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

Christel Baier[a,1], Christian Hensel[c,2], Lisa Hutschenreiter[b,1], Sebastian Junges[c,2], Joost-Pieter Katoen[c,2], Joachim Klein[a,1]

[a] *Technische Universität Dresden, Dresden, Germany*
[b] *Heidelberg University, Heidelberg, Germany*
[c] *RWTH Aachen University, Aachen, Germany*

Abstract

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 over a finite set of parameters. Important tasks in the analysis of parametric Markov chains are (1) computing closed-form solutions for reachability 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.

Keywords: parametric Markov chain, parametric model checking, Gaussian elimination, PCTL, complexity

1. Introduction

Finite-state Markovian models are widely used as an operational model for the quantitative analysis of systems with probabilistic behavior. 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, the transition probabilities of Markovian models often 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.

Email addresses: `Christel.Baier@tu-dresden.de` (Christel Baier), `dehnert@cs.rwth-aachen.de` (Christian Hensel), `lisa.kruse@iwr.uni-heidelberg.de` (Lisa Hutschenreiter), `sebastian.junges@cs.rwth-aachen.de` (Sebastian Junges), `katoen@cs.rwth-aachen.de` (Joost-Pieter Katoen), `Joachim.Klein@tu-dresden.de` (Joachim Klein)

[1]The authors are supported by the DFG through the Collaborative Research Center SFB 912 – HAEC, the CRC/TR 248 – Foundations of Perspicuous Software Systems, the Excellence Initiative by the German Federal and State Governments (clusters of excellence cfaed and CeTI), the Research Training Group QuantLA (GRK 1763) and the DFG-projects BA-1679/11-1 and BA-1679/12-1.

[2]The authors are supported by the Research Training Group UnRAVeL (GRK 2236) and the CDZ project CAP (GZ 1023).

The possible change of transition probabilities motivated the investigation of *interval Markov chains* (IMCs) [1]. The transitions of IMCs are equipped with intervals of transition probabilities rather than concrete probability values. The model of *parametric Markov chains* (pMCs) has been been introduced independently by Daws [2] and Lanotte et al. [3]. pMCs are more general than IMCs as their transition probabilities are given by polynomials with rational coefficients over a fixed set of real-valued parameters $x_1, \ldots, x_k$, and allow for expressing dependencies between transition probabilities. These concepts can be further generalized to accommodate rational functions, that is, quotients of polynomials, as transition probabilities (see, e. g., [4]).

It is well-known that the probabilities p_s for reachability conditions $\Diamond \mathit{Goal}$ in pMCs with a finite state space S and a fixed underlying graph structure 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. By construction, the denominator is never 0. 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 [2] and Lanotte et al. [3]. Daws [2] 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. [4], the naïve implementation of Gaussian elimination for pMCs, which 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 [5] (as well as in the re-implementation within PRISM [6]), the authors of [4] 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 (k=1), 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., $k \geqslant 2$) [7]. To mitigate the cost of the gcd-computations, the model checker Storm [8] successfully uses techniques proposed in [9] 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 [10], this variant of Gaussian elimination goes at least 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. So for any fixed number of parameters k, one-step fraction-free Gaussian elimination yields an alternative polynomial-time algorithm for computing the rational functions for reachability probabilities. Analogous statements hold for expectations of random variables that are computable via linear equation systems, such as the expected accumulated weights until reaching a goal, and the expected mean payoff.

Contribution and paper structure. The purpose of this paper is to study the complexity of the model-checking problem for pMCs and probabilistic computation tree logic (PCTL) [11], 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 computing reachability probabilities and expected accumulated rewards. The main advantage of the one-step fraction-free Gaussian elimination is that it both avoids a blow-up of the intermediate equations, and the use of the costly gcd-

Table 1: Complexity results in a nutshell with references to the crucial theorems.

	Univariate / Fixed	Multivariate
PCTL (no nesting)	in P [Thm.10]	NP-hard (conjunction [Thm.9] or augmented [Thm.8]) in PSPACE [Thm.7]
PCTL	NP-complete (cyclic) [Thm.11]	NP-hard [Thm.8,9] in PSPACE [Thm.7]
PCTL+EC	NP-complete (acyclic) [Thm.11]	NP-hard [Thm.8,9] in PSPACE [Thm.7]

computations on multivariate polynomials. We implemented the fraction-free Gaussian elimination approach as an alternative solver for parametrized linear equation systems within Storm [8], the state-of-the-art probabilistic model checker for pMCs, and empirically evaluate its performance in comparison with the standard approaches. The second part of the paper (Section 4) presents complexity-theoretic results for the PCTL model-checking problem for 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 [12, 13] 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 model checking is solvable in polynomial time: (1) Boolean combinations of threshold constraints for reachability probabilities, expected accumulated weights until reaching a goal, and expected mean payoffs, and (2) 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 non-parametric Markov chains where the weights of the states are given as univariate polynomials, when restricting to Boolean combinations of the expectation operators.

