Parametric Markov Chains: PCTL Complexity and Fraction-free Gaussian Elimination*

(extended version, 2017-08-28)

Lisa Hutschenreiter Christel Baier
Joachim Klein
Technische Universität Dresden, Dresden, Germany
{Lisa.Hutschenreiter,Christel.Baier,Joachim.Klein}@tu-dresden.de

Parametric Markov chains have been introduced as a model for families of stochastic systems that rely on the same graph structure, but differ in the concrete transition probabilities. The latter are specified by polynomial constraints for the parameters. Among the tasks typically addressed in the analysis of parametric Markov chains are (1) the computation of closed-form solutions for reachabilty probabilities and other quantitative measures and (2) finding symbolic representations of the set of parameter valuations for which a given temporal logical formula holds as well as (3) the decision variant of (2) that asks whether there exists a parameter valuation where a temporal logical formula holds. Our contribution to (1) is to show that existing implementations for computing rational functions for reachability probabilities or expected costs in parametric Markov chains can be improved by using fraction-free Gaussian elimination, a long-known technique for linear equation systems with parametric coefficients. Our contribution to (2) and (3) is a complexity-theoretic discussion of the model checking problem for parametric Markov chains and probabilistic computation tree logic (PCTL) formulas. We present an exponential-time algorithm for (2) and a PSPACE upper bound for (3). Moreover, we identify fragments of PCTL and subclasses of parametric Markov chains where (1) and (3) are solvable in polynomial time and establish NP-hardness for other PCTL fragments.

1 Introduction

Finite-state Markovian models are widely used as an operational model for the quantitative analysis of systems with probabilistic behaviour. In many cases, only estimates of the transition probabilities are available. This, for instance, applies to fault-tolerant systems where the transition probabilities are derived from error models obtained using statistical methods. Other examples are systems operating with resource-management protocols that depend on stochastic assumptions on the future workload, or cyber-physical systems where the interaction with its environment is represented stochastically. Furthermore, often the transition probabilities of Markovian models depend on configurable system parameters that can be adjusted at design-time. The task of the designer is to find a parameter setting that is optimal with respect to a given objective. This motivated the investigation of *interval Markov chains* (IMCs) [18] specifying intervals for the transition probabilities (rather than concrete values). More general is the model of *parametric Markov chains* (pMCs), which has been introduced independently by Daws [8] and Lanotte et al. [22], where the transition probabilities are given by polynomials with rational coefficients over a fixed set of real-valued parameters $x_1, \ldots, x_k$. These concepts can be further generalized to accommodate rational functions, i. e., quotients of polynomials, as transition probabilities (see, e. g., [14]).

*The authors are supported by the DFG through the Collaborative Research Center SFB 912 – HAEC, the Excellence Initiative by the German Federal and State Governments (cluster of excellence cfaed), the Research Training Group QuantLA (GRK 1763) and the DFG-projects BA-1679/11-1 and BA-1679/12-1.

It is well-known that the probabilities p_s for reachability conditions $\Diamond Goal$ in parametric Markov chains with a finite state space S can be characterized as the unique solution of a linear equation system $A \cdot p = b$ where $p = (p_s)_{s \in S}$ is the solution vector, and $A = A(x_1, \ldots, x_k)$ is a matrix where the coefficients are rational functions. Likewise, $b = b(x_1, \ldots, x_k)$ is a vector whose coefficients are rational functions. Note that it is no limitation to assume that the entries in A and b are polynomials, as rational function entries can be converted to a common denominator, which can then be removed. Now, $A \cdot p = b$ can be viewed as a linear equation system over the field $\mathbb{Q}(x_1, \ldots, x_k)$ of rational functions with rational coefficients. As a consequence, the probabilities for reachability conditions are rational functions. This has been observed independently by Daws [8] and Lanotte et al. [22] for pMCs. Daws [8] describes a computation scheme that relies on a state-elimination algorithm inspired by the state-elimination algorithm for computing regular expressions for nondeterministic finite automata. This, however, is fairly the same as Gaussian elimination for matrices over the field of rational functions.

As observed by Hahn et al. [14], the naïve implementation of Gaussian elimination for pMCs, that treats the polynomials in A and b as syntactic atoms, leads to a representation of the rational functions $p_s = p_s(x_1, \ldots, x_k)$ as the quotient of extremely (exponentially) large polynomials. In their implementation PARAM [13] (as well as in the re-implementation within the tool PRISM [21]), the authors of [14] use computer-algebra tools to simplify rational functions in each step of Gaussian elimination by identifying the greatest common divisor (gcd) of the numerator and the denominator polynomial. Together with polynomial-time algorithms for the gcd-computation of univariate polynomials, this approach yields a polynomial-time algorithm for computing the rational functions for reachability probabilities in pMCs with a single parameter. Unfortunately, gcd-computations are known to be expensive for the multivariate case (i. e., $k \geqslant 2$) [12]. To mitigate the cost of the gcd-computations, the tool Storm [10] successfully uses techniques proposed in [17] such as caching and the representation of the polynomials in partially factorized form during the elimination steps. However, it is possible to completely avoid gcd-computations by using *one-step fraction-free Gaussian elimination*. Surprisingly, this has not yet been investigated in the context of pMCs, although it is a well-known technique in mathematics. According to Bareiss [2], this variant of Gaussian elimination probably goes back to Camille Jordan (1838–1922), and has been rediscovered several times since. Like standard Gaussian elimination it relies on the triangulation of the matrix, and finally obtains the solution by back substitution. Applied to matrices over polynomial rings the approach generates matrices with polynomial coefficients (rather than rational functions) and ensures that the degree of the polynomials in all intermediate matrices grows at most linearly. This is achieved by dividing, in each elimination step, by a factor known by construction. Thus, when applied to a pMC with linear expressions for the transition probabilities, the degree of all polynomials in the solution vector is bounded by the number of states. For the univariate case ($k = 1$), this yields an alternative polynomial-time algorithm for the computation of the rational functions for reachability probabilities. Analogous statements hold for expectations of random variables that are computable via linear equation systems. This applies to expected accumulated weights until reaching a goal, and to the expected mean payoff.

Contribution. The purpose of the paper is to study the complexity of the model checking problem for pMCs and probabilistic computation tree logic (PCTL) [15], and its extensions by expectation operators for pMCs augmented by weights for its states. In the first part of the paper (Section 3), we discuss the use of Bareiss' one-step fraction-free Gaussian elimination for the computation of reachability probabilities. The second part of the paper (Section 4) presents complexity-theoretic results for the PCTL model checking problem in pMCs. We describe an exponential-time algorithm for computing a symbolic representation of all parameter valuations under which a given PCTL formula holds, and provide a PSPACE upper bound for the decision variants that ask whether a given PCTL formula holds for some or all admissible parameter valuations. The known NP-/coNP-hardness results for IMCs [26, 6] carry over to the parametric case. We strengthen this result by showing that the existential PCTL model checking problem remains NP-hard even for acyclic pMCs and PCTL formulas with a single probability operator. For the univariate case, we prove NP-completeness for the existential PCTL model checking problem, and identify two fragments of PCTL where the model checking is solvable in polynomial time. The first fragment are Boolean combinations of threshold constraints for reachability probabilities, expected accumulated weights until reaching a goal, and expected mean payoffs. The second fragment consists of PCTL formulas in positive normal form with lower probability thresholds interpreted over pMCs satisfying some monotonicity properties. Furthermore, we observe that the model checking problem for PCTL with expectation operators for reasoning about expected costs until reaching a goal is in P for Markov chains where the weights of the states are given as polynomials over a single parameter, when restricting to Boolean combinations of the expectation operators.

Proofs and further details on the experiments omitted in the main part due to space constraints can be found in the appendix of this extended version [16].

Related work. Fraction-free Gaussian elimination is well-known in mathematics, and has been further investigated in various directions for matrices over unique factorization domains (such as polynomial rings), see e. g. [23, 19, 27, 24]. To the best of our knowledge, fraction-free Gaussian elimination has not yet been studied in the context of parametric Markovian models.

Besides the above mentioned work [8, 13, 14, 17, 9] on the computation of the rational functions for reachability probabilities in pMCs, [22] identifies instances where the parameter synthesis problem for pMCs with 1 or 2 parameters and probabilistic reachability constraints is solvable in polynomial time. These rely on the fact that there are closed-form representations of the (complex) zero's for univariate polynomials up to degree 4 and rather strong syntactic characterizations of pMCs. In Section 3 we will provide an example to illustrate that the number of monomials in the numerators of the rational functions for reachability probabilities can grow exponentially in the number of states. We hereby reveal a flaw in [22] where the polynomial-time computability of the rational functions for reachability probabilities has been stated even for the multivariate case. [11] considers an approach for solving the parametric linear equation system obtained from sparse pMCs via Laplace expansion.

Model checking problems for IMCs and temporal logics have been studied by several authors. Most in the spirit of our work on the complexity of the PCTL model checking problem for pMCs is the paper [26] which studies the complexity of PCTL model checking in IMCs. Further complexity-theoretic results of the model checking problem for IMCs and temporal logics have been established in [6] for omega-PCTL (extending PCTL by Boolean combinations of Büchi and co-Büchi conditions), and in [5] for linear temporal logic (LTL). Our results of the second part can be seen as an extension of the work [26, 6] for the case of pMCs. The NP lower bound for the multivariate case and a single threshold constraint for reachability probabilities strengthen the NP-hardness results of [26].

There exist several approaches to obtain regions of parameter valuations of a pMC in which PCTL formulas are satisfied or not, resulting in an approximative covering of the parameter space. PARAM [14, 13] employs a heuristic, sampling based approach, while PROPhESY [9] relies on SMT solving via the existential theory of the reals to determine whether a given formula holds for all valuations in a sub region. For the same problem, [25] uses a parameter lifting technique that avoids having to solve the parametric equation system by obtaining lower and upper bounds for the values in a given region by a reduction to non-parametric Markov decision processes.

2 Preliminaries

The definitions in this section require a general understanding of Markov models, standard model checking, and temporal logics. More details can be found, e. g., in [20, 1].

Discrete-time Markov chain. A *(discrete-time) Markov chain* (MC) $\mathscr{M}$ is a tuple $(S, s_{\mathit{init}}, E, P)$ where S is a non-empty, finite set of *states* with the *initial state* $s_{\mathit{init}} \in S$, $E \subseteq S \times S$ is a transition relation, and $P\colon S \times S \to [0,1]$ is the *transition probability function* satisfying $P(s,t) = 0$ if and only if $(s,t) \notin E$, and $\sum_{t \in S} P(s,t) = 1$ for all $s \in S$ with $\mathit{Post}(s) \overset{\text{def}}{=} \{t \in S : (s,t) \in E\}$ nonempty. We refer to $G_{\mathscr{M}} = (S,E)$ as the *graph* of $\mathscr{M}$. A state $s \in S$ in which $\mathit{Post}(s) = \varnothing$ is called a *trap (state)* of $\mathscr{M}$.

An *infinite path* in $\mathscr{M}$ is an infinite sequence $s_0 s_1 \ldots \in S^\omega$ of states such that $(s_i, s_{i+1}) \in E$ for $i \in \mathbb{N}$. Analogously, a *finite path* in $\mathscr{M}$ is a finite sequence $s_0 s_1 \ldots s_m \in S^*$ of states in $\mathscr{M}$ such that $(s_i, s_{i+1}) \in E$ for $i = 0, 1, \ldots, m-1$. A path is called *maximal* if it is infinite or ends in a trap. $\mathrm{Paths}(s)$ denotes the set of all maximal paths in $\mathscr{M}$ starting in s. Relying on standard techniques, every MC induces a unique probability measure $\Pr_s^{\mathscr{M}}$ on the set of all paths.

Parameters, polynomials, and rational functions. Let $x_1, \ldots, x_k$ be parameters that can assume any real value, $\overline{x} = (x_1, \ldots, x_k)$. We write $\mathbb{Q}[\overline{x}]$ for the *polynomial ring* over the rationals with variables $x_1, \ldots, x_k$. Each $f \in \mathbb{Q}[\overline{x}]$ can be written as a sum of monomials, i. e., $f = \sum_{(i_1,\ldots,i_k) \in I} \alpha_{i_1,\ldots,i_k} \cdot x_1^{i_1} \cdot x_2^{i_2} \cdot \ldots \cdot x_k^{i_k}$ where I is a finite subset of $\mathbb{N}^k$ and $\alpha_{i_1,\ldots,i_k} \in \mathbb{Q}$. If I is empty, or $\alpha_{i_1,\ldots,i_k} = 0$ for all tuples $(i_1, \ldots, i_k) \in I$, then f is the *null function*, generally denoted by 0. The *degree* of f is $\deg(f) = \max\big\{\, i_1 + \ldots + i_k : (i_1, \ldots, i_k) \in I, \alpha_{i_1,\ldots,i_k} \neq 0 \,\big\}$ where $\max(\varnothing) = 0$. A *linear function* is a function $f \in \mathbb{Q}[\overline{x}]$ with $\deg(f) \leqslant 1$. A *rational function* is a function of the form f/g with $f, g \in \mathbb{Q}[\overline{x}]$, $g \neq 0$. The field of all rational functions is denoted by $\mathbb{Q}(\overline{x})$. We write $\mathit{Constr}[\overline{x}]$ for the set of all *polynomial constraints* of the form $f \bowtie g$ where $f, g \in \mathbb{Q}[\overline{x}]$, and $\bowtie \in \{<, \leqslant, >, \geqslant, =\}$.

Parametric Markov chain. A *(plain) parametric Markov chain* on $\overline{x}$, pMC for short, is a tuple $\mathfrak{M} = (S, s_{\mathit{init}}, E, \mathbf{P})$ where S, s_{init}, and E are defined as for MCs, and $\mathbf{P}\colon S \times S \to \mathbb{Q}(\overline{x})$ is the transition probability function with $\mathbf{P}(s,t) = 0$, i. e., the null function, iff $(s,t) \notin E$. Intuitively, a pMC defines the family of Markov chains arising by plugging in concrete values for the parameters. A parameter valuation $\overline{\xi} = (\xi_1, \ldots, \xi_k) \in \mathbb{R}^k$ is said to be *admissible* for $\mathfrak{M}$ if for each state $s \in S$ we have $\sum_{t \in S} P_{\overline{\xi}}(s,t) = 1$ if $\mathit{Post}(s)$ nonempty, and $P_{\overline{\xi}}(s,t) > 0$ iff $(s,t) \in E$, where $P_{\overline{\xi}}(s,t) = \mathbf{P}(s,t)(\overline{\xi})$ for all $(s,t) \in S \times S$. Let $X_{\mathfrak{M}}$, or briefly X, denote the set of admissible parameter valuations for $\mathfrak{M}$. Given $\overline{\xi} \in X$ the Markov chain associated with $\overline{\xi}$ is $\mathscr{M}_{\overline{\xi}} = \mathfrak{M}(\overline{\xi}) = (S, s_{\mathit{init}}, E, P_{\overline{\xi}})$. The semantics of the pMC $\mathfrak{M}$ is then defined as the family of Markov chains induced by admissible parameter valuations, i. e., $[\![\mathfrak{M}]\!] = \big\{\, \mathfrak{M}(\overline{\xi}) : \overline{\xi} \in X \,\big\}$.

An *augmented pMC* is a tuple $\mathfrak{M} = (S, s_{\mathit{init}}, E, \mathbf{P}, \mathfrak{C})$ where S, s_{init}, E, and $\mathbf{P}$ are defined as for plain pMCs, and $\mathfrak{C} \subset \mathit{Constr}[\overline{x}]$ is a finite set of polynomial constraints. A parameter valuation $\overline{\xi}$ is *admissible* for an augmented pMC if it is admissible for the induced plain pMC $(S, s_{\mathit{init}}, E, \mathbf{P})$, and satisfies all polynomial constraints in $\mathfrak{C}$. As for plain pMC, we denote the set of admissible parameter valuations of an augmented pMC by $X_{\mathfrak{M}}$, or briefly X.

A, possibly augmented, pMC $\mathfrak{M}$ is called *linear*, or *polynomial*, if all transition probability functions and constraints are linear functions in $\overline{x}$, or polynomials in $\overline{x}$, respectively.

Interval Markov chain. An *interval Markov chain* (IMC) [26] can be seen as a special case of a linear augmented pMC with one parameter $x_{s,t}$ for each edge $(s,t) \in E$, and linear constraints $\alpha_{s,t} \trianglelefteq_1 x_{s,t} \trianglelefteq_2 \beta_{s,t}$ for each edge with $\alpha_{s,t}, \beta_{s,t} \in \mathbb{Q} \cap [0,1]$ and $\trianglelefteq_1, \trianglelefteq_2 \in \{<, \leqslant\}$. According to the terminology introduced in [26], this corresponds to the semantics of IMC as an "uncertain Markov chain". The alternative semantics of IMC as a Markov decision process will not be considered in this paper.

Labellings and weights. Each of these types of Markov chain, whether MC, plain or augmented pMC, or IMC, can be equipped with a *labelling function* $\mathscr{L}\colon S \to 2^{\mathrm{AP}}$, where AP is a finite set of *atomic propositions*. If not explicitly stated, we assume the implicit labelling of the Markov chain defined by using the state names as atomic propositions and assigning each name to the respective state. Furthermore, we can extend any Markov chain with a *weight function* $wgt\colon S \to \mathbb{Q}$. The value assigned to a specific state $s \in S$ is called the weight of s. It is sometimes also referred to as the *reward* of s. In addition to assigning rational values we also consider parametric weight functions $wgt\colon S \to \mathbb{Q}(\bar{x})$.

Probabilistic computation tree logic. We augment the standard notion of probabilistic computation tree logic with operators for the expected accumulated weight and mean payoff, and for comparison. Let AP be a finite set of atomic propositions. $\bowtie$ stands for $\leqslant, \geqslant, <, >$, or $=$, $c \in [0,1]$, $r \in \mathbb{Q}$. Then

$$\Phi \quad ::= \quad \mathtt{true} \mid a \mid \Phi \wedge \Phi \mid \neg\Phi \mid \mathbb{P}_{\bowtie c}(\varphi) \mid \mathbb{E}_{\bowtie r}(\rho) \mid \mathbb{C}_{\mathrm{Pr}}(\varphi, \bowtie, \varphi) \mid \mathbb{C}_{\mathrm{E}}(\rho, \bowtie, \rho) \quad \textit{state formula}$$

$$\varphi \quad ::= \quad \bigcirc\Phi \mid \Phi \,\mathsf{U}\, \Phi \quad \textit{path formula} \qquad \rho \ ::= \ \diamondsuit\!\!\!\!\!\!+ \Phi \mid \mathrm{mp}(\Phi) \quad \textit{terms for random variables}$$

where $a \in \mathrm{AP}$. The basic temporal modalities are $\bigcirc$ (*next*) and U (*until*). The usual derived temporal modalities $\Diamond$ (*eventually*), R (*release*) and $\Box$ (*always*) are defined by $\Diamond\Phi \overset{\text{def}}{=} \mathtt{true}\,\mathsf{U}\,\Phi$, and $\mathbb{P}_{\bowtie c}(\Phi_1 \,\mathsf{R}\, \Phi_2) \overset{\text{def}}{=} \mathbb{P}_{\overline{\bowtie}1-c}((\neg\Phi_1)\,\mathsf{U}\,(\neg\Phi_2))$, where, e. g., $\overline{\leqslant}$ is $\geqslant$ and $\overline{<}$ is $>$, and $\Box\Phi \overset{\text{def}}{=} \mathtt{false}\,\mathsf{R}\,\Phi$.

