\varphi \mid \psi)$ with the method presented in Section 3 and via the quotient of $\Pr_s^{\mathcal{M}}(\varphi \wedge \psi)$ and $\Pr_s^{\mathcal{M}}(\psi)$. In addition to the total time for the calculation, for our method we list as well the size of the transformed model $\mathcal{M}_\psi$, the time spent in the transformation phase and the time spent to calculate the probabilities of φ in $\mathcal{M}_\psi$. In these experiments, our transformation method outperforms the quotient approach by separating the treatment of ψ and φ. As expected, the particular condition significantly influences the size of $\mathcal{M}_\psi$ and the time spent for the calculation in $\mathcal{M}_\psi$. We plan to allow caching of $\mathcal{M}_\psi$ if the task is to treat multiple objectives under the same condition ψ. We have carried out experiments for conditional rewards with similar scalability results as well, see [6].

Experiments with MDPs. We report on experimental studies with our implementation of the calculation of $\Pr_{\mathcal{M},s}^{\max}(\Diamond F \mid \Diamond G)$ for the initial state $s = s_{init}$ of the parameterized MDP presented in [23]; see also [22], `http://www.prismmodelchecker.org/casestudies/wlan.php`. It models a two-way handshake mechanism of the IEEE 802.11 (WLAN) medium access control scheme with two senders S_1 and S_2 that compete for the medium. As messages get corrupted when both senders send at the same time (called a collision), a probabilistic back-off mechanism is employed. The model deals with the case where a single message from S_1 and S_2 should be successfully sent. We consider here:

(W1) $\Pr_{\mathcal{M},s}^{\max}\big(\Diamond \text{“}c_2 \text{ collisions”} \bigm| \Diamond \text{“}c_1 \text{ collisions”}\big)$

(W2) $\Pr_{\mathcal{M},s}^{\max}\big(\Diamond \text{“deadline } t \text{ expired without success of } \mathsf{S}_1\text{”} \bigm| \Diamond \text{“}c \text{ collisions”}\big)$

The parameter N specifies the maximal number of back-offs that each sender performs. The atomic propositions “c collisions” are supported by a global counter variable in the model that counts the collisions (up to the maximal interesting value for the property). For (W2), the deadline t is encoded in the model by a global variable counting down until the deadline is expired.

Calculating (W1). Table 2 lists results for the calculation of (W1) with $c_2 = 4$ and $c_1 = 2$. We report the number of states and the time for building the model. The states in the transformed MDP $\mathcal{M}_{\varphi|\psi}$ consist of the states in the original MDP $\mathcal{M}$ plus the three trap states introduced in the transformation. We list the time for the transformation $\mathcal{M} \rightsquigarrow \mathcal{M}_{\varphi|\psi}$ and for the computation in $\mathcal{M}_{\varphi|\psi}$ separately. For comparison, we list as well the time for calculating the unconditional probabilities $\Pr_{\mathcal{M}}^{\max}(\varphi)$ and $\Pr_{\mathcal{M}}^{\max}(\psi)$ for all states in the

Table 2. Statistics for the calculation of (W1) with $c_1 = 2$ and $c_2 = 4$

	model $\mathcal{M}$		$\mathrm{Pr}^{\max}_{\mathcal{M},s}(\varphi \mid \psi)$			$\mathrm{Pr}^{\max}_{\mathcal{M}}(\varphi)$	$\mathrm{Pr}^{\max}_{\mathcal{M}}(\psi)$
N	states	build	$\mathcal{M} \rightsquigarrow \mathcal{M}_{\varphi \mid \psi}$	calc in $\mathcal{M}_{\varphi \mid \psi}$	total time	total	total
3	118,280	2.3 s	1.6 s	3.2 s	4.8 s	1.1 s	0.4 s
4	345,120	5.5 s	3.2 s	9.0 s	12.3 s	1.6 s	1.3 s
5	1,295,338	21.0 s	12.6 s	33.8 s	46.5 s	3.9 s	4.9 s
6	5,007,668	99.4 s	38.8 s	126.0 s	164.9 s	12.7 s	18.7 s