We summarize the main complexity results in Table 1. A preliminary conference version of this article has appeared as [14]. This article extends the conference version with proofs omitted due to lack of space, a stronger NP-hardness result for existential model checking, a significantly improved implementation and an extended experimental evaluation.

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. [15, 16, 17, 18]. 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 [2, 5, 4, 9, 19] on the computation of the rational functions for reachability probabilities in pMCs, [3] identifies instances where the parameter synthesis problem for pMCs with one or two 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 four and rather strong syntactic characterizations of pMCs. In Section 3, we 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 [3] where the polynomial-time computability of the rational functions for reachability probabilities has been stated even for the multivariate case. [20] 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 [12] 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 [13] for ω-PCTL (extending PCTL by Boolean combinations of Büchi and co-Büchi conditions), and in [21] for linear temporal logic (LTL). Our results of the second part can be seen as an extension of the work [12, 13] 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 [12]. In [22], NP-completeness of existential model checking for pMCs with changing graph structure is shown. Additionally, that paper provides a proof for square-root-sum hardness.

There exist several approaches to check whether all valuations (in some defined region) of a pMC satisfy a PCTL formula. PARAM [4, 5] employs a heuristic, sampling based approach, while PROPhESY [19] relies on SMT solving via the existential theory of the reals. For the same problem, [23] 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. The existence of some value that satisfies the property is also addressed in [24], which reduces the problem to a series of geometric programs.

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 [25, 26].

Discrete-time Markov chains. A *(discrete-time) Markov chain* (MC) $\mathcal{M}$ is a tuple $(S, s_{\mathit{init}}, E, P)$ where S is a non-empty, finite set of *states* containing 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 $G_{\mathcal{M}} = (S, E)$ as the *graph* of $\mathcal{M}$. A state $s \in S$ in which $\mathit{Post}(s) = \varnothing$ is called a *trap (state)* of $\mathcal{M}$.

An *infinite path* in $\mathcal{M}$ 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 $\mathcal{M}$ is a finite sequence $s_0 s_1 \ldots s_m \in S^*$ of states in $\mathcal{M}$ 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. Let $\mathrm{Paths}(s)$ denote the set of all maximal paths in $\mathcal{M}$ starting in s. Relying on standard techniques, every MC induces a unique probability measure $\Pr_s^{\mathcal{M}}$ on the set of all paths.

Furthermore, we can extend an MC with a *weight function* $\mathit{wgt} \colon S \to \mathbb{Q}$. 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.

Steady-state probabilities and mean payoff. Given a strongly connected MC $\mathcal{M} = (S, s_{\mathit{init}}, E, P)$, 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 solution 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 the matrix A arises from $I - P$ by replacing the column of one state t with the column vector $(1, 1, \ldots, 1)$, and where b is the row vector $(0, 0, \ldots, 0, 1)$.

Given an MC $\mathcal{M} = (S, s_{\mathit{init}}, E, P)$ without traps that is augmented with a weight function $\mathit{wgt} \colon S \to \mathbb{Q}$, and $T \subseteq S$, 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 $\pi = s_0 s_1 s_2 \ldots \in \mathrm{Paths}(s_0)$ then

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

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 in a *bottom strongly connected component* (BSCC) of $\mathcal{M}$, i. e., a subgraph of $G_{\mathcal{M}}$ from which no states in S outside this subgraph can be reached, within finitely many steps, it suffices to consider their behavior within this BSCC.

It is known that for almost all paths π eventually entering a BSCC B, the mean payoff is

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

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

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

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

where $\mathrm{Pr}^{\mathcal{M}}_s(\Diamond B)$ stands for $\mathrm{Pr}^{\mathcal{M}}_s\big\{\, \pi = s_0 s_1 s_2 \ldots \in \mathrm{Paths}(s) : s_i \in B \text{ for some } i \,\big\}$ and $\mathcal{B}$ denotes the set of BSCCs in $\mathcal{M}$. Thus, the expected mean payoffs $\mathrm{E}^{\mathcal{M}}_s(\mathrm{mp}(T))$ are computable by solving the linear equation systems for the steady-state probabilities for each BSCC of $\mathcal{M}$ (see above) and the linear equation systems for the reachability probabilities for the BSCCs.