For an MC $\mathscr{M}$ with states labelled by $\mathscr{L}\colon S \to \mathrm{AP}$ we use the standard semantics. We only state the semantics of the probability, expectation, and comparison operators here. For each state $s \in S$, $s \models_{\mathscr{M}} \mathbb{P}_{\bowtie c}(\varphi)$ iff $\mathrm{Pr}_s^{\mathscr{M}}(\varphi) \bowtie c$, and $s \models_{\mathscr{M}} \mathbb{C}_{\mathrm{Pr}}(\varphi_1, \bowtie, \varphi_2)$ iff $\mathrm{Pr}_s^{\mathscr{M}}(\varphi_1) \bowtie \mathrm{Pr}_s^{\mathscr{M}}(\varphi_2)$. Here $\mathrm{Pr}_s^{\mathscr{M}}(\varphi)$ is short for $\mathrm{Pr}_s^{\mathscr{M}}\{\pi \in \mathrm{Paths}(s) : \pi \models_{\mathscr{M}} \varphi\}$. Furthermore, $s \models_{\mathscr{M}} \mathbb{E}_{\bowtie r}(\rho)$ iff $\mathrm{E}_s^{\mathscr{M}}(\rho^{\mathscr{M}}) \bowtie r$, and $s \models_{\mathscr{M}} \mathbb{C}_{\mathrm{E}}(\rho_1, \bowtie, \rho_2)$ iff $\mathrm{E}_s^{\mathscr{M}}(\rho_1^{\mathscr{M}}) \bowtie \mathrm{E}_s^{\mathscr{M}}(\rho_2^{\mathscr{M}})$, where $\mathrm{E}_s^{\mathscr{M}}(\cdot)$ denotes the expected value of the respective random variable. For detailed semantics of the expectation operators, see Appendix A.4. We write $\mathscr{M} \models \Phi$ iff $s_{init} \models_{\mathscr{M}} \Phi$.

Notation: PCTL+EC and sublogics. We use PCTL to refer to unaugmented probabilistic computation tree logic. If we add only the expectation operator we write PCTL+E, and, analogously, PCTL+C if we only add the comparison operator for probabilities. PCTL+EC denotes the full logic defined above.

DAG-representation and length of formulas. We consider for any PCTL+EC state formula the *directed acyclic graph* (DAG) representing its syntactic structure. Each node of the DAG represents one of the sub-state formulas. The use of a DAG rather than the syntax tree allows the representation of subformulas that occur several times in the formula Φ by a single node. The leaves of the DAG can be the Boolean constant `true`, and atomic propositions. The inner nodes of the DAG, e. g., of a PCTL formula, are labelled with one of the operators $\wedge$, $\neg$, $\mathbb{P}_{\bowtie c}(\cdot\,\mathsf{U}\,\cdot)$, $\mathbb{P}_{\bowtie c}(\bigcirc\,\cdot)$. Nodes labelled with $\neg$ and $\mathbb{P}_{\bowtie c}(\bigcirc\,\cdot)$ have a single outgoing edge, while nodes labelled with $\wedge$ or $\mathbb{P}_{\bowtie c}(\cdot\,\mathsf{U}\,\cdot)$ have two outgoing edges. For the above-mentioned extensions of PCTL the set of possible inner node labels is extended accordingly. For example, a node v representing the PCTL+C formula $\mathbb{C}_{\mathrm{Pr}}(\bigcirc\Phi_1, \bowtie, \Phi_2\,\mathsf{U}\,\Phi_3)$ has three outgoing edges. If $\Phi_1 = \Phi_2$ then there are two parallel edges from v to a node representing Φ_1. The length of a PCTL+EC formula is defined as the number of nodes in its DAG.

3 Fraction-free Gaussian elimination

Given a pMC $\mathfrak{M}$ as in Section 2, the probabilities $\mathrm{Pr}_s^{\mathfrak{M}(\bar{x})}(\Diamond a)$ for reachability conditions are rational functions and computable via Gaussian elimination. As stated in the introduction, this has been originally observed in [8, 22] and realized, e. g., in the tools PARAM [13] and Storm [9, 10] together with techniques

Algorithm 1 One-step fraction-free Gaussian elimination [2]

```
procedure FractionFreeGauss($A = (a_{ij})_{i,j=1,\dots,n}$, $b = (b_i)_{i=1,\dots,n}$)
    $a_{0,0} = 1$
    for $m = 1,\dots,n-1$ do                                        ▷ triangulation, assuming $a_{m,m} \neq 0$
        for $i = m+1,\dots,n$ do
            for $j = m+1,\dots,n$ do
                $a_{i,j} = \big(a_{m,m} \cdot a_{i,j} - a_{i,m} \cdot a_{m,j}\big)/a_{m-1,m-1}$     ▷ exploit exact divisibility by $a_{m-1,m-1}$
            $b_i = \big(a_{m,m} \cdot b_i - a_{i,m} \cdot b_m\big)/a_{m-1,m-1}$              ▷ exploit exact divisibility by $a_{m-1,m-1}$
            $a_{i,m} = 0$
    for $m = n-1,\dots,1$ do                                        ▷ back substitution
        $b_m = \big(a_{n,n} \cdot b_m - \sum_{i=m+1}^{n} a_{m,i} \cdot b_i\big)/a_{m,m}$          ▷ exploit exact divisibility by $a_{m,m}$
    return $\big(b_i/a_{n,n}\big)_{i=1,\dots,n}$                                 ▷ rational solution functions
```

based on gcd-computations on multivariate polynomials. In this section, we discuss the potential of fraction-free Gaussian elimination as an alternative, which is well-known in mathematics [2, 12], but to the best of our knowledge, has not yet been considered in the context of pMCs.

While the given definitions allow for rational functions in the transition probability functions of (augmented) pMCs, we will focus on polynomial (augmented) pMCs throughout the remainder of the paper. Generally, a linear equation systems containing rational functions as coefficients can be rearranged to one containing only polynomials by multiplying each line with the common denominator of the respective rational functions. Due to the multiplications this involves the risk of a blow-up in the coefficient size. To avoid this we add variables in the following way. Let $\mathfrak{M} = (S, s_{init}, E, \mathbf{P}, \mathfrak{C})$ be an (augmented) pMC. For all $(s,t) \in E$ introduce a fresh variable $x_{s,t}$. By definition $\mathbf{P}(s,t) = \frac{f_{s,t}}{g_{s,t}}$ for some $f_{s,t}, g_{s,t} \in \mathbb{Q}[\bar{x}]$. Let $\mathbf{P}'(s,t) = f_{s,t} \cdot x_{s,t}$ if $(s,t) \in E$, $\mathbf{P}'(s,t) = 0$ if $(s,t) \notin E$, $\mathfrak{C}' = \mathfrak{C} \cup \{g_{s,t} \cdot x_{s,t} = 1 : (s,t) \in E\}$. Then $\mathfrak{M}' = (S, s_{init}, E, \mathbf{P}', \mathfrak{C}')$ is a polynomial augmented pMC.

Linear equation systems with polynomial coefficients. Let $x_1,\dots,x_k$ be parameters, $\bar{x} = (x_1,\dots,x_k)$. We consider linear equation systems of the form $A \cdot p = b$, where $A = (a_{i,j})_{i,j=1,\dots,n}$ is a non-singular $n \times n$-matrix with $a_{i,j} = a_{i,j}(\bar{x}) \in \mathbb{Q}[\bar{x}]$. Likewise, $b = (b_i)_{i=1,\dots,n}$ is a vector of length n with $b_i = b_i(\bar{x}) \in \mathbb{Q}[\bar{x}]$. The solution vector $p = (p_i)_{i=1,\dots,n}$ is a vector of rational functions $p_i = f_i/g_i$ with $f_i, g_i \in \mathbb{Q}[\bar{x}]$. By Cramer's rule we obtain $p_i = \frac{\det(A_i)}{\det(A)}$, where $\det(A)$ is the determinant of A, and $\det(A_i)$ the determinant of the matrix obtained when substituting the i-th column of A by b. If the coefficients of A and b have at most degree d, the Leibniz formula implies that f_i and g_i have at most degree $n \cdot d$.

▶ **Lemma 1.** *There is a family $(\mathfrak{M}_k)_{k \geqslant 2}$ of acyclic linear pMCs where $\mathfrak{M}_k$ has k parameters and $n = k+3$ states, including distinguished states s_0 and goal, such that $\Pr_{s_0}^{\mathfrak{M}(\bar{x})}(\Diamond goal)$ is a polynomial for which even the shortest sum-of-monomial representation has 2^k monomials.*

One-step fraction-free Gaussian elimination is a variant of fraction-free Gaussian elimination that allows for divisions which are known to be exact at the respective point in the algorithm. When using *naïve fraction-free Gaussian elimination* the new coefficients after the m-th step, $m = 1,\dots,n-1$, are computed as $a_{i,j}^{(m)} = a_{i,j}^{(m-1)} a_{m,m}^{(m-1)} - a_{i,m}^{(m-1)} a_{m,j}^{(m-1)}$ for $i, j = m+1,\dots,n$, where $a_{i,j}^{(0)} = a_{i,j}$. When applied to systems with polynomial coefficients this results in the degree doubling after each step, so the degree grows exponentially. In a step of *one-step fraction-free Gaussian elimination* (see Algorithm 1), the computation of the coefficients changes to $a_{i,j}^{(m)} = \big(a_{i,j}^{(m-1)} a_{m,m}^{(m-1)} - a_{i,m}^{(m-1)} a_{m,j}^{(m-1)}\big)/a_{m-1,m-1}^{(m-1)}$ with $a_{0,0}^{(0)} = 1$.

Using Sylvester's identity one can prove that $a_{i,j}^{(m)}$ is again a polynomial, and that $a_{m-1,m-1}^{(m-1)}$ is in general the maximal possible divisor. The b_i are updated analogously. If the maximal degree of the initial coefficients of A and b is d, this technique therefore guarantees, that after m steps the degree of the coefficients is at most $(m+1)\cdot d$, i. e., it grows linear in d during the procedure. For polynomials the division by $a_{m-1,m-1}^{(m-1)}$ can be done using standard polynomial division. The time-complexity of the exact multivariate polynomial division in this case is in each step $\mathcal{O}\big(\text{poly}(md)^k\big)$, so for the full one-step fraction-free Gaussian elimination it is $\mathcal{O}\big(\text{poly}(n,d)^k\big)$.

Proposition 4.3 in [22] states that the rational functions $p_i = f_i/g_i$ for reachability probabilities in pMC with a representation of the polynomials f_i, g_i as sums of monomials (called normal form in [22]) are computable in polynomial time. This contradicts Lemma 1 which shows that the number of monomials in the representation of a reachability probability as a sum of monomials can be exponential in the number of parameters. However, the statement is correct for the univariate case.

▶ **Lemma 2.** *Let $\mathfrak{M}$ be a polynomial pMC over a single parameter and T a set of states. Then, the rational functions for the reachability probabilities $\Pr_s^{\mathfrak{M}}(\Diamond T)$ are computable in polynomial time. Analogously, rational functions for the expected accumulated weight until reaching T or the expected mean payoff are computable in polynomial time.*

Note that the degrees of the polynomials $a_{i,j}^{(m)}$ and $b_j^{(m)}$ computed by one-step fraction-free Gaussian elimination for reachability probabilities are bounded by $(m+1)\cdot d$, where $d = \max_{s,t\in S}\deg(\mathbf{P}(s,t))$, so the polynomials have representations as sums of at most $md+1$ monomials. In particular, the degree and representation size of the final polynomials $f_s = b_s^{(n)}$ and $g_s = a_{s,s}^{(n)}$ for the rational functions $\Pr_s^{\mathfrak{M}(x)}(\Diamond goal) = f_s/g_s$ is in $\mathcal{O}(nd)$ where n is the number of states of $\mathfrak{M}$. Another observation concerns the case where only the right-hand side of the linear equation system is parametric. Systems of this form occur, e. g., when considering expectation properties for MCs with parametric weights.

▶ **Lemma 3.** *Let $A\cdot p = b$ be a parametric linear equation system as defined above where A is parameter-free. Then the solution vector $p = (p_i)_{i=1,\dots,n}$ consist of polynomials of the form $p_i = \sum_{i=1}^{n}\beta_i\cdot b_i$ with $\beta_i\in\mathbb{Q}$ and can be computed in polynomial time.*

Stratification via SCC-decomposition. It is well known (e. g., [7, 17]) that for probabilistic/parametric model checking a decomposition into strongly-connected components (SCCs) can yield significant performance benefits due to the structure of the underlying models. We have adapted the one-step fraction-free Gaussian elimination approach by a preprocessing step that permutes the matrix according to the topological ordering of the SCCs. This results in the coefficient matrix already having a stair-like form at the start of the algorithm. In the triangulation part of the algorithm, each SCC can now be considered separately, as non-zero entries below the main diagonal only occur within each SCC. While the back-substitution in the general one-step fraction-free elimination will result in each entry on the main diagonal being equal to the last, this property is now only maintained within the SCCs. Formally, this means that the back substitution step in Algorithm 1 is replaced by the following:

$$b_m \quad = \quad \Big(a^*(\text{current SCC})\cdot b_m - \sum_{i=m+1}^{n} a_{m,i}\cdot b_i\cdot\frac{a^*(\text{current SCC})}{a^*(\text{SCC at } i)}\Big) \,/\, a_{m,m}$$

where $a^*(\text{SCC at } n) = a_{n,n}$, and, for $i = 1,\dots,n-1$, $a^*(\text{SCC at } i) = a^*(\text{SCC at } i+1)$ if the i-th and $(i+1)$-st state belong to the same SCC and $a^*(\text{SCC at } i) = a_{i,i}\cdot a^*(\text{SCC at } i+1)$ otherwise. Intuitively, $a^*(\text{SCC at } i)$ is the product of the a's on the diagonal corresponding to the last states in the current SCC and the SCCs below. Of course, the return statement also has to be adjusted accordingly. The advantage of

Table 1: Statistics for "complete pMC". Matrix rows and number of distinct parameters, as well as time for solving the parametric equation system per solver. For $n = 7$, all solvers timed out (30min).

n	rows	param.	*eigen*	*state-elim*	*GE-ff*	*red(GE-ff)*
4	4	20	0.47 s	0.64 s	0.06 s	0.01 s
5	5	30	44.47 s	42.09 s	2.13 s	1.52 s
6	6	42	time-out	time-out	221.27 s	21.53 s

this approach is that the polynomials in the rational functions aside from the ones in the first strongly connected component will have an even lower degree.

Implementation and experiments. For a first experimental evaluation of the one-step fraction-free Gaussian elimination approach (*GE-ff*) in the context of probabilistic model checking, we have implemented this method (including the SCC decomposition and topological ordering described above) as an alternative solver for parametric linear equation systems in the state-of-the-art probabilistic model checker Storm [10]. We compare *GE-ff* against the two solvers provided by Storm (v1.0.1) for solving parametric equation systems, i. e., the solver based on the *eigen* linear algebra library[1] and on state elimination (*state-elim*) [14]. Both of Storm's solvers use partially factorized representations of the rational functions provided by the CArL library[2]. This approach, together with caching, was shown [17] to be beneficial due to improved performance of the gcd-computations during the simplification steps.

It should be noted that our implementation is intended to provide first results that allow to gauge whether the fraction-free method, by avoiding gcd-computations, can be beneficial in practice and is thus rather naïve in certain aspects. As an example, it currently relies on a dense matrix representation, with performance improvements for larger models to be expected from switching to sparse representations as used in Storm's *eigen* and *state-elim* solvers. In addition to the fraction-free approach, our solver can also be instantiated to perform a straight-forward Gaussian elimination, using any of the representations for rational functions provided by the CArL library. In all our experiments, we have compared the solutions obtained by the different solvers and verified that they are the same.

Experimental studies. The source code of our extension of Storm and the artifacts of the experiments are available online.[3] As our *GE-ff* implementation is embedded as an alternative solver in Storm, we mainly report the time actually spent for solving the parametric equation system, as the other parts of model checking (model building, precomputations) are independent of the chosen solver. For benchmarking, we used a machine with two Intel Xeon E5-2680 8-core CPUs at 2.70GHz and with 384GB RAM, a time out of 30 minutes and a memory limit of 30GB. All the considered solvers run single-threaded. We have considered three different classes of case studies for experiments.

Complete pMC. As a first experiment to gauge the efficiency in the presence of a high ratio of parameters to states, we considered a family of pMCs with a complete graph structure (over n states) and one parameter per transition, resulting in $n \cdot (n+1)$ parameters (for details see Appendix B.2).

Table 1 depicts statistics for the corresponding computations, for the two standard solvers in Storm (*eigen* and *state-elim*), as well as our fraction-free implementation (*GE-ff*). For *state-elim*, we always use the default elimination order (forward). The time for *GE-ff* corresponds to the time until a solution rational function (for all states) is obtained. As the numerator and denominator of these rational functions

[1] `http://eigen.tuxfamily.org/`

[2] `https://github.com/smtrat/carl`

[3] `http://wwwtcs.inf.tu-dresden.de/ALGI/PUB/GandALF17/`

Table 2: Statistics for "Israeli-Jalfon", with strong bisimulation quotienting. Matrix rows and number of distinct parameters, as well as time for solving the parametric equation system per solver.

n	*k*	rows	param.	*eigen*	*state-elim*	*GE-fac*	*GE-ff*	*red(GE-ff)*
4	3	21	4	1.01 s	0.86 s	0.73 s	0.16 s	0.20 s
4	4	15	4	0.94 s	0.58 s	0.58 s	0.16 s	0.13 s
5	2	16	5	19.13 s	30.83 s	29.46 s	9.36 s	0.33 s
5	3	36	5	360.43 s	747.32 s	172.16 s	485.78 s	95.92 s
5	4	51	5	457.55 s	1542.97 s	442.80 s	614.01 s	742.69 s
5	5	31	5	368.70 s	1597.29 s	252.92 s	622.00 s	414.72 s

are not necessarily coprime, for comparison purposes we list as well the time needed for simplification (*red(GE-ff)*) via division by the gcd. As can be seen, here, the fraction-free approach significantly outperforms Storm's standard solvers and scales to a higher number of parameters. We confirmed using profiling that the standard solvers indeed spend most of the time in gcd-computations.

Multi-parameter Israeli-Jalfon self-stabilizing. The benchmarks used to evaluate parametric model checking implementations in previous papers tend to be scalable in the number of components but use a fixed number of parameters, usually 2. To allow further experiments with an increasing number of parameters, we considered a pMC-variant of the Israeli-Jalfon self-stabilizing protocol with n processes, k initial tokens and n parameters (for details see Appendix B.2).