Table 3. Statistics for the calculation of (W2) with $N = 3$

		model $\mathcal{M}$		$\mathrm{Pr}^{\max}_{\mathcal{M},s}(\varphi \mid \psi)$			$\mathrm{Pr}^{\max}_{\mathcal{M}}(\varphi)$	$\mathrm{Pr}^{\max}_{\mathcal{M}}(\psi)$
t	c	states	build	$\mathcal{M} \rightsquigarrow \mathcal{M}_{\varphi \mid \psi}$	calc in $\mathcal{M}_{\varphi \mid \psi}$	total time	total	total
50	1	539,888	10.0 s	6.4 s	0.4 s	6.8 s	6.0 s	0.1 s
50	2	539,900	9.5 s	7.1 s	4.6 s	11.7 s	6.0 s	0.6 s
100	1	4,769,199	95.1 s	194.6 s	2.4 s	197.1 s	192.0 s	0.5 s
100	2	4,769,235	93.3 s	199.8 s	85.5 s	285.5 s	184.4 s	10.4 s

model, which account for a large part of the transformation. As can be seen, our approach scales reasonably well.

Calculating (W2). Table 3 lists selected results and statistics for (W2) with $N = 3$, deadline $t \in \{50, 100\}$ and number of collisions in the condition $c \in \{1, 2\}$. Again, the time for the transformation is dominated by the computations of $\mathrm{Pr}^{\max}_{\mathcal{M}}(\varphi)$ and $\mathrm{Pr}^{\max}_{\mathcal{M}}(\psi)$. However, in contrast to (W1), the time for the computation in $\mathcal{M}_{\varphi \mid \psi}$ is significantly lower. The complexity in practice thus varies significantly with the particularities of the model and the condition.

6 Conclusion

We presented new methods for the computation of (maximal) conditional probabilities via reductions to the computation of ordinary (maximal) probabilities in discrete Markov chains and MDPs. These methods rely on transformations of the model to encode the effect of conditional probabilities. For MDPs we concentrated on the computation of maximal conditional probabilities. Our techniques are, however, also applicable for reasoning about minimal conditional probabilities as: $\mathrm{Pr}^{\min}_{\mathcal{M},s}\big(\,\varphi \mid \psi\,\big) = 1 - \mathrm{Pr}^{\max}_{\mathcal{M},s}\big(\neg\varphi \mid \psi\,\big)$. By our results, the complexity of the problem that asks whether the (maximal) conditional probabilities meets a given probability bound is not harder than the corresponding question for unconditional probabilities. This is reflected in our experiments. In our experiments with Markov chains, our new method outperforms the naïve approach. In future work, we will extend our implementations for MDPs that currently only supports reachability objectives and conditions and study methods for the computation of maximal or minimal expected conditional accumulated rewards.