Accumulated weight. Given an MC $\mathcal{M}$ with weights, and $T \subseteq S$, the *accumulated weight* along a path in $\mathcal{M}$ until reaching a state in T is the sum of weights assigned to states before the first state in T. For a finite path $s_0 s_1 \ldots s_m$, let

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

We now define $\oplus T$ 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., π is infinite, or finite and ending in a trap. If π visits T, i.e., there is some i with $s_i \in T$, then

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

where n is the smallest index with $s_n \in T$. Note that $(\oplus T)(\pi) = 0$ if $s_0 \in T$. If π does not visit T, then $(\oplus T)^{\mathcal{M}}(\pi) = 0$. The *expected accumulated weight* for T in s, denoted by $\mathrm{E}^{\mathcal{M}}_s(\oplus T)$, is defined as the expectation of the random variable $\oplus T$. In this article, we consider the case that almost all paths reach T, i.e., that $\mathrm{Pr}^{\mathcal{M}}_s(\Diamond T) = 1$ for all states $s \in S$. In this case, the value assigned by $(\oplus T)^{\mathcal{M}}(\pi)$ to paths not visiting T is irrelevant, as the probability mass of the set of those paths is zero.

If $\mathrm{Pr}^{\mathcal{M}}_s(\Diamond T) = 1$ for all states $s \in S$, then the expected accumulated weight can be computed by solving the linear equation system resulting from the equations

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

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

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 *polynomial* $f \in \mathbb{Q}[\overline{x}]$ 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 defined as $\deg(f) = \max\big\{\, i_1 + \ldots + i_k : (i_1, \ldots, i_k) \in I, \alpha_{i_1,\ldots,i_k} \neq 0 \,\big\}$ 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 *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 chains. A *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, if and only if $(s,t) \notin E$. Intuitively, a pMC defines the family of Markov chains arising by plugging in concrete values for the parameters. As for Markov chains, we can extend pMCs with weight functions. In addition to assigning rational values, i.e., a weight function of the form $wgt\colon S \to \mathbb{Q}$, we also consider parametric weight functions $wgt\colon S \to \mathbb{Q}(\overline{x})$.

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 $Post(s)$ nonempty, and $P_{\overline{\xi}}(s,t) > 0$ if and only if $(s,t) \in E$, 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 $\mathcal{M}_{\overline{\xi}} = \mathfrak{M}(\overline{\xi}) = (S, s_{\mathit{init}}, E, P_{\overline{\xi}})$. The semantics of the pMC $\mathfrak{M}$ is then defined as the family of Markov chains induced by admissible parameter valuations, that is, $[\![\mathfrak{M}]\!] = \{\, \mathfrak{M}(\overline{\xi}) : \overline{\xi} \in X \,\}$. The admissibility constraints ensure that all of the Markov chains $[\![\mathfrak{M}]\!]$ share the same underlying graph-structure $G_{\mathcal{M}}$ and that qualitative reachability probabilities (e. g. "can be reached with positive probability", "can be reached with probability 1") do not depend on the concrete parameter valuations. This property can be used in a graph-based preprocessing step to identify states where a reachability probability is always 0 and remove those during the construction of the linear equation system, ensuring the uniqueness of the solution.

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 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 chains. An *interval Markov chain* (IMC) [12] 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 [12], this interpretation of the intervals 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. Each of these types of Markov chain, whether MC, (augmented) pMC, or IMC, can be equipped with a *labelling function* $\mathcal{L}\colon S \to 2^{\mathrm{AP}}$, 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.