Table 2 depicts the time spent for computing the rational functions for several instances. As can be seen, the fraction-free approach is competitive for the smaller instances, with performance between the *eigen* and *state-elim* solvers for the larger instances. We have also included running times for *GE-fac*, i. e., for our naïve implementation of Gaussian elimination using the representation for rational functions as used by Storm for the standard solvers, including automatic gcd-based simplification after each step to ensure that numerator and denominator are coprime. *GE-fac* operates on the same, topologically sorted matrix as the fraction-free *GE-ff*. Curiously, *GE-fac* is able to outperform the *eigen* solver for some of the larger instances. We believe this is mainly due to differences in the matrix permutation and their effect on the elimination order, which is known to have a large impact on performance (e. g., [9]).

Benchmark case studies from [9]. Furthermore, we considered several case study instances that were used in [9] to benchmark parametric model checkers, namely the *brp*, *crowds*, *egl*, *nand*, *zeroconf* models. Table 3 depicts statistics for selected instances, for further details see Appendix B.2. The application of bisimulation quotienting often has a large impact on the size of the linear equation system, so we performed experiments with and without quotienting. For *crowds*, bisimulation quotienting was particularly effective, with all considered instances having a very small state space and negligible solving times. For the non-quotiented instances, Storm's standard solvers outperform *GE-ff*. For the *zeroconf* instance in Table 3, *GE-ff* is competitive. Note that the models in the *brp*, *egl* and *nand* case studies are acyclic and that the parametric transition probabilities and rewards are polynomial. As a consequence, the gcd-computations used in Storm's solvers don't impose a significant overhead as the rational functions during the computation all have denominator polynomials of degree zero.

Overall, the experiments have shown that there are instances where the fraction-free approach can indeed have a positive impact on performance. Keeping in mind that our implementation has not yet been significantly optimized, we believe that the fraction-free approach is an interesting addition to the gcd-based solver approaches. In particular, the application of better heuristics for the order of processing (i. e., the permutation of the matrix) could still lead to significant performance increases.

Table 3: Selected statistics for the benchmarks of [9]. Matrix rows and number of distinct parameters, as well as time for solving the parametric equation system per solver.

model	rows	param.	*eigen*	*state-elim*	*GE-ff*	*red(GE-ff)*
Crowds (3,5), weak-bisim	40	2	0.08 s	0.06 s	0.02 s	0.13 s
Crowds (5,5), weak-bisim	40	2	0.08 s	0.06 s	0.02 s	0.11 s
Crowds (10,5), weak-bisim	40	2	0.08 s	0.06 s	0.02 s	0.11 s
Crowds (3,5)	715	2	0.99 s	0.80 s	11.44 s	63.39 s
Crowds (5,5)	2928	2	6.36 s	5.51 s	1271.95 s	time-out
Crowds (10,5)	25103	2	139.82 s	173.15 s	time-out	—
Zeroconf (1000)	1002	2	81.03 s	45.01 s	49.43 s	11.35 s

4 Complexity of the PCTL+EC model checking problem

We now study the complexity of the following variants of the PCTL+EC model checking problem. Given an augmented pMC $\mathfrak{M} = (S, s_{init}, E, \mathbf{P}, \mathfrak{C})$ and a PCTL+EC (state) formula Φ:

(All) Compute a representation of the set of all satisfying parameter valuations,
i. e., the set of all admissible parameter valuations $\overline{\xi} \in X$ such that $\mathfrak{M}(\overline{\xi}) \models \Phi$.

(MC-E) Does there exist a valuation $\overline{\xi} \in X$ such that $\mathfrak{M}(\overline{\xi}) \models \Phi$?

(MC-U) Does $\mathfrak{M}(\overline{\xi}) \models \Phi$ hold for all admissible valuations $\overline{\xi} \in X$?

(MC-E) and (MC-U) are essentially duals of each other. Note that the answer for the universal variant (MC-U) is obtained by the negation of the answer for (MC-E) with formula $\neg\Phi$, and vice versa. In what follows, we shall concentrate on (All) and the existential model checking problem (MC-E).

Computing all satisfying parameter valuations. As before, $X = X_{\mathfrak{M}}$ denotes the set of admissible valuations. In what follows, let χ be the conjunction of the polynomial constraints in $\mathfrak{C}$ as well as the constraints $\sum_{t \in S} \mathbf{P}(s,t) = 1$ for each non-trap state $s \in S$, and $0 < \mathbf{P}(s,t)$ for each edge $(s,t) \in E$. We then have $\overline{\xi} \models \chi$ if and only if $\overline{\xi}$ is admissible, i. e., $\overline{\xi} \in X$.

Let Φ be a PCTL+EC formula. The *satisfaction function* $\mathrm{Sat}_{\mathfrak{M}}(\Phi) \colon X \to 2^S$ is defined by:

$$\mathrm{Sat}_{\mathfrak{M}}(\Phi)(\overline{\xi}) \overset{\text{def}}{=} \left\{ s \in S : s \models_{\mathfrak{M}(\overline{\xi})} \Phi \right\} = \mathrm{Sat}_{\mathfrak{M}(\overline{\xi})}(\Phi)$$

We now present an algorithm to compute a symbolic representation of the satisfaction function that groups valuations with the same corresponding satisfaction set together. More precisely, we deal with a representation of the satisfaction function $\mathrm{Sat}_{\mathfrak{M}}(\Phi)$ by a finite set $\mathtt{Sat}_{\mathfrak{M}}(\Phi)$ of pairs (γ, T) where γ is a Boolean combination of constraints and $T \subseteq S$ such that (i) $(\gamma, T) \in \mathtt{Sat}_{\mathfrak{M}}(\Phi)$ and $\overline{\xi} \models \gamma$ implies $T = \mathrm{Sat}_{\mathfrak{M}}(\Phi)(\overline{\xi})$, and (ii) whenever $T = \mathrm{Sat}_{\mathfrak{M}}(\Phi)(\overline{\xi})$ then there is a pair $(\gamma, T) \in \mathtt{Sat}_{\mathfrak{M}}(\Phi)$ such that $\overline{\xi} \models \gamma$.

Given the DAG representation of the PCTL formula Φ, we follow the standard model checking procedure for CTL-like branching-time logics and compute $\mathtt{Sat}_{\mathfrak{M}}(\Psi)$ for the subformulas Ψ assigned to the nodes in the DAG for Φ in a bottom-up manner. As the leaves of the DAG can be atomic propositions a or the formula $\mathtt{true}$, the base cases are $\mathtt{Sat}_{\mathfrak{M}}(\mathtt{true}) = \big\{(\chi, S)\big\}$, and $\mathtt{Sat}_{\mathfrak{M}}(a) = \big\{ (\chi, \{s \in S : a \in \mathscr{L}(s)\}) \big\}$. Consider now the inner node v of the DAG for Φ labelled by the outermost operator of the subformula Ψ. Suppose that the children of v have already been treated, so when computing $\mathtt{Sat}_{\mathfrak{M}}(\Psi)$ the satisfaction sets of the proper subformulas of Ψ are known. If v is labelled by $\neg$ or $\wedge$, i. e., $\Psi = \neg\Psi'$ or $\Psi = \Psi_1 \wedge \Psi_2$, then

$\mathtt{Sat}_{\mathfrak{M}}(\Psi) = \big\{ (\gamma, S \setminus T) : (\gamma, T) \in \mathtt{Sat}_{\mathfrak{M}}(\Psi') \big\}$ respectively $\mathtt{Sat}_{\mathfrak{M}}(\Psi) = \big\{ \ (\gamma_1 \wedge \gamma_2, T_1 \cap T_2) \ : \ (\gamma_i, T_i) \in \mathtt{Sat}_{\mathfrak{M}}(\Psi_i), \ i = 1,2 \ \big\}$. If $\Psi = \mathbb{P}_{\bowtie c}(\Psi_1 \mathbin{\mathsf{U}} \Psi_2)$, then

$$\mathtt{Sat}_{\mathfrak{M}}(\Psi) \ = \ \big\{ \ (\gamma_1 \wedge \gamma_2 \wedge \delta_{\gamma_1,T_1,\gamma_2,T_2,R}) \ : \ (\gamma_1, T_1) \in \mathtt{Sat}_{\mathfrak{M}}(\Psi_1), \ (\gamma_2, T_2) \in \mathtt{Sat}_{\mathfrak{M}}(\Psi_2), \ R \subseteq S \ \big\}$$

where $\delta_{\gamma_1,T_1,\gamma_2,T_2,R}$ is the conjunction of the constraints $\Pr^{\mathfrak{M}}_s(T_1 \mathbin{\mathsf{U}} T_2) \bowtie c$ for each state $s \in R$, and $\Pr^{\mathfrak{M}}_s(T_1 \mathbin{\mathsf{U}} T_2) \not\bowtie c$ for each state $s \in S \setminus R$. Here, $\Pr^{\mathfrak{M}}_s(T_1 \mathbin{\mathsf{U}} T_2)$ is the rational function that has been computed using (i) a graph analysis to determine the set U of states s with $s \models \exists (T_1 \mathbin{\mathsf{U}} T_2)$ and (ii) fraction-free Gaussian elimination (Section 3) to compute the rational functions $\Pr^{\mathfrak{N}}_s(\Diamond T_2)$ in the pMC $\mathfrak{N}$ resulting from $\mathfrak{M}$ by turning the states in $(S \setminus U) \cup T_2$ into traps. If f_s and g_s are polynomials computed by fraction-free Gaussian elimination such that $\Pr^{\mathfrak{M}}_s(T_1 \mathbin{\mathsf{U}} T_2) = f_s/g_s$ then $\Pr^{\mathfrak{M}}_s(T_1 \mathbin{\mathsf{U}} T_2) \bowtie c$ is a shortform notation for $f_s - c \cdot g_s \bowtie 0$. The treatment of $\mathbb{P}_{\bowtie c}(\bigcirc \Psi)$ and the expectation operators is similar, and can be found in Appendix C.1. After treating a node of the DAG, we can simplify the set $\mathtt{Sat}_{\mathfrak{M}}(\Psi)$ by first removing all pairs (γ, T) where γ is not satisfiable (using algorithms for the existential theory of the reals), and afterwards combining all pairs with the same T-component, that is, instead of m pairs $(\gamma_1, T), \ldots, (\gamma_m, T) \in \mathtt{Sat}_{\mathfrak{M}}(\Psi)$, we consider a single pair $(\gamma_1 \vee \ldots \vee \gamma_m, T)$. To answer question (All), the algorithm finally returns the disjunction of all formulas γ with $s_{init} \in T$ for $(\gamma, T) \in \mathtt{Sat}_{\mathfrak{M}}(\Phi)$.

Complexity bounds of (All) and (MC-E). The existential theory of the reals is known to be in PSPACE and NP-hard, and there is an upper bound on the time-complexity, namely $\ell^{k+1} \cdot d^{\mathcal{O}(k)}$ where ℓ is the number of constraints, d the maximum degree of the polynomials in the constraints, and k the number of parameters [3]. Recall from Section 3 that a known upper bound on the time-complexity of one-step fraction-free Gaussian elimination is $\mathcal{O}\big(\mathrm{poly}(n,d)^k\big)$, where n is the number of equations, d the maximum degree of the initial coefficient polynomials, and k the number of parameters. Combining both approaches, the one-step fraction-free Gaussian elimination for solving linear equation systems with polynomial coefficients, and the existential theory of the reals for treating satisfiability of conjunctions of polynomial constraints, one directly obtains the following bound for the computational complexity of PCTL+EC model checking on augmented polynomial pMCs. Note that this assumes that the number of constraints in $\mathfrak{C}$ is at most polynomial in the size of S.

▶ **Theorem 4** (Exponential-time upper bound for problem (All)). *Let Φ be a PCTL+EC formula. Given an augmented polynomial pMC $\mathfrak{M}$, where the maximum degree of transition probabilities $\mathbf{P}(s,t)$, and polynomials in the constraints in $\mathfrak{C}$ is d, a symbolic representation of the satisfaction function* $\mathrm{Sat}_{\mathfrak{M}}(\Phi)$ *is computable in time $\mathcal{O}\big(|\Phi| \cdot \mathrm{poly}\big(size(\mathfrak{M}), d\big)^{k \cdot |\Phi|_{\mathbb{P},\mathbb{E},\mathbb{C}}}\big)$, where $|\Phi|_{\mathbb{P},\mathbb{E},\mathbb{C}}$ is the number of probability, expectation and comparison operators in Φ.*

▶ **Theorem 5** (PSPACE upper bound for problem (MC-E)). *The existential PCTL+EC model checking problem (MC-E) for augmented pMC is in PSPACE.*

Sketch of proof. The main idea of a polynomially space-bounded algorithm is to guess nondeterministically sets T_Ψ of states for the subformulas Ψ where the outermost operator is a probability, expectation or comparison operator, and then apply a polynomially space-bounded algorithm for the existential theory of the reals [3] to check whether there is a parameter valuation $\overline{\xi}$ such that $T_\Psi = \mathrm{Sat}_{\mathfrak{M}}(\Psi)(\overline{\xi})$ for all Ψ. ◀

NP- and coNP-hardness of (MC-E) follow from results for IMCs [26, 6]. More precisely, [6] provides a polynomial reduction from SAT to the (existential and universal) PCTL model checking problem for IMCs. In fact, the reduction of [6] does not require full PCTL, instead Boolean combinations of simple probabilistic constraints $\mathbb{P}_{\geqslant c_i}(\bigcirc a_i)$ without nesting of the probability operators are sufficient. The following theorem strengthens this result by stating NP-hardness of (MC-E) even for formulas $\mathbb{P}_{>c}(\Diamond a)$ consisting of a single probability constraint for a reachability condition.

▶ **Theorem 6** (NP-hardness for single probabilistic operator, multivariate case)**.** *Given an augmented polynomial pMC $\mathfrak{M}$ on parameters $\overline{x}$ with initial state s_{init} and an atomic proposition a, and a probability threshold $c \in \mathbb{Q} \cap]0,1[$, the problem to decide whether there exists $\overline{\xi} \in X$ such that $\mathrm{Pr}_{s_{init}}^{\mathfrak{M}(\overline{\xi})}(\Diamond a) > c$ is NP-hard, even for acyclic pMCs with the assigned transition probabilities being either constant, or linear in one parameter, i. e., $\mathbf{P}(s,t) \in \bigcup_{i=1}^{k} \mathbb{Q}[x_i]$, $\deg(\mathbf{P}(s,t)) \leq 1$, for all $(s,t) \in E$, and where the polynomial constraints for the parameters $x_1,\dots,x_k$ are of the form $f(x_i) \geqslant 0$ with $f \in \mathbb{Q}[x_i]$, $\deg(f) \leq 2$.*

Univariate pMCs. In many scenarios, the number of variables has a fixed bound instead of increasing with the model size. We consider here the case of *univariate* pMC, i. e., pMC with a single parameter.

▶ **Theorem 7** (PCTL+EC model checking without nesting in P, univariate case)**.** *Let Φ be a PCTL+EC formula without nested probability, expectation or comparison operators, and let $\mathfrak{M}$ be a polynomial pMC on the single parameter x. The problem to decide whether there exists an admissible parameter valuation $\xi \in X$ such that $\mathfrak{M}(\xi) \models \Phi$ is in P.*

Sketch of proof. If we restrict PCTL+EC to Boolean combinations of probability, expectation, and comparison operators, (MC-E) can be dealt with by first computing polynomial constraints for s_{init} for each probability, expectation, and comparison operator independently (this can be done in polynomial time by Lemma 2), and afterwards applying a polynomial-time algorithm for the univariate existential theory of the reals [4] once to the appropriate Boolean combination of the constraints. ◀

▶ **Theorem 8** (NP-completeness for full PCTL+EC, univariate case)**.** *Let Φ be a PCTL+EC formula, and let $\mathfrak{M}$ be a polynomial pMC on the single parameter x. The PCTL+EC model checking problem to decide whether there exists an admissible parameter valuation $\xi \in X$ such that $\mathfrak{M}(\xi) \models \Phi$ is NP-complete. NP-hardness even holds for acyclic polynomial pMCs and the fragment of PCTL+C that uses the comparison operator $\mathbb{C}_{\mathrm{Pr}}$, but not the probability operator $\mathbb{P}$, as well as for (cyclic) polynomial pMC in combination with PCTL.*

(MC-E) for monotonic PCTL on univariate pMCs. The parameters in pMC typically have a fixed meaning, e. g., probability for the occurrence of an error, in which case the probability to reach a state where an error has occurred is increasing in x. This motivates the consideration of univariate pMCs and PCTL formulas that are monotonic in the following sense.

Given a univariate polynomial pMC $\mathfrak{M} = (S, s_{init}, E, \mathbf{P})$, let E_+ denote the set of edges $(s,t) \in E$ such that the polynomial $\mathbf{P}(s,t)$ is monotonically increasing in X, i. e., whenever $\xi_1, \xi_2 \in X$ and $\xi_1 < \xi_2$ then $\mathbf{P}(s,t)(\xi_1) \leqslant \mathbf{P}(s,t)(\xi_2)$. Let S_+ denote the set of states s such that for each finite path $\pi = s_0 s_1 \dots s_m$ with $s_m = s$ we have $(s_i, s_{i+1}) \in E_+$ for $i = 0,1,\dots,m-1$.

As $(s,t) \in E_+$ iff there is no value $\xi \in \mathbb{R}$ such that $\xi \models \chi \wedge (\mathbf{P}(s,t)' < 0)$, the set E_+ is computable in polynomial time using a polynomial-time algorithm for the univariate theory of the reals [4]. Here, χ is as before the Boolean combination of polynomial constraints characterizing the set X of admissible parameter values, and $\mathbf{P}(s,t)'$ is the first derivative of the polynomial $\mathbf{P}(s,t)$. Thus, the set S_+ is computable in polynomial time.

▶ **Lemma 9.** *Let $\mathfrak{M}$ be a univariate polynomial pMC and Ψ a monotonic PCTL formula, that is, Ψ is in the PCTL fragment obtained by the following grammar:*

$$\begin{array}{rcl} \Phi & ::= & a \in S_+ \mid \Phi \wedge \Phi \mid \Phi \vee \Phi \mid \mathbb{P}_{\geqslant c}(\varphi) \mid \mathbb{P}_{>c}(\varphi) \\ \varphi & ::= & \bigcirc \Phi \mid \Phi \,\mathsf{U}\, \Phi \mid \Phi \,\mathsf{R}\, \Phi \mid \Diamond \Phi \mid \Box \Phi \end{array}$$

where $c \in \mathbb{Q}_{>0}$. Then, $\mathrm{Sat}_{\mathfrak{M}(\xi_1)}(\Psi) \subseteq \mathrm{Sat}_{\mathfrak{M}(\xi_2)}(\Psi)$ for any two valuations ξ_1 and ξ_2 of x with $\xi_1 < \xi_2$.