References

1. Andrés, M.E., van Rossum, P.: Conditional probabilities over probabilistic and nondeterministic systems. In: Ramakrishnan, C.R., Rehof, J. (eds.) TACAS 2008. LNCS, vol. 4963, pp. 157–172. Springer, Heidelberg (2008)
2. Andrés, M.E.: Quantitative Analysis of Information Leakage in Probabilistic and Nondeterministic Systems. PhD thesis, UB Nijmegen (2011)
3. Baier, C.: On the algorithmic verification of probabilistic systems. Universität Mannheim, Habilitation Thesis (1998)
4. Baier, C., Groesser, M., Ciesinski, F.: Quantitative analysis under fairness constraints. In: Liu, Z., Ravn, A.P. (eds.) ATVA 2009. LNCS, vol. 5799, pp. 135–150. Springer, Heidelberg (2009)
5. Baier, C., Katoen, J.-P.: Principles of Model Checking. MIT Press (2008)
6. Baier, C., Klein, J., Klüppelholz, S., Märcker, S.: Computing conditional probabilities in Markovian models efficiently. Technical report, TU Dresden (2014), `http://wwwtcs.inf.tu-dresden.de/ALGI/PUB/TACAS14/`
7. Bianco, A., De Alfaro, L.: Model checking of probabilistic and non-deterministic systems. In: Thiagarajan, P.S. (ed.) FSTTCS 1995. LNCS, vol. 1026, pp. 499–513. Springer, Heidelberg (1995)
8. Clarke, E., Grumberg, O., Peled, D.: Model Checking. MIT Press (2000)
9. Courcoubetis, C., Yannakakis, M.: The complexity of probabilistic verification. Journal of the ACM 42(4), 857–907 (1995)
10. D'Argenio, P.R., Jeannet, B., Jensen, H.E., Larsen, K.G.: Reachability analysis of probabilistic systems by successive refinements. In: de Alfaro, L., Gilmore, S. (eds.) PAPM-PROBMIV 2001. LNCS, vol. 2165, pp. 39–56. Springer, Heidelberg (2001)
11. de Alfaro, L.: Formal Verification of Probabilistic Systems. PhD thesis, Stanford University, Department of Computer Science (1997)
12. de Alfaro, L.: Computing minimum and maximum reachability times in probabilistic systems. In: Baeten, J.C.M., Mauw, S. (eds.) CONCUR 1999. LNCS, vol. 1664, pp. 66–81. Springer, Heidelberg (1999)
13. Gao, Y., Xu, M., Zhan, N., Zhang, L.: Model checking conditional CSL for continuous-time Markov chains. Information Processing Letters 113(1-2), 44–50 (2013)
14. Grädel, E., Thomas, W., Wilke, T. (eds.): Automata, Logics, and Infinite Games. LNCS, vol. 2500. Springer, Heidelberg (2002)
15. Hansson, H., Jonsson, B.: A logic for reasoning about time and reliability. Formal Aspects of Computing 6, 512–535 (1994)
16. Haverkort, B.: Performance of Computer Communication Systems: A Model-Based Approach. Wiley (1998)
17. Ji, M., Wu, D., Chen, Z.: Verification method of conditional probability based on automaton. Journal of Networks 8(6), 1329–1335 (2013)
18. Katoen, J.-P., Zapreev, I.S., Hahn, E.M., Hermanns, H., Jansen, D.N.: The ins and outs of the probabilistic model checker MRMC. Performance Evaluation 68(2), 90–104 (2011)
19. Kulkarni, V.: Modeling and Analysis of Stochastic Systems. Chapman & Hall (1995)
20. Kwiatkowska, M., Norman, G., Parker, D.: Probabilistic symbolic model checking with PRISM: A hybrid approach. STTT 6(2), 128–142 (2004)
21. Kwiatkowska, M., Norman, G., Parker, D.: PRISM 4.0: Verification of probabilistic real-time systems. In: Gopalakrishnan, G., Qadeer, S. (eds.) CAV 2011. LNCS, vol. 6806, pp. 585–591. Springer, Heidelberg (2011)

22. Kwiatkowska, M., Norman, G., Parker, D.: The PRISM benchmark suite. In: QEST 2012. IEEE (2012)
23. Kwiatkowska, M., Norman, G., Sproston, J.: Probabilistic model checking of the IEEE 802.11 wireless local area network protocol. In: Hermanns, H., Segala, R. (eds.) PAPM-PROBMIV 2002. LNCS, vol. 2399, pp. 169–187. Springer, Heidelberg (2002)
24. Puterman, M.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons, Inc., New York (1994)
25. Vardi, M.: Automatic verification of probabilistic concurrent finite-state programs. In: FOCS 1985, pp. 327–338. IEEE (1985)
26. Vardi, M.Y.: Probabilistic linear-time model checking: An overview of the automata-theoretic approach. In: Katoen, J.-P. (ed.) ARTS 1999. LNCS, vol. 1601, pp. 265–276. Springer, Heidelberg (1999)