Hence, if Ψ is monotonic then the satisfaction function $X \to 2^S$, $\xi \mapsto \mathrm{Sat}_{\mathfrak{M}}(\Psi)(\xi) = \mathrm{Sat}_{\mathfrak{M}(\xi)}(\Psi)$ is monotonic. For each monotonic PCTL formula Ψ there exist $S_\Psi \subseteq S$ and $\xi_\Psi \in X$ such that $\mathrm{Sat}_{\mathfrak{M}(\xi)}(\Psi) = S_\Psi$ for all $\xi \geqslant \xi_\Psi$ and $\mathrm{Sat}_{\mathfrak{M}(\xi')}(\Psi) \subseteq S_\Psi$ for all $\xi' < \xi_\Psi$. To decide (MC-E) for a given monotonic formula Φ, it suffices to determine the sets S_Ψ for the sub-state formulas Ψ of Φ. This can be done in polynomial time. Using this observation, we obtain:

▶ **Theorem 10** ((MC-E) for monotonic PCTL on univariate pMC). *Let $\mathfrak{M} = (S, s_{init}, E, \mathbf{P}, \mathfrak{C})$ be a univariate polynomial pMC on x, and Φ a monotonic PCTL formula. Then the model checking problem to decide whether there exists an admissible parameter valuation ξ for x such that $\mathfrak{M}(\xi) \models \Phi$ is in P.*

Model checking PCTL+EC on MCs with parametric weights. We now consider the case where $\mathscr{M}$ is an ordinary Markov chain augmented with a parametric weight function $wgt\colon S \to \mathbb{Q}[\bar{x}]$. Given a set $T \subseteq S$ such that $\mathrm{Pr}_s^{\mathscr{M}}(\Diamond T) = 1$ for all states $s \in S$, the vector of the expected accumulated weights $e = (\mathrm{E}_s^{\mathscr{M}}(\Diamond\!\!\!\!+ T))_{s\in S}$ is computable as the unique solution of a linear equation system of the form $A \cdot e = b$, where the matrix A is non-parametric, and only the vector b depends on $\bar{x}$. By Lemma 3, $\mathrm{E}_s^{\mathscr{M}}(\Diamond\!\!\!\!+ T)$ is a polynomial of the form $\sum_{t\in S} \beta_{s,t} \cdot wgt(t)$ with $\beta_{s,t} \in \mathbb{Q}$ for all $s \in S$, and can be computed in polynomial time. The expected mean payoff for a given set T is given by $\mathrm{E}_s^{\mathscr{M}}(\mathrm{mp}(T)) = \sum_{\mathrm{BSCC}\, B} \mathrm{Pr}_s^{\mathscr{M}}(\Diamond B) \cdot \mathrm{mp}(B)(T)$ where $\mathrm{mp}(B)(T) = \sum_{t\in T} \zeta_t \cdot wgt_T(t)$ with ζ_t being the steady-state probability for state t inside B (viewed as a strongly connected Markov chain), and $wgt_T(t) = 0$ if $t \notin T$, $wgt_T(t) = wgt(t)$ for $t \in T$. As the transition probabilities are non-parametric, the steady-state probabilities are obtained as the unique solution of a non-parametric linear equation system. So both types of expectations can be computed in polynomial time. Unfortunately, the treatment of formulas with nested expectation operators is more involved. Using the standard computation scheme that processes the DAG-representation of the given PCTL+EC formula in a bottom-up manner to treat inner subformulas first, the combination of polynomial constraints after the consideration of an inner node is still as problematic as in the pMC-case. Using known algorithms for the existential theory of the reals yields the following bound.

▶ **Theorem 11** (Time complexity of PCTL+EC model checking with parametric weights). *Let $\mathscr{M}$ be an MC with parametric weights over k parameters, and Φ a PCTL+EC formula. The problem (MC-E) is solvable in time $\mathscr{O}\big(|\Phi| \cdot \mathrm{poly}\big(size(\mathscr{M}), d\big)^{k\cdot|\Phi|_{\mathbb{E},\mathbb{C}_\mathrm{E}}}\big)$, where $|\Phi|_{\mathbb{E},\mathbb{C}_\mathrm{E}}$ is the number of expectation and expectation comparison operators in the formula, and d the maximum degree of the polynomials assigned as weights.*

If there is only one parameter, the model checking for MCs with parametric weights is solvable in polynomial time for the fragment of PCTL+EC without nested formulas (cf. Theorem 7).

5 Conclusion

In this paper we revisited the model checking problem for pMC and PCTL-like formulas. The purpose of the first part is to draw attention to the fraction-free Gaussian elimination for computing rational functions for reachability probabilities, expected accumulated weights and expected mean payoffs as an alternative to the gcd-based algorithms that have been considered before and are known to suffer from the high complexity of gcd-computations for multivariate polynomials. The experiments with our (not yet optimized) implementation indicate that such an approach can indeed be feasible and beneficial in practice. We thus intend to refine this implementation in future work, including research into further structural heuristics and the potential of a combination with gcd-based simplifications at opportune moments.

In the second part of the paper we studied the complexity of the model checking problem for pMC and PCTL and its extension PCTL+EC by expectation and comparison operators. We identified instances where the model checking problem is NP-hard as well as fragments of PCTL+EC where the model checking problem is solvable in polynomial time. The latter includes the model checking problem for Boolean combinations of probability or expectation conditions for univariate pMCs. This result has been obtained using the fraction-free Gaussian elimination to compute rational functions for reachability probabilities or expected accumulated weights or expected mean payoffs, and polynomial time algorithms for the theory of the reals over a fixed number of variables. As the time complexity of the fraction-free Gaussian elimination is also polynomial for matrices and vectors with a fixed number of parameters and the polynomial-time decidability for the theory of the reals also holds when the number of variables is fixed [4], Theorem 7 also holds for pMC with a fixed number of parameters.

References

[1] Christel Baier & Joost-Pieter Katoen (2008): *Principles of Model Checking*. The MIT Press.

[2] Erwin H. Bareiss (1972): *Computational Solutions of Matrix Problems over an Integral Domain*. *IMA Journal of Applied Mathematics* 10(1), pp. 68–104, doi:10.1093/imamat/10.1.68.

[3] Saugata Basu, Richard Pollack & Marie-Françoise Roy (2008): *Algorithms in Real Algebraic Geometry*. Springer.

[4] Michael Ben-Or, Dexter Kozen & John Reif (1986): *The Complexity of Elementary Algebra and Geometry*. *Journal of Computer and System Sciences* 32(2), pp. 251–264, doi:10.1016/0022-0000(86)90029-2.

[5] Michael Benedikt, Rastislav Lenhardt & James Worrell (2013): *LTL Model Checking of Interval Markov Chains*. In: *19th Int. Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, *LNCS* 7795, Springer, pp. 32–46, doi:10.1007/978-3-642-36742-7_3.

[6] Krishnendu Chatterjee, Koushik Sen & Thomas A. Henzinger (2008): *Model-Checking omega-Regular Properties of Interval Markov Chains*. In: *11th Int. Conference on Foundations of Software Science and Computational Structures (FoSSaCS)*, *LNCS* 4962, Springer, pp. 302–317, doi:10.1007/978-3-540-78499-9_-22.

[7] Frank Ciesinski, Christel Baier, Marcus Größer & Joachim Klein (2008): *Reduction Techniques for Model Checking Markov Decision Processes*. In: *5th Int. Conference on Quantitative Evaluation of Systems (QEST)*, IEEE, pp. 45–54, doi:10.1109/QEST.2008.45.

[8] Conrado Daws (2005): *Symbolic and Parametric Model Checking of Discrete-Time Markov Chains*. In: *1st Int. Colloquium on Theoretical Aspects of Computing (ICTAC)*, *LNCS* 3407, Springer, pp. 280–294, doi:10.1007/978-3-540-31862-0_21.

[9] Christian Dehnert, Sebastian Junges, Nils Jansen, Florian Corzilius, Matthias Volk, Harold Bruintjes, Joost-Pieter Katoen & Erika Ábrahám (2015): *PROPhESY: A PRObabilistic ParamEter SYnthesis Tool*. In: *27th Int. Conference on Computer Aided Verification (CAV)*, *LNCS* 9206, Springer, pp. 214–231, doi:10.1007/978-3-319-21690-4_13.

[10] Christian Dehnert, Sebastian Junges, Joost-Pieter Katoen & Matthias Volk (2017): *A Storm is Coming: A Modern Probabilistic Model Checker*. In: *29th Int. Conference on Computer Aided Verification (CAV)*, *LNCS* 10427, Springer, pp. 592–600, doi:10.1007/978-3-319-63390-9_31.

[11] Antonio Filieri, Carlo Ghezzi & Giordano Tamburrelli (2011): *Run-time efficient probabilistic model checking*. In: *33rd Int. Conference on Software Engineering (ICSE)*, ACM, pp. 341–350, doi:10.1145/1985793.1985840.

[12] Keith O. Geddes, Stephen R. Czapor & George Labahn (1993): *Algorithms for Computer Algebra*. Kluwer.

[13] Ernst Moritz Hahn, Holger Hermanns, Björn Wachter & Lijun Zhang (2010): *PARAM: A Model Checker for Parametric Markov Models*. In: *22nd Int. Conference on Computer Aided Verification (CAV)*, *LNCS* 6174, Springer, pp. 660–664, doi:10.1007/978-3-642-14295-6_56.

[14] Ernst Moritz Hahn, Holger Hermanns & Lijun Zhang (2011): *Probabilistic reachability for parametric Markov models*. Int. Journal on Software Tools for Technology Transfer 13(1), pp. 3–19, doi:10.1007/s10009-010-0146-x.

[15] Hans Hansson & Bengt Jonsson (1994): *A logic for reasoning about time and reliability*. Formal Aspects of Computing 6(5), pp. 512–535, doi:10.1007/bf01211866.

[16] Lisa Hutschenreiter, Christel Baier & Joachim Klein (2017): *Parametric Markov Chains: PCTL Complexity and Fraction-free Gaussian Elimination (extended version)*. Available at `http://wwwtcs.inf.tu-dresden.de/ALGI/PUB/GandALF17/`.

[17] Nils Jansen, Florian Corzilius, Matthias Volk, Ralf Wimmer, Erika Ábrahám, Joost-Pieter Katoen & Bernd Becker (2014): *Accelerating Parametric Probabilistic Verification*. In: 11th Conference on Quantitative Evaluation of Systems (QEST), LNCS 8657, Springer, pp. 404–420, doi:10.1007/978-3-319-10696-0_31.

[18] Bengt Jonsson & Kim Guldstrand Larsen (1991): *Specification and Refinement of Probabilistic Processes*. In: 6th Annual Symposium on Logic in Computer Science (LICS), IEEE, pp. 266–277, doi:10.1109/LICS.1991.151651.

[19] Ravindran Kannan (1985): *Solving Systems of Linear Equations over Polynomials*. Theoretical Computer Science 39, pp. 69–88, doi:10.1016/0304-3975(85)90131-8.

[20] Vidyadhar G. Kulkarni (1995): *Modeling and analysis of stochastic systems*. Chapman & Hall.

[21] Marta Z. Kwiatkowska, Gethin Norman & David Parker (2011): *PRISM 4.0: Verification of Probabilistic Real-Time Systems*. In: 23rd Int. Conference on Computer Aided Verification (CAV), LNCS 6806, Springer, pp. 585–591, doi:10.1007/978-3-642-22110-1_47.

[22] Ruggero Lanotte, Andrea Maggiolo-Schettini & Angelo Troina (2007): *Parametric probabilistic transition systems for system design and analysis*. Formal Aspects of Computing 19(1), pp. 93–109, doi:10.1007/s00165-006-0015-2.

[23] Michael T. McClellan (1973): *The exact solution of systems of linear equations with polynomial coefficients*. Journal of the Association for Computing Machinery 20(4), pp. 563–588, doi:10.1145/321784.321787.

[24] George Nakos, Peter R. Turner & Robert M. Williams (1997): *Fraction-free algorithms for linear and polynomial equations*. ACM SIGSAM Bulletin 31(3), pp. 11–19, doi:10.1145/271130.271133.

[25] Tim Quatmann, Christian Dehnert, Nils Jansen, Sebastian Junges & Joost-Pieter Katoen (2016): *Parameter Synthesis for Markov Models: Faster Than Ever*. In: 14th Int. Symposium on Automated Technology for Verification and Analysis (ATVA), LNCS 9938, Springer, pp. 50–67, doi:10.1007/978-3-319-46520-3_4.

[26] Koushik Sen, Mahesh Viswanathan & Gul Agha (2006): *Model-Checking Markov Chains in the Presence of Uncertainties*. In: 12th Int. Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), LNCS 3920, Springer, pp. 394–410, doi:10.1007/11691372_26.

[27] William Y. Sit (1992): *An Algorithm for Solving Parametric Linear Systems*. Journal of Symbolic Computation 13(4), pp. 353–394, doi:10.1016/S0747-7171(08)80104-6.

Appendix

A Background

Throughout the paper, we shall use LTL-like notations to specify temporal properties for maximal paths and identify them with the corresponding set of maximal paths. Thus, e. g., if $T \subseteq S$ then $\Pr_s^{\mathcal{M}}(\Diamond T)$ denotes the probability of the set of maximal paths π starting in state s that eventually visit a T-state.

A.1 Steady-state probabilities

Given a strongly connected MC $\mathcal{M} = (S, s_{init}, E, P)$, the steady-state probability ζ_t for a state $t \in S$ is the long-run frequency of visiting t along infinite paths. It is well-known that in finite-state strongly connected MC, the steady-state probabilities do not depend on the starting state and can be obtained as the unique solutions of the linear equations

$$\sum_{t \in S} \zeta_t = 1 \quad \text{and} \quad \zeta_t \ = \ \sum_{s \in S} P(s,t) \cdot \zeta_s \quad \text{for each state } s \in S.$$

In matrix notations, $\zeta = (\zeta_t)_{t \in S}$ is the unique (row) vector satisfying $\zeta \cdot A = b$ where matrix A arises from $I - P$ by replacing the column of one state t with the column vector $(1, 1, \ldots, 1)$ and b is the row vector $(0, 0, \ldots, 0, 1)$.

A.2 Mean payoff

Given an MC $\mathcal{M} = (S, s_{init}, E, P)$ without traps that is augmented with a weight function $wgt : S \to \mathbb{Q}$, and T a subset of the set of states, the *mean payoff* along an infinite path in $\mathcal{M}$ with respect to T is the mean weight accumulated along the path when setting all weights assigned to states not in T to zero. Formally, if $\pi = s_0 s_1 s_2 \ldots \in \mathrm{Paths}_{inf}(s_0)$ then

$$\mathrm{mp}(T)(\pi) \ = \ \limsup_{n \to \infty} \frac{1}{n} \cdot \sum_{i=0}^{n} wgt_T(s_i),$$

where $wgt_T(s) = wgt(s)$ if $s \in T$, and $wgt_T(s) = 0$ if $s \notin T$. As almost all such paths will end up within a BSCC of $\mathcal{M}$ within finitely many steps, it suffices to consider their behaviour within this BSCC. It is known that for almost all paths π eventually entering a BSCC B, the mean payoff is

$$\mathrm{mp}(T)(\pi) \ = \ \sum_{s \in B} \zeta_s \cdot wgt_T(s) \ \stackrel{\text{def}}{=} \ \mathrm{mp}(T)(B)$$

where ζ_s is the steady-state probability. Note that the value $\mathrm{mp}(T)(B)$ only depends on B.

$\mathrm{E}_s^{\mathcal{M}}(\mathrm{mp}(T))$ denotes the expectation of the random variable $\mathrm{mp}(T)$ from state s. With the above observation we obtain:

$$\mathrm{E}_s^{\mathcal{M}}(\mathrm{mp}(T)) \ = \ \sum_{B \in \mathscr{B}} \mathrm{mp}(T)(B) \cdot \Pr_s^{\mathcal{M}}(\Diamond B)$$

where $\mathscr{B}$ denotes the set of BSCCs in $\mathcal{M}$. Thus, the expected mean payoffs $\mathrm{E}_s^{\mathcal{M}}(\mathrm{mp}(T))$ are computable by solving the linear equation systems for the steady-state probabilities for each BSCC of $\mathcal{M}$ (see Section A.1) and the linear equation systems for the reachability probabilities for the BSCCs.

A.3 Accumulated weight

Given an MC $\mathscr{M}$ with weights, and T a subset of the set of states, the *accumulated weight* along a path in $\mathscr{M}$ until reaching a state in T is the sum of weights assigned to states before the first state in T.

We first introduce the accumulated weight of finite paths. Given a finite path $s_0 s_1 \ldots s_m$, we define

$$wgt(s_0 s_1 \ldots s_m) \ = \ wgt(s_0) + wgt(s_1) + \ldots + wgt(s_m).$$

We now define $\diamondplus T$ as a random variable that maps maximal paths to values in $\mathbb{R} \cup \{\pm\infty\}$. Let $\pi = s_0 s_1 s_2 \ldots$ be a maximal path (i. e., infinite or finite and ending in a trap). If $\pi \models_{\mathscr{M}} \Diamond T$ then

$$(\diamondplus T)^{\mathscr{M}}(\pi) = wgt(s_0\, s_1 \ldots s_{n-1})$$

where n is the smallest index with $s_n \in T$. (Note that $(\diamondplus T)(\pi) = 0$ if $s_0 \in T$.) If $\pi \not\models_{\mathscr{M}} \Diamond T$ then

$$(\diamondplus T)(\pi) = \limsup_{n\to\infty} wgt(s_0\, s_1 \ldots s_n) \in \mathbb{Q} \cup \{\pm\infty\} \qquad \text{for } \pi \text{ infinite, and}$$

$$(\diamondplus T)(\pi) = wgt(s_0 s_1 \ldots s_m) + \limsup_{n\to\infty} n \cdot wgt(s_m) \in \mathbb{Q} \cup \{\pm\infty\} \qquad \text{for } \pi \text{ finite.}$$

Now, the *expected accumulated weight* from s is defined as the expectation of the random variable $\diamondplus T$. So, if $\Pr_s^{\mathscr{M}}\big\{\, \pi \in \mathrm{Paths}(s) : \pi \models \Diamond T \,\big\} = 1$ then the expected accumulated weight until reaching T from s is well-defined. Unfortunately, if $\Pr_s^{\mathscr{M}}(\Diamond T) < 1$ then the expectation of $\diamondplus T$ is not generally well-defined. This applies, e. g., to the weighted MC consisting of three states s,t,u with $P(s,t) = P(s,u) = \frac{1}{2}$, $P(t,t) = P(u,u) = 1$, $wgt(s) = 0$, $wgt(t) = 1$, $wgt(u) = -1$, and $s,t,u \not\models_{\mathscr{M}} T$.

To ensure well-definedness of the expected accumulated weight we suppose that either a T-state is reached almost surely from each state, or all weights are non-negative. The latter case is reducible to the former. This is a consequence of the following facts. We have $\mathrm{E}_s^{\mathscr{M}}\big(\diamondplus T\big) = +\infty$ if $\mathscr{M}$ contains a BSCC B that is reachable from s and satisfies either $\mathrm{mp}(T)(B) > 0$, or $\mathrm{mp}(T)(B) = 0$ and B does not contain a positive cycle (i. e., a cycle π with $wgt(\pi) > 0$). Suppose now that each BSCC B reachable from s satisfies $\mathrm{mp}(T)(B) = 0$ and $wgt(\pi) = 0$ for each cycle π in B. Then, $\mathrm{E}_s^{\mathscr{M}}(\diamondplus T) = \mathrm{E}_s^{\mathscr{M}'}(\diamondplus a)$ where $\mathscr{M}'$ denotes the MC resulting from $\mathscr{M}$ by (i) removing all states that are not reachable from s, (ii) turning all states in $s \in T$ into trap states and (iii) assigning weight 0 to all states that belong to a BSCC. The labelling function in $\mathscr{M}'$ assigns a to the states in T and all states that are contained in some BSCC (and to no other state). Note that $\Pr_s^{\mathscr{M}'}(\Diamond a) = 1$. This observation allows us to consider expected accumulated weight only for those MCs where T will be reached almost surely from each state s, in which case we can handle also weight functions with positive and negative values.

If the expected accumulated weight is well-defined for each state $s \in S$, it can be computed by solving the linear equation system resulting from the equations

$$\mathrm{E}_s^{\mathscr{M}}(\diamondplus T) - \sum_{t\in S} P(s,t) \cdot \mathrm{E}_t^{\mathscr{M}}(\diamondplus T) = wgt(s)$$

for $s \in S \setminus T$, with $\mathrm{E}_t^{\mathscr{M}}(\diamondplus T) = 0$ for all $t \in T$ in mind.

A.4 Semantics of the expectation operators

As introduced in the paper, we augment PCTL with an expectation operator and random variables. Furthermore, we also add a comparison operator for expectations. Given a Markov chain $\mathscr{M}$, the semantics $\rho^{\mathscr{M}}$ of the terms $\rho = \diamondplus \Phi$ or $\rho = \mathrm{mp}(T)$ are random variables.

For the mean payoff operator, we suppose that the underlying Markov chain has no traps. This ensures that all maximal paths in $\mathscr{M}$ are infinite and the well-definedness of the mean payoff function $\mathrm{mp}\,T$ for each $T \subseteq S$. The random variable $\mathrm{mp}(\Phi)^{\mathscr{M}}$ assigned to the PCTL+EC term $⟠\Phi$ in $\mathscr{M}$ is given by $\mathrm{mp}(\Phi)^{\mathscr{M}} = \mathrm{mp}(T)$ for $T = \mathrm{Sat}_{\mathscr{M}}(\Phi)$. The semantics of the term $⟠\Phi$ is the random variable $⟠T$ where $T = \mathrm{Sat}_{\mathscr{M}}(\Phi)$.

The semantics of the expectation operator $\mathbb{E}_{\bowtie r}$ and the comparison operator $\mathbb{C}_{\mathrm{E}}(\rho_1, \bowtie, \rho_2)$ is then defined by:

$$\begin{array}{lll} s \models_{\mathscr{M}} \mathbb{E}_{\bowtie r}(\rho) & \text{iff} & \mathrm{E}_s^{\mathscr{M}}\big(\rho^{\mathscr{M}}\big) \bowtie r \\ s \models_{\mathscr{M}} \mathbb{C}_{\mathrm{E}}(\rho_1, \bowtie, \rho_2) & \text{iff} & \mathrm{E}_s^{\mathscr{M}}\big(\rho_1^{\mathscr{M}}\big) \bowtie \mathrm{E}_s^{\mathscr{M}}\big(\rho_2^{\mathscr{M}}\big) \end{array}$$

where $\mathrm{E}_s^{\mathscr{M}}(\rho^{\mathscr{M}})$ denotes the expectation of $\rho^{\mathscr{M}}$ (if existent).

For the semantics of the expectation operator $\mathbb{E}_{\bowtie r}(⟠\Phi)$, we need some restrictions on $\mathscr{M}$ or its weight function to ensure the existence of the expectation of $(⟠\Phi)^{\mathscr{M}}$. As stated before, this can either be the requirement that all weights are non-negative (i. e., $wgt(s) \geqslant 0$ for all states s) or that $\mathrm{Pr}_s^{\mathscr{M}}(\Diamond\Phi) = 1$ for all states s.

Here and in what follows, we often identify the terms for random variables ρ with their meanings $\rho^{\mathscr{M}}$. For instance, $\mathrm{E}_s^{\mathscr{M}}(⟠\Phi)$ stands for $\mathrm{E}_s^{\mathscr{M}}\big((⟠\Phi)^{\mathscr{M}}\big)$. Likewise, $\mathrm{Pr}_s^{\mathscr{M}}(\Diamond\Phi)$ stands shortly for $\mathrm{Pr}_s^{\mathscr{M}}(\Diamond T)$ where $T = \mathrm{Sat}_{\mathscr{M}}(\Phi) \overset{\text{def}}{=} \{s \in S : s \models_{\mathscr{M}} \Phi\}$,

B Fraction-free Gaussian elimination

We use the same notations as in Section 3.

B.1 Properties

Linear equation systems with polynomial coefficients. We consider here the case where the degree of the polynomials is at most one, that is, all coefficients of the matrix A and the vector b have the form $\beta + \alpha_1 x_1 + \ldots + \alpha_k x_k$ with $\beta, \alpha_1, \ldots, \alpha_k \in \mathbb{Q}$.

▶ **Lemma 12.** *If $d = 1$ (where d is the maximum degree of the coefficients in A and b) then the number of monomials of the polynomials f_i and g_i in the rational functions $p_i = f_i/g_i$, $i = 1, \ldots, n$, obtained as solutions of $A \cdot p = b$, is at most $\binom{n+k}{k}$, where k is the number of parameters.*

Proof. Recall that we observed in the main paper (Section 3): if the coefficients of A and b have at most degree d, the Leibniz formula implies that f_i and g_i have at most degree $n \cdot d$. Thus, with $d = 1$ the polynomials f_i and g_i have degree at most n. The upper bound stated in Lemma 12 is obtained by simple combinatorics. Furthermore, an estimate for this upper bound is $\left(2\frac{n}{k}\right)^k \leq \binom{n+k}{k} \leq \left(3\frac{n+k}{k}\right)^k$. So the number of monomials is at most exponential in k. ◀

▶ **Lemma 13** (Lemma 1 in the main part)**.** *There is a family $(\mathfrak{M}_k)_{k \geqslant 2}$ of acyclic linear pMCs where $\mathfrak{M}_k$ has k parameters and $n = k+3$ states, including distinguished states s_0 and goal such that $\mathrm{Pr}_{s_0}^{\mathfrak{M}(\overline{x})}(\Diamond goal)$ is a polynomial for which even the shortest sum-of-monomial representation has 2^k monomials.*

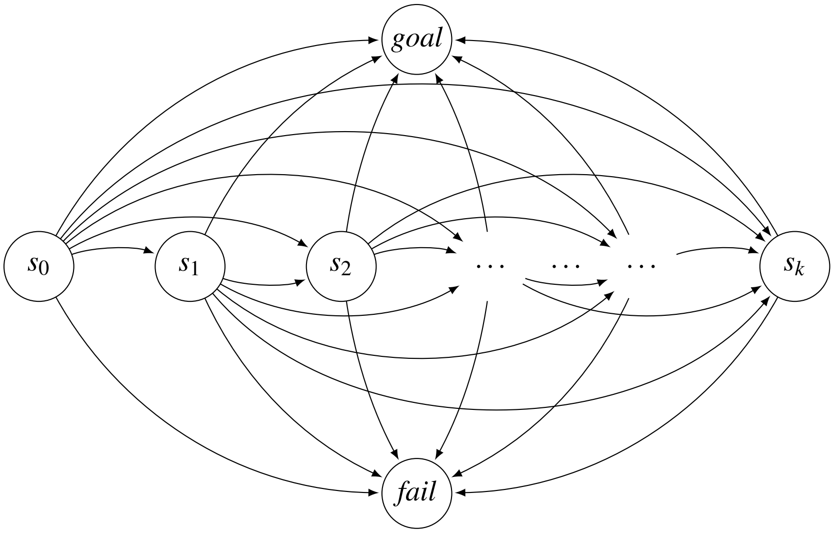


Figure 1 – Graph structure of an acyclic parametric Markov chain on parameters $x_1,\ldots,x_k$, with transition probabilities $\frac{1}{k+2}$ for transitions from s_0 to any other state, x_i for transitions from state s_i, $i=1,\ldots,k$, to *goal*, and $\frac{1-x_i}{k-i+1}$ for transitions from s_i to *fail* or a state s_j with $j>i$, and 0 in any other case.

Proof. Let $\mathfrak{M} = (S, s_{init}, E, \mathbf{P})$ be a pMC on $(x_1,\ldots,x_k)$ with $S = \{s_0,\ldots,s_k, fail, goal\}$, $s_{init} = s_0$, and

$$\mathbf{P}(s,t) = \begin{cases} \frac{1}{k+2} & \text{if } s = s_0,\, t \neq s_0 \\ x_i & \text{if } s = s_i,\, t = goal,\, 0 < i \leq k \\ \frac{1-x_i}{k-i+1} & \text{if } s = s_i,\, t = s_j \text{ with } 0 < i < j \leq k, \text{ or } s = s_i,\, t = fail,\, 0 < i \leq k \\ 0 & \text{otherwise} \end{cases},$$

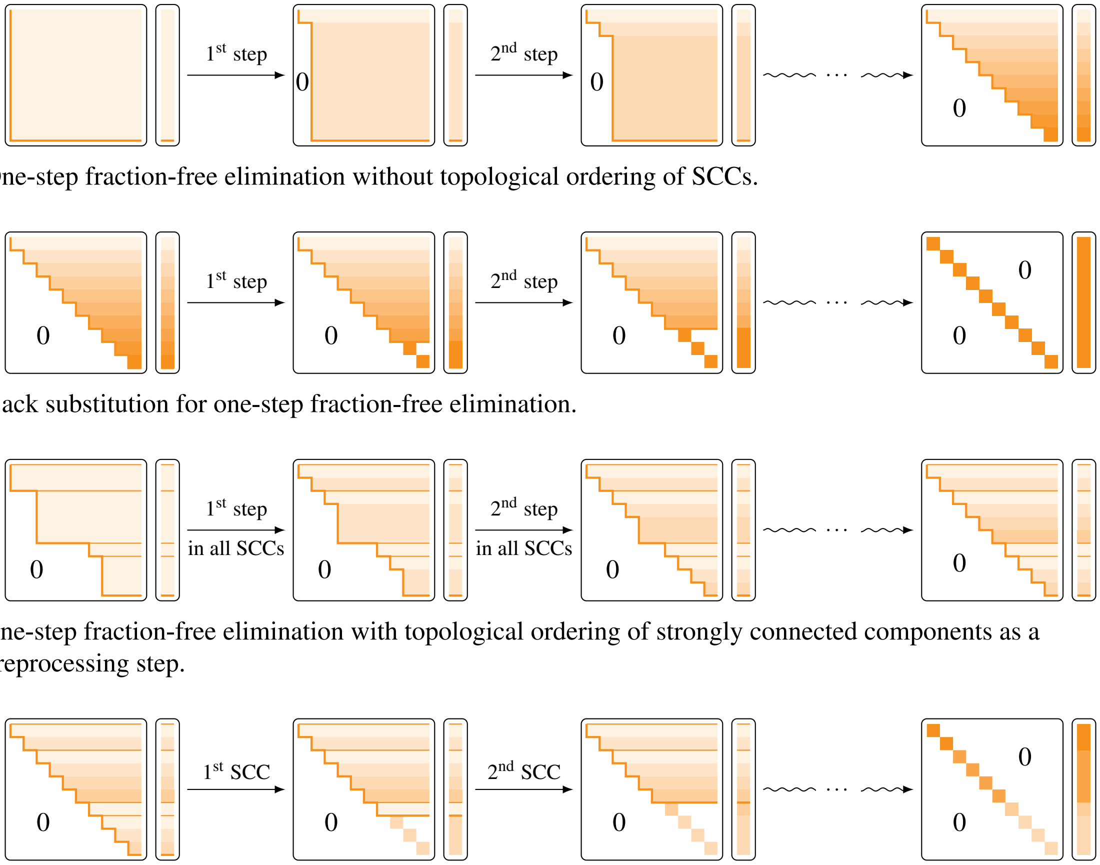
whose graph $G = (S,E)$ is depicted in Figure 1. Then the probability of reaching *goal* from the initial state is:

$$\mathrm{Pr}^{\mathfrak{M}(\overline{x})}_{s_{init}}(\Diamond\, goal) \;=\; \frac{1}{n+2} + \frac{1}{n+2} \cdot \sum_{\substack{1 \leq m \leq n \\ i_1 < \ldots < i_m \leq n}} x_{i_m} \cdot \prod_{j=1}^{m-1} \frac{1 - x_{i_j}}{n - i_j + 1}$$

For any combination of indices $(i_1,\ldots,i_m)$ the highest order monomial in each summand contains the parameters in the form $\prod_{j=1}^{m} x_{i_j}$. Therefore, any combination of parameters occurs as highest order monomial in one of the summands.

Two summands corresponding to the index combinations $(i_1,\ldots,i_m)$ and $(j_1,\ldots,j_{m'})$ can only have common non-zero monomials if $i_m = j_{m'}$. Observe that any common monomials consisting of k parameters have to have the same sign, namely $(-1)^{k-1}$. So they cannot cancel out. Thus, the rational function for $\mathrm{Pr}^{\mathfrak{M}(\overline{x})}_{s_{init}}(\Diamond\, goal)$ has the form $\sum_{I \subseteq \{1,\ldots,n\}} \alpha_I \cdot \prod_{i \in I} x_i$ with non-zero coefficients $\alpha_I \in \mathbb{Q}$. The number of monomials with non-zero coefficients is therefore 2^n, that is, exponential in the number of parameters. ◀

One-step fraction-free Gaussian elimination. Figure 2 provides an illustration of the behaviour and resulting maximal degrees of the polynomials, both for the one-step fraction-free approach and for the one additionally relying on an SCC decomposition and topological sorting.



(a) One-step fraction-free elimination without topological ordering of SCCs.

(b) Back substitution for one-step fraction-free elimination.

(c) One-step fraction-free elimination with topological ordering of strongly connected components as a preprocessing step.

(d) Back substitution for one-step fraction-free elimination with topological ordering of SCCs.

Figure 2 – General form of coefficient matrix and vector without and with topological ordering of strongly connected components, and behaviour when applying one-step fraction-free elimination. The intensity of the background colour indicates the maximum total degree of the polynomials, with darker colours representing higher degree.

▶ **Lemma 14** (Lemma 2 in the main part). *Let $\mathfrak{M}$ be a polynomial pMC over a single parameter and T a set of states. Then, the rational functions for the reachability probabilities $\Pr_s^{\mathfrak{M}}(\Diamond T)$ are computable in polynomial time. Analogously, rational functions for the expected accumulated weight until reaching T or the expected mean payoff are computable in polynomial time.*

Proof. As the calculation of the reachability probabilities as well as the expected accumulated weight can always be done by solving a linear equation system with the transition probabilities in the coefficient matrix and vector, or two such systems when considering the expected mean payoff, they all fall into the same complexity class. Here we only consider the reachability probabilities.

Note that for $k = 1$ and $d = \max_{s,t \in S} \deg(\mathbf{P}(s,t))$, the degree of the polynomials $a_{i,j}^{(m)}$ and $b_j^{(m)}$ computed by the fraction-free Gaussian elimination (Algorithm 1) for reachability probabilities are bounded by md. Hence, they have representations as sums of at most $md+1$ monomials. In particular, the degree and representation size of the final polynomials $f_s = b_s^{(n)}$ and $g_s = a_{s,s}^{(n)}$ in the rational functions $\Pr_s^{\mathfrak{M}(x)}(\Diamond goal) = f_s/g_s$ is in $\mathcal{O}(nd)$ when n is the number of states in $\mathfrak{M}$.

Therefore, these functions can be computed in polynomial time. ◀

▶ **Lemma 15** (Lemma 3 in the main part). *Let $A \cdot p = b$ be a parametric linear equation system as defined above where A is parameter-free. Then the solutions p_i, $i = 1, \dots, n$ can be computed in polynomial time. The solutions are of the form $p_i = \sum_{i=1}^{n} \beta_i \cdot b_i$ with $\beta_i \in \mathbb{Q}$, $i = 1, \dots, n$.*

Proof. Using one-step fraction-free Gaussian elimination, the only interesting step in the algorithm (cf. Algorithm 1) concerns the calculation of the $b_i^{(m)}$ in each step, as all other computations are done with rationals only, for which Gaussian elimination is known to be in P.

Since

$$b_i^{(m)} = \left(b_i^{(m-1)} a_{m,m}^{(m-1)} - a_{i,m}^{(m-1)} b_m^{(m-1)}\right) / a_{m-1,m-1}^{(m-1)},$$

$b_i^{(m)}$ is a linear combination of the previous $b_j^{(m-1)}$, and each step can be performed in polynomial time. Therefore, at the end of the triangulation, all $b^{(n-1)}$ are of the form $b^{(n-1)} = \sum_{i=1}^{n} \sigma_i \cdot b_i$.

The same argument applies to the back substitution. So the right-hand side of the equation system after diagonalisation contains only linear combinations of the original b_i. As there are only rationals on the left-hand side, the results that are returned are also of the form $p_i = \sum_{i=1}^{n} \beta_i \cdot b_i$ with $\beta_i \in \mathbb{Q}$, $i = 1, \dots, n$. As Gaussian elimination without parameters is in P, and the computations for the right-hand side can also be done in polynomial time, the p_i can be computed in polynomial time. ◀

B.2 Experiments

We provide here some more details on the experiments with the benchmarks from [9]. The model sources, properties and log files for the experiments are available online at `http://wwwtcs.inf.tu-dresden.de/ALGI/PUB/GandALF17/`. All experiments were carried out using a time out of 30 minutes, with a memory limit of 30GB.

Complete pMC. We considered the family of pMCs $\mathfrak{M}_n = (S, s_{\mathit{init}}, E, \mathbf{P}, \mathfrak{C})$ with n regular states and a *goal* and *fail* state, $S = \{s_1, \dots, s_n, \mathit{fail}, \mathit{goal}\}$, and with initial state $s_{\mathit{init}} = s_1$. The graph structure for the regular states is complete and the probability for going from state s_i to s_j is given by a parameter $x_{i,j}$, i. e., $\mathbf{P}(s_i, s_j) = x_{i,j}$ for $1 \leq i, j \leq n$. The probability of going to the *goal* state is similarly encoded using a parameter $x_{i,\mathit{goal}}$, i. e., $\mathbf{P}(s_i, \mathit{goal}) = x_{i,\mathit{goal}}$ for $1 \leq i \leq n$, and $\mathbf{P}(s_i, \mathit{fail}) = 1 - \mathbf{P}(s_i, \mathit{goal}) - \sum_{1 \leq j \leq n} \mathbf{P}(s_i, s_j)$. Overall, the pMC is defined on parameters $(x_{1,1}, \dots, x_{1,n}, x_{1,\mathit{goal}}, \dots, x_{n,1}, \dots, x_{n,n}, x_{n,\mathit{goal}})$, i. e., on $n \cdot (n+1)$ parameters. For $\mathfrak{M}_n$, we computed the probability of reaching the *goal* state from the initial state, i. e., $\Pr_{s_{\mathit{init}}}^{\mathfrak{M}_n}(\Diamond \mathit{goal})$.

Multi-parameter Israeli-Jalfon self-stabilizing. In the Israeli-Jalfon self-stabilizing protocol, with the model taken from the PRISM case study repository, [4] multiple processes in a ring can send tokens to each other, with the protocol ensuring that almost surely eventually a stable situation is reached, i. e., a single token remains. We consider a variant with n processes and an initial number of tokens k, where non-deterministic scheduling is replaced by uniform scheduling to obtain a pMC. The uniform probabilistic choice between sending the token to the left or right neighbour of a process in the original model is replaced by a biased, parametrized choice, i. e., there are n parameters x_i specifying the probability for process i of sending to the right instead of the left neighbour. An initial gadget ensures that the k initial

[4] `http://www.prismmodelchecker.org/casestudies/self-stabilisation.php#ij`

Table 4: Statistics for the benchmarks of [9], “BRP”, “Crowds”, “Zeroconf”. Matrix rows and number of distinct parameters, as well as time for solving the parametric equation system per solver.

model	rows	param.	*eigen*	*state-elim*	*GE-ff*	*red(GE-ff)*
BRP (128,2), weak-bisim	768	2	1.39 s	1.20 s	0.69 s	0.03 s
BRP (128,5), weak-bisim	1536	2	6.94 s	9.01 s	3.94 s	0.20 s
BRP (256,2), weak-bisim	1536	2	5.92 s	4.70 s	2.84 s	0.15 s
BRP (256,5), weak-bisim	3072	2	30.63 s	37.05 s	16.81 s	1.11 s
BRP (128,2)	3964	2	5.52 s	6.43 s	2.12 s	0.18 s
BRP (128,5)	8950	2	59.81 s	112.91 s	14.21 s	1.43 s
BRP (256,2)	7932	2	21.83 s	27.91 s	8.83 s	0.88 s
BRP (256,5)	17910	2	257.71 s	499.42 s	61.79 s	8.03 s
Crowds (3,5), weak-bisim	40	2	0.08 s	0.06 s	0.02 s	0.13 s
Crowds (5,5), weak-bisim	40	2	0.08 s	0.06 s	0.02 s	0.11 s
Crowds (10,5), weak-bisim	40	2	0.08 s	0.06 s	0.02 s	0.11 s
Crowds (3,5)	715	2	0.99 s	0.80 s	11.44 s	63.39 s
Crowds (5,5)	2928	2	6.36 s	5.51 s	1271.95 s	time-out
Crowds (10,5)	25103	2	139.82 s	173.15 s	time-out	—
Zeroconf (1000)	1002	2	83.22 s	45.24 s	50.11 s	11.15 s
Zeroconf (10000)	10002	2	time-out	time-out	time-out	time-out

tokens are distributed uniformly between the processes. We then compute the expected number of steps until reaching the stable situation.

Further statistics for the experiments using the benchmark models of [9]. Table 4 lists the full statistics for the *brp*, *crowds* and *zeroconf* instances. As can be seen, the fraction-free approach significantly outperformed Storm’s standard solvers for the *brp* instances. As the models are acyclic, this is most likely due to differences in the order in which the rational functions are computed and the resulting differences in the size of intermediate results.

Table 5 lists statistics for the *nand* instances with a reachability probability computation while Table 6 details the experiments with the same instances but an expected accumulated reward computation. For the former, weak bisimulation quotienting provides the most beneficial reduction, while for the latter strong bisimulation quotienting is better. Table 7 lists statistics for the *egl* case study. Here, using bisimulation quotienting, the state space is reduced significantly and leads to negligible computation times for the linear equation system solving. In the case without quotienting, the state space is quite large, with *GE-ff* being at a significant disadvantage already due to the dense matrix representation used in its implementation.

C Complexity of the PCTL model checking problem

We use the notation introduced in Section 4. Recall that, for a possibly augmented pMC $\mathfrak{M}$, $X = X_{\mathfrak{M}}$ denotes the set of admissible valuations, and χ is the conjunction of the polynomial constraints in $\mathfrak{C}$ as well as the constraints $\sum_{t \in S} \mathbf{P}(s,t) = 1$ for each non-trap state $s \in S$, and $0 < \mathbf{P}(s,t)$ for each edge $(s,t) \in E$. So $\overline{\xi} \models \chi$ if and only if $\overline{\xi}$ is admissible, i. e., $\overline{\xi} \in X$.

Table 5: Statistics for the benchmarks of [9], “NAND”, for a reachability probability. Matrix rows and number of distinct parameters, as well as time for solving the parametric equation system per solver.

model	rows	param.	*eigen*	*state-elim*	*GE-ff*	*red(GE-ff)*
NAND (10,1)	4660	2	1.07 s	0.46 s	1.07 s	0.03 s
NAND (10,2)	11590	2	3.42 s	1.30 s	5.64 s	0.11 s
NAND (10,3)	18520	2	6.88 s	2.52 s	13.49 s	0.24 s
NAND (10,4)	25450	2	11.40 s	3.91 s	25.26 s	0.40 s
NAND (10,5)	32380	2	17.18 s	5.65 s	41.03 s	0.62 s
NAND (20,1)	49040	2	26.44 s	8.65 s	134.61 s	0.56 s
NAND (20,2)	125650	2	103.76 s	27.09 s	mem-out	—
NAND (20,3)	202260	2	222.04 s	50.09 s	mem-out	—
NAND (20,4)	278870	2	534.06 s	81.30 s	mem-out	—
NAND (20,5)	355480	2	1103.26 s	120.37 s	mem-out	—
NAND (10,1), weak-bisim	1610	2	0.58 s	0.66 s	0.31 s	0.01 s
NAND (10,2), weak-bisim	3830	2	1.80 s	2.58 s	1.22 s	0.04 s
NAND (10,3), weak-bisim	6050	2	3.67 s	5.74 s	2.80 s	0.09 s
NAND (10,4), weak-bisim	8270	2	7.67 s	9.68 s	5.18 s	0.15 s
NAND (10,5), weak-bisim	10490	2	10.09 s	14.52 s	8.35 s	0.22 s
NAND (20,1), weak-bisim	20870	2	18.38 s	42.38 s	18.54 s	0.26 s
NAND (20,2), weak-bisim	52710	2	68.97 s	203.33 s	mem-out	—
NAND (20,3), weak-bisim	84550	2	166.21 s	442.95 s	294.88 s	2.54 s
NAND (20,4), weak-bisim	116390	2	320.40 s	785.83 s	mem-out	—
NAND (20,5), weak-bisim	148230	2	548.72 s	1173.71 s	mem-out	—

Table 6: Statistics for the benchmarks of [9], “NAND”, for an expected accumulated reward. Matrix rows and number of distinct parameters, as well as time for solving the parametric equation system per solver.

model	rows	param.	*eigen*	*state-elim*	*GE-ff*	*red(GE-ff)*
NAND (10,1)	7381	2	0.77 s	0.19 s	1.85 s	0.02 s
NAND (10,2)	14311	2	1.95 s	0.56 s	7.09 s	0.06 s
NAND (10,3)	21241	2	4.12 s	1.27 s	15.03 s	0.13 s
NAND (10,4)	28171	2	7.34 s	2.38 s	27.77 s	0.25 s
NAND (10,5)	35101	2	11.47 s	3.69 s	42.29 s	0.42 s
NAND (20,1)	78311	2	15.99 s	1.91 s	mem-out	—
NAND (20,2)	154921	2	46.91 s	6.47 s	mem-out	—
NAND (20,3)	231531	2	101.09 s	16.50 s	mem-out	—
NAND (20,4)	308141	2	190.81 s	34.79 s	mem-out	—
NAND (20,5)	384751	2	322.69 s	59.24 s	mem-out	—
NAND (10,1), strong-bisim	5381	2	0.45 s	0.13 s	0.98 s	0.01 s
NAND (10,2), strong-bisim	10421	2	1.23 s	0.37 s	3.56 s	0.04 s
NAND (10,3), strong-bisim	15461	2	2.92 s	0.84 s	7.93 s	0.10 s
NAND (10,4), strong-bisim	20501	2	6.57 s	1.55 s	14.40 s	0.19 s
NAND (10,5), strong-bisim	25541	2	8.92 s	2.43 s	22.48 s	0.31 s
NAND (20,1), strong-bisim	63311	2	13.28 s	1.45 s	125.12 s	0.18 s
NAND (20,2), strong-bisim	125341	2	34.84 s	4.81 s	mem-out	—
NAND (20,3), strong-bisim	187371	2	73.68 s	12.30 s	mem-out	—
NAND (20,4), strong-bisim	249401	2	218.72 s	25.44 s	mem-out	—
NAND (20,5), strong-bisim	311431	2	357.45 s	44.22 s	mem-out	—

Table 7: Statistics for the benchmarks of [9], “egl”, for an expected accumulated reward. Matrix rows and number of distinct parameters, as well as time for solving the parametric equation system per solver.

model	rows	param.	*eigen*	*state-elim*	*GE-ff*	*red(GE-ff)*
EGL(5,2)	33789	1	0.17 s	0.04 s	46.34 s	0.01 s
EGL(5,4)	74749	1	0.29 s	0.08 s	225.38 s	0.03 s
EGL(8,2)	3342333	1	18.52 s	3.67 s	mem-out	—
EGL(8,4)	7536637	1	35.04 s	54.60 s	mem-out	—
EGL(5,2), strong-bisim	238	1	0.00 s	0.00 s	0.00 s	0.00 s
EGL(5,4), strong-bisim	478	1	0.00 s	0.00 s	0.01 s	0.00 s
EGL(8,2), strong-bisim	466	1	0.00 s	0.00 s	0.01 s	0.00 s
EGL(8,4), strong-bisim	946	1	0.00 s	0.00 s	0.03 s	0.00 s

C.1 Computing all satisfying parameter valuations

We now provide some more details on the algorithm that generates symbolic representations of the satisfaction functions for PCTL+EC formulas. Given the DAG representation of the PCTL formula Φ, we follow the standard model checking procedure for CTL-like branching-time logics, and compute $\mathtt{Sat}_{\mathfrak{M}}(\Psi)$ for the sub-formulas Ψ assigned to the nodes in the DAG for Φ in a bottom-up manner. The base case is the treatment of the nodes in the DAG for Φ labelled by an atomic proposition a or the formula $\mathtt{true}$, that is, the leaf nodes of the DAG:

$$\begin{aligned} \mathtt{Sat}_{\mathfrak{M}}(\mathtt{true}) &= \big\{(\chi, S)\big\} \\ \mathtt{Sat}_{\mathfrak{M}}(a) &= \big\{\, (\chi, \{s \in S : a \in \mathscr{L}(s)\})\,\big\}. \end{aligned}$$

Consider now an inner node v of the DAG labelled by formula Ψ. Under the assumption that the child nodes of v have already been treated, i. e., the satisfaction sets of the proper subformulas of Ψ are known, we obtain the following.

- If $\Psi = \neg\Psi'$ then $\mathtt{Sat}_{\mathfrak{M}}(\Psi) \;=\; \big\{\; (\gamma, S \setminus T) \;:\; (\gamma, T) \in \mathtt{Sat}_{\mathfrak{M}}(\Psi') \;\big\}$.
- If $\Psi = \Psi_1 \wedge \Psi_2$ then
$$\mathtt{Sat}_{\mathfrak{M}}(\Psi) \;\;=\;\; \big\{\; (\gamma_1 \wedge \gamma_2, T_1 \cap T_2) \;:\; (\gamma_i, T_i) \in \mathtt{Sat}_{\mathfrak{M}}(\Psi_i), i = 1,2 \;\big\}.$$
- If $\Psi = \mathbb{P}_{\bowtie c}(\bigcirc\Psi')$ then
$$\mathtt{Sat}_{\mathfrak{M}}(\Psi) \;\;=\;\; \big\{\; (\gamma \wedge \delta_{\gamma,T,R}, R) \;:\; (\gamma, T) \in \mathtt{Sat}_{\mathfrak{M}}(\Psi'),\; R \subseteq S' \;\big\}$$
where S' denotes the set of states that are not traps and $\delta_{\gamma,T,R}$ is the conjunction of the following constraints:
$$\begin{array}{lll} \sum\limits_{t \in T} \mathbf{P}(s,t) \;\bowtie\; c & & \text{for each state } s \in R, \\ \sum\limits_{t \in T} \mathbf{P}(s,t) \;\not\bowtie\; c & & \text{for each state } s \in S' \setminus R. \end{array}$$
- If $\Psi = \mathbb{P}_{\bowtie c}(\Psi_1 \mathbin{\mathrm{U}} \Psi_2)$ then
$$\mathtt{Sat}_{\mathfrak{M}}(\Psi) \;\;=\;\; \big\{\; (\gamma_1 \wedge \gamma_2 \wedge \delta_{\gamma_1,T_1,\gamma_2,T_2,R}) \;:\; (\gamma_1, T_1) \in \mathtt{Sat}_{\mathfrak{M}}(\Psi_1),\; (\gamma_2, T_2) \in \mathtt{Sat}_{\mathfrak{M}}(\Psi_2),\; R \subseteq S \;\big\}$$
where $\delta_{\gamma_1,T_1,\gamma_2,T_2,R}$ is the conjunction of the following constraints:
$$\begin{array}{lll} \mathrm{Pr}^{\mathfrak{M}}_s(T_1 \mathbin{\mathrm{U}} T_2) \;\bowtie\; c & & \text{for each state } s \in R, \\ \mathrm{Pr}^{\mathfrak{M}}_s(T_1 \mathbin{\mathrm{U}} T_2) \;\not\bowtie\; c & & \text{for each state } s \in S \setminus R. \end{array}$$
Here, $\mathrm{Pr}^{\mathfrak{M}}_s(T_1 \mathbin{\mathrm{U}} T_2)$ is the rational function that has been computed using (i) a graph analysis to determine the set U of states s with $s \models \exists(T_1 \mathbin{\mathrm{U}} T_2)$ and (ii) fraction-free Gaussian elimination (Section 3) to compute the rational functions $\mathrm{Pr}^{\mathfrak{N}}_s(\Diamond T_2)$ in the pMC $\mathfrak{N}$ resulting from $\mathfrak{M}$ by turning the states in $(S \setminus U) \cup T_2$ into traps. If f_s and g_s are polynomials computed by the fraction-free Gaussian elimination such that $\mathrm{Pr}^{\mathfrak{M}}_s(T_1 \mathbin{\mathrm{U}} T_2) = f_s / g_s$ then $\mathrm{Pr}^{\mathfrak{M}}_s(T_1 \mathbin{\mathrm{U}} T_2) \bowtie c$ is a shortform notation for $f_s - c \cdot g_s \bowtie 0$.
- If $\Psi = \mathbb{C}_{\mathrm{Pr}}(\psi_1, \bowtie, \psi_2)$ then the further computation depends of course on ψ_1, and ψ_2. Here, we will only deal with the case $\psi_i = \bigcirc\Psi_i$, $i = 1,2$, but the cases for U, and combinations of both work similarly.
$$\mathtt{Sat}_{\mathfrak{M}}(\Psi) \;\;=\;\; \big\{\; (\gamma_1 \wedge \gamma_2 \wedge \delta_{\gamma_1,T_1,\gamma_2,T_2,R}) \;:\; (\gamma_1, T_1) \in \mathtt{Sat}_{\mathfrak{M}}(\Psi_1),\; (\gamma_2, T_2) \in \mathtt{Sat}_{\mathfrak{M}}(\Psi_2),\; R \subseteq S \;\big\}$$
where $\delta_{\gamma_1,T_1,\gamma_2,T_2,R}$ is the conjunction of the following constraints:

$$\sum_{t \in T_1} \mathbf{P}(s,t) \bowtie \sum_{t \in T_2} \mathbf{P}(s,t) \quad \text{for each state } s \in R,$$
$$\sum_{t \in T_1} \mathbf{P}(s,t) \not\bowtie \sum_{t \in T_2} \mathbf{P}(s,t) \quad \text{for each state } s \in S \setminus R.$$

Note that the expectation and expectation comparison operators are dealt with in an analogous fashion to the probability and probability comparison operator. This is feasible as the computation of the expectation functions requires the solution of a linear equation system similar to the one for probabilities.

In all cases, we can simplify the set $\mathtt{Sat}_{\mathfrak{M}}(\Psi)$ as follows. We first remove all pairs $(\gamma, T) \in$ where γ is not satisfiable. This can be checked using algorithms for the existential theory of the reals. Furthermore, we can simplify $\mathtt{Sat}_{\mathfrak{M}}(\Psi)$ by combining pairs with the same T-component. This is, instead of m pairs $(\gamma_1, T), \ldots, (\gamma_m, T) \in \mathtt{Sat}_{\mathfrak{M}}(\Psi)$, we can deal with a single pair $(\gamma_1 \vee \ldots \vee \gamma_m, T)$.

C.2 Complexity bounds of (All) and (MC-E)

▶ **Theorem 16** (Theorem 5 in the main part)**.** *The existential PCTL+EC model checking problem (MC-E) for augmented polynomial pMC is in PSPACE.*

Proof. As PSPACE = NPSPACE, it suffices to provide a nondeterministic polynomially space-bounded algorithm for (MC-E). Given an augmented pMC $\mathfrak{M} = (S, s_{init}, E, \mathbf{P}, \mathfrak{C})$ over parameters $x_1, \ldots, x_k$, and a PCTL+EC formula Φ, we process the DAG-representation of Φ in a bottom-up manner and assign to each sub-formula Ψ a pair (γ_Ψ, T_Ψ) consisting of a polynomial constraint γ_Ψ and a subset T_Ψ of the state space S. (Nodes of the DAG for Φ are identified with the corresponding sub-formula of Φ.) The computation of the pairs (γ_Ψ, T_Ψ) is similar as in the algorithm to compute $\mathtt{Sat}_{\mathfrak{M}}(\Phi)$. The essential difference is that we do not explore all alternative satisfaction sets for Φ and its subformulas. The treatment of the leaves is trivial: the pair (χ, S) is assigned to $\mathtt{true}$, while $(\chi, \{s \in S : a \in \mathscr{L}(s)\})$ is assigned to atomic proposition a. Here, χ is – as before – a conjunction of polynomial constraints that characterizes the set X of admissible parameter valuations. The treatment of the inner nodes is as follows:

- If $\Psi = \neg\Psi'$ then $\gamma_\Psi = \gamma_{\Psi'}$ and $T_\Psi = S \setminus T_{\Psi'}$.
- If $\Psi = \Psi_1 \wedge \Psi_2$ then $\gamma_\Psi = \gamma_{\Psi_1} \wedge \gamma_{\Psi_2}$ and $T_\Psi = T_{\Psi_1} \cap T_{\Psi_2}$
- Suppose now $\Psi = \mathbb{P}_{\bowtie c}(\bigcirc \Psi')$. Let $(\gamma, T) = (\gamma_{\Psi'}, T_{\Psi'})$. Then, we nondeterministically guess a subset R of S (in $\mathscr{O}(n)$ steps by guessing one bit per state where $n = |S|$) and put $\gamma_\Psi = \delta_{\gamma,T,R}$ and $T_\Psi = R$ where $\delta_{\gamma,T,R}$ is obtained as in the algorithm to compute $\mathtt{Sat}_{\mathfrak{M}}(\cdot)$.
- For $\Psi = \mathbb{P}_{\bowtie c}(\Psi_1 \cup \Psi_2)$ and $(\gamma_i, T_i) = (\gamma_{\Psi_i}, T_{\Psi_i})$, $i = 1, 2$, we nondeterministically guess a subset R of S and assign $\gamma_\Psi = \delta_{\gamma_1, T_1, \gamma_2, T_2, R}$ (defined as in the algorithm to compute $\mathtt{Sat}_{\mathfrak{M}}(\cdot)$), and $T_\Psi = R$.
- As above, we deal with the probability comparison, expectation, and expectation comparison operators in an analogous fashion.

Finally, we check whether (i) $s_{init} \in T_\Phi$, and (ii) there is a parameter valuation $\overline{\xi} = (\xi_1, \ldots, \xi_k)$ for $\overline{x} = (x_1, \ldots, x_k)$ such that $\overline{\xi} \models \gamma_\Phi$ using a polynomial space-bounded algorithm for the existential theory of the reals. If so, then the algorithm returns "yes". Otherwise, the algorithm returns "no".

Note that γ_Φ logically implies γ_Ψ for all sub-formulas Ψ of Φ, and that $T_\Psi = \mathrm{Sat}_{\mathfrak{M}}(\Psi)(\overline{\xi})$ for each sub-formula Ψ of Φ and each parameter valuation $\overline{\xi}$ satisfying γ_Φ. Vice versa, if $\overline{\xi}$ is a parameter valuation such that $\mathfrak{M}(\overline{\xi}) \models \Phi$ then $\overline{\xi} \models \gamma_\Phi$ provided that the guessed sets for the probability operator enjoy the property $T_\Psi = \mathrm{Sat}_{\mathfrak{M}}(\Psi)(\overline{\xi})$. Thus, the sketched algorithm has a run returning the answer "yes" if and only if there exists a parameter valuation $\overline{\xi}$ such that $\mathfrak{M}(\overline{\xi}) \models \Phi$. The memory requirements of the algorithm are dominated by the space requirements of the called algorithm for the existential theory of the reals. Hence, the algorithm is polynomially space-bounded. ◀

▶ **Theorem 17** (Theorem 6 in the main part). *Given an augmented polynomial pMC $\mathfrak{M}$ on parameters $\overline{x}$ with initial state s_{init} and an atomic proposition a, and a probability threshold $c \in \mathbb{Q} \cap]0,1[$, the problem to decide whether there exists $\overline{\xi} \in X$ such that $\Pr^{\mathfrak{M}(\overline{\xi})}_{s_{\mathit{init}}}(\Diamond a) > c$ is NP-hard, even for acyclic pMCs with the assigned transition probabilities being either constant, or linear in one parameter, i. e., $\mathbf{P}(s,t) \in \bigcup_{i=1}^{k} \mathbb{Q}[x_i]$, $\deg(\mathbf{P}(s,t)) \leq 1$, for all $(s,t) \in E$, and where the polynomial constraints for the parameters $x_1,\dots,x_k$ are of the form $f(x_i) \geqslant 0$ with $f \in \mathbb{Q}[x_i]$, $\deg(f) \leq 2$.*

Proof. We provide a polynomial reduction from 3SAT. Let

$$\alpha \;=\; \bigwedge_{i=1}^{m} \big(L_{i,1} \vee L_{i,2} \vee L_{i,3}\big)$$

be a 3CNF formula where the $L_{i,j}$'s are literals, say $L_{i,j} \in \{\kappa_h, \neg\kappa_h : h = 1,\dots,k\}$ for $i = 1,\dots,m$, $j = 1,2,3$. Consider the pMC $\mathfrak{M} = (S, s_{\mathit{init}}, E, \mathbf{P}, \mathfrak{C})$ over the parameters $x_1,\dots,x_k$:

$$\begin{aligned}
S &= \{s_i, s_{i,1}, s_{i,2}, s_{i,3} : i = 1,\dots,m\} \cup \{s_{m+1}, \mathit{fail}\}, \quad s_{\mathit{init}} = s_1,\\
E &= \big\{(s_i,s_{i,j}), (s_{i,j},s_{i+1}), (s_{i,j},\mathit{fail}) : i = 1,\dots,m, j = 1,2,3\big\}\\
&\quad \cup \big\{(\mathit{fail},\mathit{fail})\big\} \cup \big\{(s_{m+1},s_{m+1})\big\}\\
\mathbf{P}(s,t) &= \begin{cases}
\frac{1}{3} & \text{if } s = s_i,\, t = s_{i,j},\, 1 \leqslant i \leqslant m,\, j \in \{1,2,3\}\\
x_h & \text{if } s = s_{i,j},\, t = s_{i+1},\, 1 \leqslant i \leqslant m,\, j \in \{1,2,3\},\, L_{i,j} = \kappa_h\\
x_h & \text{if } s = s_{i,j},\, t = \mathit{fail},\, 1 \leqslant i \leqslant m,\, j \in \{1,2,3\},\, L_{i,j} = \neg\kappa_h\\
1 - x_h & \text{if } s = s_{i,j},\, t = s_{i+1},\, 1 \leqslant i \leqslant m,\, j \in \{1,2,3\},\, L_{i,j} = \neg\kappa_h\\
1 - x_h & \text{if } s = s_{i,j},\, t = \mathit{fail},\, 1 \leqslant i \leqslant m,\, j \in \{1,2,3\},\, L_{i,j} = \kappa_h\\
1 & \text{if } s = t = s_{m+1} \text{ or } s = t = \mathit{fail}\\
0 & \text{otherwise}
\end{cases},\\
\mathfrak{C} &= \left\{\, x_h^2 - x_h + \frac{1}{3^m} - \frac{1}{3^{2m}} \geqslant 0 \;:\; h = 1,\dots,k \,\right\}
\end{aligned}$$

The graph $G = (S,E)$ of $\mathfrak{M}$ is pictured in Figure 3.

Let us now take a closer look at the PCTL formula $\Phi = \mathbb{P}_{>c}(\Diamond s_{m+1})$ with $c = \frac{1}{3^m}$. We will prove that there exists a parameter valuation $\overline{\xi} = (\xi_1,\dots,\xi_k)$ for $x_1,\dots,x_k$ such that $\mathfrak{M}(\overline{\xi}) \models \Phi$ if and only if α is satisfiable. Firstly, assume that there exists a satisfying assignment $\mu : \{\kappa_1 \dots, \kappa_k\} \to \{0,1\}$ for α. We then choose the following valuation $\xi_1,\dots,\xi_k$ for the parameters $x_1,\dots,x_k$ where $\wp = 1 - \frac{1}{3^m}$:

$$\xi_h \;=\; \begin{cases} \wp & \text{if } \mu(\kappa_h) = 1\\ 1 - \wp & \text{if } \mu(\kappa_h) = 0 \end{cases}.$$

This valuation is admissible as it satisfies the constraints stated for the pMC. In the worst case, exactly one literal is satisfied in each clause of α. Thus, one obtains the following inequality for the probability of reaching s_{m+1} from s_1 in $\mathfrak{M}$:

$$\Pr^{\mathfrak{M}(\overline{\xi})}_{s_1}\big(\Diamond s_{m+1}\big) \;\geqslant\; \left(\frac{1}{3}\wp + \frac{2}{3}(1-\wp)\right)^m \;=\; \frac{1}{3^m} \cdot \underbrace{(2-\wp)^m}_{>1} \;>\; \frac{1}{3^m} \;=\; c$$

Hence, $\mathfrak{M}(\overline{\xi}) \models \Phi$.

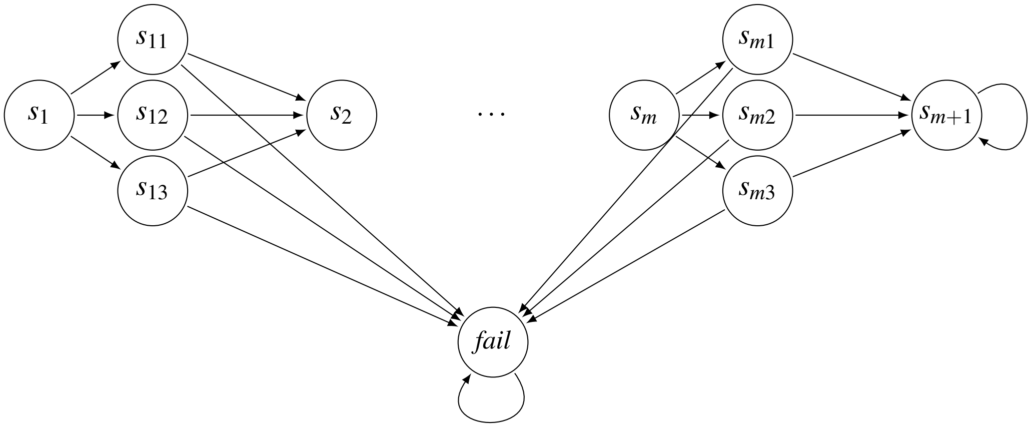


Figure 3 – Graph of the pMC constructed for reduction of 3SAT to PCTL model checking on parametric Markov chains

Suppose now that there exists an admissible valuation $\xi_1,\ldots,\xi_k$ of $x_1,\ldots,x_k$ with $\mathfrak{M}(\overline{\xi}) \models \Phi$. By definition of $\mathfrak{M}$, for each $h = 1,\ldots,k$ either $0 < \xi_h \leqslant 1-\wp$ or $\wp \leqslant \xi_h < 1$. (As before, $\wp = 1 - \frac{1}{3^m}$.) Suppose by contradiction that α is not satisfiable. Then, the assignment μ given by $\mu(\kappa_h) = 0$ if $0 < \xi_h \leqslant 1-\wp$ and $\mu(\kappa_h) = 1$ if $\wp \leqslant \xi_h < 1$ is not satisfying for α. Therefore, there exists $\iota \in \{1,\ldots,m\}$ such that the ι-th clause of α does not hold under μ. But then, $\mathbf{P}(s_{\iota,j}, s_{\iota+1})(\overline{\xi}) \leqslant 1-\wp$ for $j = 1,2,3$. Hence:

$$\begin{aligned} c \quad < \quad & \Pr_{s_1}^{\mathfrak{M}(\overline{\xi})}\big(\Diamond s_{m+1}\big) \quad = \quad \Pr_{s_1}^{\mathfrak{M}(\overline{\xi})}\big(\Diamond s_\iota\big)\cdot \Pr_{s_\iota}^{\mathfrak{M}(\overline{\xi})}\big(\Diamond s_{\iota+1}\big)\cdot \Pr_{s_{\iota+1}}^{\mathfrak{M}(\overline{\xi})}\big(\Diamond s_{m+1}\big) \\ < \quad & \Pr_{s_\iota}^{\mathfrak{M}(\overline{\xi})}\big(\Diamond s_{\iota+1}\big) \quad \leqslant \quad 3\cdot\tfrac{1}{3}\cdot(1-\wp) \quad = \quad 1-\wp \quad = \quad \tfrac{1}{3^m} \quad = \quad c \end{aligned}$$

This is a contradiction. Thus, α is satisfiable. ◀

C.3 Model checking PCTL+EC on univariate pMCs

▶ **Theorem 18** (Theorem 7 in main part). *Let Φ be a PCTL+EC formula without nested probability, expectation or comparison operators, and let $\mathfrak{M}$ be a polynomial pMC on the single parameter x. The problem to decide whether there exists an admissible parameter valuation $\xi \in X$ such that $\mathfrak{M}(\xi) \models \Phi$ is decidable in polynomial-time.*

Proof. We process the DAG-representation of Φ in a bottom-up manner to assign a Boolean combination of polynomial constraints γ_Ψ to each subformula Ψ of Φ (represented by a node in the DAG) such that for all $\xi \in \mathbb{R}$: $\xi \models \chi \wedge \gamma_\Psi$ if and only if $\xi \in X$ and $s_{init} \models_{\mathfrak{M}(\xi)} \Psi$. We finally check the existence of a value ξ for the parameter x such that $\xi \models \gamma_\Phi$. This can be done in polynomial time [4].

To compute the constraints γ_Ψ for the subformulas Ψ of Φ, we use an analogous approach as in the computation scheme for the satisfaction functions in multivariate pMCs. As before, let χ be a formula representing the admissible parameter valuations. The treatment of the leaves in the DAG-presentation of Φ is obvious, e. g., $\gamma_a = \texttt{true}$ if s_{init} is labelled with a, otherwise $\gamma_s = \texttt{false}$. The treatment of inner nodes representing subformulas of the form $\Psi_1 \wedge \Psi_2$ or $\neg\Psi$ is also clear: $\gamma_{\Psi_1\wedge\Psi_2} = \gamma_{\Psi_1} \wedge \gamma_{\Psi_2}$ and $\gamma_{\neg\Psi} = \neg\gamma_\Psi$. For formulas Ψ where the outermost operator is a probability, expectation or comparison operator, we rely on Lemma 2 to generate the polynomial constraint γ_Ψ. Note that the length of the resulting formula γ_Φ is polynomially bounded in the length of Φ and the size of $\mathfrak{M}$. ◀

▶ **Theorem 19** (Theorem 8 in the main part). *Let Φ be a PCTL+EC formula, and let $\mathfrak{M}$ be a polynomial pMC on the single parameter x. The PCTL+EC model checking problem to decide whether there exists an admissible parameter valuation $\xi \in X$ such that $\mathfrak{M}(\xi) \models \Phi$ is NP-complete. NP-hardness even holds for acyclic polynomial pMC and the fragment of PCTL+C that uses the comparison operator $\mathbb{C}_{\mathrm{Pr}}$, but not the probability operator $\mathbb{P}$, as well as for (cyclic) polynomial pMC in combination with PCTL.*

Proof. Membership to NP can be shown using a guess-and-check algorithm as in the proof of the PSPACE upper bound for the multivariate case (see Theorem 16). However, instead of polynomially space-bounded algorithms for the existential theory of the reals, we can apply here polynomially time-bounded algorithms for the univariate case [4].

We prove NP-hardness by a reduction from 3SAT. Given a 3CNF formula $\alpha = \bigwedge_{i=1}^{m} \big(L_{i,1} \vee L_{i,2} \vee L_{i,3}\big)$ with literals $L_{i,j} \in \{\kappa_1, \ldots, \kappa_\ell, \neg\kappa_1, \ldots, \neg\kappa_\ell\}$, we construct a PCTL formula Φ_α and a univariate pMC $\mathfrak{M}_\alpha$ with parameter x such that α is satisfiable if and only if there is an admissible valuation ξ of x such that $\mathfrak{M}_\alpha(\xi) \models \Phi_\alpha$.

In what follows, we write $\xi[i]$ to denote the i-th position of the fractional part of ξ's binary encoding, i. e., $\xi[i] \in \{0,1\}$ for all i and $\xi = \sum_{i=1}^{\infty} \xi[i]/2^i$. The idea is that the Boolean variable κ_i stands for the requirement $\xi[i] = 1$. The latter will be encoded by a PCTL formula Ψ_i. The PCTL formula Φ_α then has the following form:

$$\Phi_\alpha \quad = \quad \alpha[\kappa_1/\Psi_1, \ldots, \kappa_\ell/\Psi_\ell]$$

NP-hardness for PCTL+C. We first provide definitions for the Ψ_i as PCTL+C formulas. The pMC $\mathfrak{M}$ has the state space

$$S \quad = \quad \big\{\, v_j, t_j, u_j, ok_j, yes_j, no_j \;:\; j = 1, \ldots, \ell \,\big\}$$

The initial state is $s_{init} = v_\ell$. $\mathfrak{M}$ has the following edges for mode j where $j \in \{1, \ldots, \ell\}$:

- from state v_j: (v_j, t_j), (v_j, u_j)
- from state u_j: (u_j, ok_j), (u_j, no_j)
- from state t_j: (t_j, yes_j), (t_j, no_j) and (t_j, v_k) for all $k \in \{1, \ldots, j-1\}$
  Note that state t_1 has only two successors, namely yes_1 and no_1.
- States no_j, yes_j, ok_j are traps.

Note that $\mathfrak{M}$ is acyclic. The transition probabilities are defined as follows.

$$\begin{aligned}
&\mathbf{P}(v_j, t_j) = \mathbf{P}(v_j, u_j) = \tfrac{1}{2} \;\text{ for } 1 \leqslant j \leqslant l\\
&\mathbf{P}(t_j, v_k) = 1/2^k \;\text{ for } 1 \leqslant k < j \leqslant l\\
&\mathbf{P}(t_j, yes_j) = \mathbf{P}(t_j, no_j) = 1/2^j\\
&\mathbf{P}(u_j, no_j) = 1 - x \;\text{ and }\; \mathbf{P}(u_j, ok_j) = x
\end{aligned}$$

We use the names of the states as atomic propositions, i. e., AP $= S$ with the obvious labelling function. If $W = \{w_1, \ldots, w_h\}$ is a subset of S then we use W as a short form notation for the propositional formula $w_1 \vee \ldots \vee w_h$.

Obviously, the size of $\mathfrak{M}$ is linear in the number ℓ of Boolean variables in the given 3CNF formula α.

Let $\xi = \sum\limits_{j=1}^{\infty} \xi[j]/2^j \in]0,1[$. We define the index set I_ξ by:

$$I_\xi \quad = \quad \big\{\, j \in \{1, \ldots, \ell\} \;:\; \xi[j] = 1 \,\big\}$$

Then, $1 \in I_\xi$ iff $\xi \geqslant \frac{1}{2}$. For $j = 2, \dots, \ell$:

$$j \in I_\xi \qquad \text{iff} \qquad \xi \ \geqslant \ \frac{1}{2^j} + \sum_{k=1}^{j-1} \frac{I_\xi(k)}{2^k}$$

where $I_\xi(k) = 1$ iff $k \in I_\xi$ (in which case $\xi[k] = 1$) and $I_\xi(k) = 0$ iff $k \notin I_\xi$ (in which case $\xi[k] = 0$).

We now establish some properties of the MC $\mathfrak{M}(\xi)$. The only path from u_j satisfying $\Diamond ok_j$ consists of the edge (u_j, ok_j). Hence:

$$\Pr_{u_j}^{\mathfrak{M}(\xi)}(\Diamond ok_j) \ = \ \mathbf{P}(u_j, ok_j)(\xi) \ = \ \xi$$

Therefore:

$$\Pr_{v_j}^{\mathfrak{M}(\xi)}\big(\ \Diamond ok_j\ \big) \ = \ \frac{\xi}{2} \ = \ \sum_{k=1}^{\infty} \frac{\xi[k]}{2^{k+1}}$$

Let $Good_j = \{yes_j\} \cup \{v_k : k < j, k \in I_\xi\}$. The probability for reaching $Good_j$ from state v_j is the sum of the probabilities of the (cylinder sets of the) paths of $v_j t_j s$ with $s \in Good_j$:

$$\Pr_{v_j}^{\mathfrak{M}(\xi)}\big(\ \Diamond Good_j\ \big) \ = \ \frac{1}{2^{j+1}} + \sum_{\substack{k=1 \\ k \in I_\xi}}^{j-1} \frac{1}{2^{k+1}} \ = \ \frac{1}{2^{j+1}} + \sum_{k=1}^{j-1} \frac{\xi[k]}{2^{k+1}}$$

This value is less than or equal to $\xi/2$ if and only if $\xi[j] = 1$. This yields:

$$\Pr_{v_j}^{\mathfrak{M}(\xi)}\big(\ \Diamond Good_j\ \big) \ \leqslant \ \Pr_{v_j}^{\mathfrak{M}(\xi)}\big(\ \Diamond ok_j\ \big) \qquad \text{iff} \qquad j \in I_\xi \qquad \text{iff} \qquad \xi[j] = 1$$

This constraint is used to define PCTL+C formulas Ξ_j as follows. For $j = 1$ let

$$\Xi_1 \ = \ v_1 \wedge \mathbb{P}_{\geqslant 0.25}(\Diamond ok_1),$$

and for $j = 2, \dots, \ell$:

$$\Xi_j \ = \ v_j \wedge \mathbb{C}_{\Pr}\big(\ \Diamond(yes_j \vee \Xi_{<j}),\ \leqslant,\ \Diamond ok_j\ \big)$$

where $\Xi_{<j} = \Xi_1 \vee \Xi_2 \vee \ldots \vee \Xi_{j-1}$. This yields $Good_j = \mathrm{Sat}_{\mathfrak{M}(\xi)}\big(yes_j \vee \Xi_{<j}\big)$, and

$$\mathrm{Sat}_{\mathfrak{M}(\xi)}(\Xi_j) \ = \ \begin{cases} \{v_j\} & : \ \text{if } \xi[j] = 1 \\ \varnothing & : \ \text{if } \xi[j] = 0 \end{cases}$$

We then have:

$$v_j \models_{\mathfrak{M}(\xi)} \Xi_j \qquad \text{iff} \qquad \xi[j] = 1.$$

We define the PCTL+C formulas Ψ_j by $\mathbb{P}_{=1}(\Box(v_j \rightarrow \Xi_j))$.

The length of the formulas Ψ_j grows linear in j. Hence, the length of Φ_α is polynomial in the length of the 3CNF formula α. Recall that the length of a state formula is defined by the number of nodes in its DAG-representation. Thus, each of the formulas Ξ_j is represented by exactly one node in the DAG for Φ_α, although Ξ_1 has exponentially many occurrences in the string representation of Φ_α.

The 3CNF formula α is satisfiable if and only if there exists $\xi \in]0,1[$ such that the constructed PCTL+C formula Φ_α holds for the MC $\mathfrak{M}(\xi)$. To see this, suppose first that α has a satisfying assignment

$\mu : \{\kappa_1, \ldots, \kappa_\ell\} \to \{0,1\}$. Let $\xi = \sum_{j=1}^{\ell} \mu(\kappa_j)/2^j$. Then, $s_{init} \models_{\mathfrak{M}(\xi)} \Phi_\alpha$. Vice versa, if $\xi \in]0,1[$ such that $s_{init} \models_{\mathfrak{M}(\xi)} \Phi_\alpha$, let μ be the assignment given by $\mu(\kappa_j) = \xi[j]$ for $j \in \{1, \ldots, \ell\}$. Then $\mu \models \alpha$.

NP-hardness for PCTL (without comparison operator). To replace the PCTL+C formulas Ξ_j with a PCTL formula we switch from $\mathfrak{M}$ to the pMC $\mathfrak{N}$ that arises from $\mathfrak{M}$ by adding *reset edges* from $\mathfrak{M}$'s trap states no_j, yes_j, ok_j to the initial state $s_{init} = s_\ell$. The transition probabilities of the reset edges are 1, i. e., $\mathbf{P}(no_j, s_\ell) = \mathbf{P}(yes_j, s_\ell) = \mathbf{P}(ok_j, s_\ell) = 1$. Obviously, $\mathfrak{N}$ is strongly connected.

We now define PCTL formulas $\Upsilon_1, \ldots, \Upsilon_\ell$ by:

$$\Upsilon_1 \;=\; v_1 \,\wedge\, \mathbb{P}_{\geqslant 0.5}(\,(\neg yes_1) \;\mathsf{U}\; ok_1\,)$$

and for $j = 2, \ldots, \ell$:

$$\Upsilon_j \;=\; v_j \,\wedge\, \mathbb{P}_{\geqslant 0.5}\big(\,\neg(yes_j \vee \Upsilon_{<j})\,\mathsf{U}\, ok_j\,\big)$$

where $\Upsilon_{<j} = \Upsilon_1 \vee \Upsilon_2 \vee \ldots \vee \Upsilon_{j-1}$. We then define $\Psi_j \;=\; \mathbb{P}_{=1}\big(\Box(v_j \to \Upsilon_j)\big)$.

The length of the resulting formula Φ_α is polynomial in the length of α. The remaining task is to prove that α is satisfiable if and only if there is some $\xi \in]0,1[$ such that $\mathfrak{N}(\xi) \models \Phi_\alpha$.

To prove this, we first consider some fixed value $\xi \in]0,1[$. Let

$$\begin{aligned} Goal_j &= \big\{yes_j, ok_j\big\} \,\cup\, \big\{v_k : k \in I_\xi, k < j\big\} \\ Fail_j &= \big\{no_j\big\} \,\cup\, \big\{v_k : k \notin I_\xi\big\} \end{aligned}$$

Then, we have:

$$\Pr^{\mathfrak{N}(\xi)}_{v_j}\big(\,(\neg Goal_j) \;\mathsf{U}\; ok_j\,\big) \;=\; \Pr^{\mathfrak{M}(\xi)}_{v_j}\big(\,\Diamond ok_j \mid \neg\Diamond Fail_j\,\big) \;=\; \frac{\Pr^{\mathfrak{M}(\xi)}_{v_j}\big(\,\Diamond ok_j\,\big)}{\Pr^{\mathfrak{M}(\xi)}_{v_j}\big(\,\neg\Diamond Fail_j\,\big)}$$

Let $Good_j$ be as above. That is, $Good_j = Goal_j \setminus \{ok_j\}$. Then:

$$\Pr^{\mathfrak{N}(\xi)}_{v_j}\big(\,(\neg Goal_j) \;\mathsf{U}\; Good_j\,\big) \;=\; \Pr^{\mathfrak{M}(\xi)}_{v_j}\big(\,\Diamond Good_j \mid \neg\Diamond Fail_j\,\big) \;=\; \frac{\Pr^{\mathfrak{M}(\xi)}_{v_j}\big(\,\Diamond ok_j\,\big)}{\Pr^{\mathfrak{M}(\xi)}_{v_j}\big(\,\neg\Diamond Fail_j\,\big)}$$

Hence:

$$\begin{aligned} & \Pr^{\mathfrak{N}(\xi)}_{v_j}\big(\,(\neg Goal_j)\,\mathsf{U}\, ok_j\,\big) \;\geqslant\; \Pr^{\mathfrak{N}(\xi)}_{v_j}\big(\,(\neg Goal_j)\,\mathsf{U}\, Good_j\,\big) \\ \text{iff}\quad & \Pr^{\mathfrak{M}(\xi)}_{v_j}\big(\,\Diamond ok_j\,\big) \;\geqslant\; \Pr^{\mathfrak{M}(\xi)}_{v_j}\big(\,\Diamond Good_j\,\big) \\ \text{iff}\quad & \xi[j] = 1 \end{aligned}$$

Moreover, we have:

$$\Pr^{\mathfrak{N}(\xi)}_{v_j}\big(\,(\neg Goal_j)\,\mathsf{U}\, ok_j\,\big) \;+\; \Pr^{\mathfrak{N}(\xi)}_{v_j}\big(\,(\neg Goal_j)\,\mathsf{U}\, Good_j\,\big) \;=\; 1$$

The latter is based on the general observation that in each finite strongly connected MC $\mathcal{M}$ we have: $\Pr^{\mathcal{M}}_s(\,(\neg B) \;\mathsf{U}\; b) + \Pr^{\mathcal{M}}_s(\,(\neg B) \;\mathsf{U}\; (B \setminus \{b\})\,) = 1$ where s is a state in $\mathcal{M}$, B is a set of states in $\mathcal{M}$, and $b \in B$.

Putting things together we get:

$$\Pr^{\mathfrak{N}(\xi)}_{v_j}\big(\,(\neg Goal_j)\,\mathsf{U}\, ok_j\,\big) \;\geqslant\; \frac{1}{2} \qquad \text{iff} \qquad \xi[j] = 1$$

But then $\mathrm{Sat}_{\mathfrak{N}(\xi)}(\Upsilon_j) = \{v_j\}$ iff $\xi[j] = 1$, and $\mathrm{Sat}_{\mathfrak{N}(\xi)}(\Upsilon_j) = \varnothing$ iff $\xi[j] = 0$. The remaining arguments are the same as for the reduction to the model checking problem for PCTL+C. ◀

C.4 Model checking monotonic PCTL on pMCs

In this section, we suppose that $\mathfrak{M} = (S, s_{init}, E, \mathbf{P})$ is a univariate (plain) polynomial pMC on a single parameter x. Recall that E_+ denotes the set of edges $(s,t) \in E$ such that the polynomial $\mathbf{P}(s,t)$ is monotonically increasing in X, i. e., whenever $\xi_1, \xi_2 \in X$ and $\xi_1 < \xi_2$ then $\mathbf{P}(s,t)(\xi_1) \leqslant \mathbf{P}(s,t)(\xi_2)$. The set S_+ is defined as the set of states s such that for each finite path $\pi = s_0 s_1 \ldots s_m$ with $s_m = s$ we have $(s_i, s_{i+1}) \in E_+$ for $i = 0, 1, \ldots, m-1$. The states in S_+ will be called *monotonic* states.

▶ **Lemma 20.** *Let $A, B \subseteq S_+$ and φ be one of the path formulas $A \,\mathsf{U}\, B$, $A \,\mathsf{R}\, B$, $\Diamond B$ or $\Box B$. Then, for all ξ_1, $\xi_2 \in X$ with $\xi_1 < \xi_2$ and all states s:*

$$\Pr_s^{\mathfrak{M}(\xi_1)}(\varphi) \;\leqslant\; \Pr_s^{\mathfrak{M}(\xi_2)}(\varphi)$$

Moreover, $\mathrm{Sat}_{\mathfrak{M}(\xi_1)}\big(\, \mathbb{P}_{\geqslant c}(\varphi) \,\big) \;\subseteq\; \mathrm{Sat}_{\mathfrak{M}(\xi_2)}\big(\, \mathbb{P}_{\geqslant c}(\varphi) \,\big) \;\subseteq\; S_+$ *for each $c \in]0,1]$. The analogous statement holds for strict probability thresholds "$> c$"*

Recall that the fragment of monotonic PCTL formulas is generated from the grammar:

$$\begin{array}{rcl} \Phi & ::= & a \in S_+ \mid \Phi \wedge \Phi \mid \Phi \vee \Phi \mid \mathbb{P}_{\geqslant c}(\varphi) \mid \mid \mathbb{P}_{>c}(\varphi) \\ \varphi & ::= & \bigcirc \Phi \mid \Phi \,\mathsf{U}\, \Phi \mid \Phi \,\mathsf{R}\, \Phi \mid \Diamond \Phi \mid \Box \Phi \end{array}$$

where $c \in \mathbb{Q}_{>0}$. The following lemma asserts the monotonicity of the satisfaction function $\mathrm{Sat}_{\mathfrak{M}}(\Phi) : X \to 2^S$ for monotonic formulas Φ. (Recall that $\mathrm{Sat}_{\mathfrak{M}}(\Phi)(\xi) = \mathrm{Sat}_{\mathfrak{M}(\xi)}(\Phi)$.)

▶ **Lemma 21.** *If $\mathfrak{M}$ and Φ are monotonic then for any two valuations ξ_1 and ξ_2 of x with $\xi_1 < \xi_2$:*

$$\mathrm{Sat}_{\mathfrak{M}(\xi_1)}(\Phi) \;\subseteq\; \mathrm{Sat}_{\mathfrak{M}(\xi_2)}(\Phi) \;\subseteq\; S_+ \qquad (*)$$

Proof. We will prove this by structural induction. Consider two arbitrary but fixed valuations ξ_1 and ξ_2 of x satisfying $\xi_1 < \xi_2$. The claim is obvious for the case where Φ is an atomic formula $a \in S_+$. In the induction step we consider a PCTL-formula Ψ of the form $\Phi = \Psi_1 \wedge \Psi_2$, $\Phi = \Psi_1 \vee \Psi_2$, $\Phi = \mathbb{P}_{\geqslant c}(\varphi)$ or $\Phi = \mathbb{P}_{>c}(\varphi)$ where $c > 0$ and $\varphi \in \{\bigcirc\Psi, \Psi_1 \,\mathsf{U}\, \Psi_2, \Psi_1 \,\mathsf{R}\, \Psi_2, \Diamond\Psi, \Box\Psi\}$. The cases $\Phi = \Psi_1 \wedge \Psi_2$ and $\Phi = \Psi_1 \vee \Psi_2$ are obvious consequences of the induction hypothesis. Let us now consider the case where Φ has the form $\mathbb{P}_{\geqslant c}(\varphi)$ where φ is one of the four formulas above. We now can rely on the following two facts:

(1) If $A \subseteq A'$, $B \subseteq B'$ and $\xi \in X$ then $\Pr_s^{\mathfrak{M}(\xi)}(A \,\mathsf{U}\, B) \leqslant \Pr_s^{\mathfrak{M}(\xi)}(A' \,\mathsf{U}\, B')$, and therefore,

$$\mathrm{Sat}_{\mathfrak{M}(\xi)}\big(\, \mathbb{P}_{\geqslant c}(A \,\mathsf{U}\, B) \,\big) \;\subseteq\; \mathrm{Sat}_{\mathfrak{M}(\xi)}\big(\, \mathbb{P}_{\geqslant c}(A' \,\mathsf{U}\, B') \,\big)$$

The analogous statement holds for strict probability thresholds "$> c$" and the release operator R, the next operator $\bigcirc$ and the derived operators $\Diamond$ and $\Box$.

(2) If $A, B \subseteq S_+$ and s is a state such that $s \models \exists\varphi$ where φ is one of the formulas $\bigcirc B$, $A \,\mathsf{U}\, B$, $A \,\mathsf{R}\, B$, $\Diamond B$ or $\Box B$ then s is a predecessor of a state in $A \cup B \subseteq S_+$. Hence, $s \in S_+$ (by definition of S_+).

These observations together with Lemma 20 will now be used to establish the monotonicity property of the satisfaction functions of monotonic formulas.

Consider the case $\Psi = \mathbb{P}_{\geqslant c}(\varphi)$ where $\varphi = \Psi_1 \,\mathsf{U}\, \Psi_2$. Let $\xi_1, \xi_2 \in X$, $\xi_1 < \xi_2$ and let $A = \mathrm{Sat}_{\mathfrak{M}(\xi_1)}(\Psi_1)$, $A' = \mathrm{Sat}_{\mathfrak{M}(\xi_2)}(\Psi_1)$ and $B = \mathrm{Sat}_{\mathfrak{M}(\xi_1)}(\Psi_2)$, $B' = \mathrm{Sat}_{\mathfrak{M}(\xi_2)}(\Psi_2)$, Then $A \subseteq A' \subseteq S_+$ and $B \subseteq B' \subseteq S_+$ by induction hypothesis. Using (1) we get:

$$\mathrm{Sat}_{\mathfrak{M}(\xi_1)}(\Phi) \;=\; \mathrm{Sat}_{\mathfrak{M}(\xi_1)}(\mathbb{P}_{\geqslant c}(A \,\mathsf{U}\, B)) \subseteq \mathrm{Sat}_{\mathfrak{M}(\xi_1)}(\mathbb{P}_{\geqslant c}(A' \,\mathsf{U}\, B')) \subseteq \mathrm{Sat}_{\mathfrak{M}(\xi_2)}(\mathbb{P}_{\geqslant c}(A' \,\mathsf{U}\, B')) \;=\; \mathrm{Sat}_{\mathfrak{M}(\xi_2)}(\Phi).$$

The other cases, i. e., $\varphi \in \{\bigcirc\Psi, \Diamond\Psi, \Box\Psi, \Psi_1 \,\mathsf{R}\, \Psi_2\}$ or monotonic PCTL with strict lower probability bounds, are analogous and omitted here. Moreover, $s \models_{\mathfrak{M}(\xi)} \Psi$ for some $\xi \in X$ implies $s \models \exists\varphi$. Thus, observation (2) yields $\mathrm{Sat}_{\mathfrak{M}(\xi)}(\Psi) \subseteq S_+$. So in particular, $\mathrm{Sat}_{\mathfrak{M}(\xi_1)}(\Psi) \subseteq \mathrm{Sat}_{\mathfrak{M}(\xi_2)}(\Psi) \subseteq S_+$. ◀

By the monotonicity of the satisfaction function $\mathrm{Sat}_{\mathfrak{M}}(\Phi) : X \to 2^S$ for monotonic formulas we obtain:

▶ **Corollary 22.** *Let Ψ be a monotonic PCTL formula. Then there exist $\xi_\Psi \in \mathbb{R}$ and $S_\Psi \subseteq S$ such that $\mathrm{Sat}_{\mathfrak{M}(\xi)}(\Psi) = S_\Psi$ for all $\xi \in X$ with $\xi \geqslant \xi_\Psi$, and $\mathrm{Sat}_{\mathfrak{M}(\xi)}(\Psi) \subseteq S_\Psi$ for all $\xi \in X$ with $\xi < \xi_\Psi$.*

In what follows, we refer to S_Ψ in Corollary 22 as the *maximal satisfaction set* of Ψ. We are now ready to prove the main result:

▶ **Theorem 23** (Theorem 10 in the main part)**.** *Let $\mathfrak{M} = (S, s_{init}, E, \mathbf{P}, \mathfrak{C})$ be a univariate polynomial pMC on x, and Φ a monotonic PCTL formula. Then the model checking problem to decide whether there exists an admissible parameter valuation ξ for x such that $\mathfrak{M}(\xi) \models \Phi$ is in P.*

Proof. The idea of a polynomial time algorithm is to compute the maximal satisfaction sets S_Ψ of the subformulas Ψ of Φ. Then, there exists $\xi \in X$ with $\mathfrak{M}(\xi) \models \Phi$ if and only if $s_{init} \in S_\Phi$.

The sets S_Ψ can be computed in an inductive way by processing the nodes in the DAG representation of Φ in a bottom-up manner. The treatment of atomic propositions $a \in S_+$ is obvious, and so are the cases $\Psi = \Psi_1 \vee \Psi_2$, and $\Psi = \Psi_1 \wedge \Psi_2$. Note that $S_{\Psi_1 \vee \Psi_2} = S_{\Psi_1} \cup S_{\Psi_2}$ and $S_{\Psi_1 \wedge \Psi_2} = S_{\Psi_1} \cap S_{\Psi_2}$. Consider now the case $\Psi = \mathbb{P}_{\geqslant c}(\Psi_1 \,\mathsf{U}\, \Psi_2)$. Let $A = S_{\Psi_1}$ and $B = S_{\Psi_2}$. Using fraction-free Gaussian elimination, we compute the rational functions $p_s = f_s/g_s$ representing the probabilities $\mathrm{Pr}_s^{\mathfrak{M}}(A \,\mathsf{U}\, B)$. Using a polynomial time algorithm for the univariate theory of the reals [4], we check for each state $s \in S$ whether there is some $\xi \in X$ with $p_s(\xi) \geqslant c$. Then, $S_\Psi = \{s \in S : \exists \xi \in X \text{ s.t. } p_s(\xi) \geqslant c\}$. Again, the remaining cases are similar and omitted here. ◀