Probabilistic computation tree logic. We augment the standard notion of probabilistic computation tree logic (PCTL) [11] with operators $\mathbb{E}_{\bowtie r}(\rho)$ for the expected accumulated weight and mean payoff, and $\mathbb{C}_{\mathrm{Pr}}(\varphi, \bowtie, \varphi), \mathbb{C}_{\mathrm{E}}(\rho, \bowtie, \rho)$ for comparison. Let AP be a finite set of atomic propositions with $a \in \mathrm{AP}$, and let $\bowtie$ denote $\leqslant, \geqslant, <, >$, or $=$, $c \in [0,1]$, and $r \in \mathbb{Q}$. Then

$$\begin{aligned} \Phi &::= \texttt{true} \mid a \mid \Phi \wedge \Phi \mid \neg\Phi \mid \mathbb{P}_{\bowtie c}(\varphi) \mid \mathbb{E}_{\bowtie r}(\rho) \mid \mathbb{C}_{\mathrm{Pr}}(\varphi, \bowtie, \varphi) \mid \mathbb{C}_{\mathrm{E}}(\rho, \bowtie, \rho) \quad \textit{state formula} \\ \varphi &::= \bigcirc\Phi \mid \Phi \,\mathsf{U}\, \Phi \quad \textit{path formula} \qquad \rho ::= \diamondsuit\!\!\!\!\;\cdot\;\Phi_p \mid \mathrm{mp}(\Phi) \quad \textit{terms for random variables} \end{aligned}$$

Here, the Φ_p in $\diamondsuit\!\!\!\!\;\cdot\;\Phi_p$ denotes the propositional fragment using only atomic propositions (i.e., $\Phi_p ::= \texttt{true} \mid a \mid \Phi_p \wedge \Phi_p \mid \neg\Phi_p$). Below, we will impose further restrictions on the Φ_p in $\diamondsuit\!\!\!\!\;\cdot\;\Phi_p$ to ensure the existence of the expected accumulated weight.

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

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.

For an MC $\mathcal{M}$ with states labelled by $\mathcal{L}\colon S \to \mathrm{AP}$ we use the standard semantics [26], with $\mathrm{Sat}_{\mathcal{M}}(\Phi)$ denoting the set of states that satisfy state formula Φ. We only recap the semantics of the probability, expectation, and comparison operators here. For each state $s \in S$, $s \models_{\mathcal{M}} \mathbb{P}_{\bowtie c}(\varphi)$ iff $\mathrm{Pr}_s^{\mathcal{M}}(\varphi) \bowtie c$, and $s \models_{\mathcal{M}} \mathbb{C}_{\mathrm{Pr}}(\varphi_1, \bowtie, \varphi_2)$ iff $\mathrm{Pr}_s^{\mathcal{M}}(\varphi_1) \bowtie \mathrm{Pr}_s^{\mathcal{M}}(\varphi_2)$. Here $\mathrm{Pr}_s^{\mathcal{M}}(\varphi)$ is short for $\mathrm{Pr}_s^{\mathcal{M}}\{\, \pi \in \mathrm{Paths}(s) : \pi \models_{\mathcal{M}} \varphi \,\}$.

For the expectation operators, the semantics $\rho^{\mathcal{M}}$ of the terms $\rho = \oint \Phi_p$ or $\rho = \mathrm{mp}(\Phi)$, given an MC $\mathcal{M}$, are random variables. For the mean-payoff operator, we suppose that $\mathcal{M}$ has no traps. This assumption ensures that all maximal paths in $\mathcal{M}$ are infinite and the well-definedness of the mean payoff function $\mathrm{mp}(T)$ for each $T \subseteq S$. The random variable $\mathrm{mp}(\Phi)^{\mathcal{M}}$ assigned to the term $\mathrm{mp}(\Phi)$ in $\mathcal{M}$ is given by $\mathrm{mp}(\Phi)^{\mathcal{M}} = \mathrm{mp}(T)$ for $T = \mathrm{Sat}_{\mathcal{M}}(\Phi)$. The semantics of the term $\oint \Phi_p$ is the random variable $\oint T$ where $T = \mathrm{Sat}_{\mathcal{M}}(\Phi_p)$. We assume here that, for every state $s \in S$, almost all paths reach T, i.e., that $\mathrm{Pr}_s^{\mathcal{M}}(\Diamond T) = 1$. Whether this assumption holds for an MC $\mathcal{M}$ solely depends on the underlying graph-structure $G_{\mathcal{M}}$ of the MC and not on the concrete probabilities. 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 $s \models_{\mathcal{M}} \mathbb{E}_{\bowtie r}(\rho)$ iff $\mathrm{E}_s^{\mathcal{M}}(\rho^{\mathcal{M}}) \bowtie r$ and $s \models_{\mathcal{M}} \mathbb{C}_{\mathrm{E}}(\rho_1, \bowtie, \rho_2)$ iff $\mathrm{E}_s^{\mathcal{M}}(\rho_1^{\mathcal{M}}) \bowtie \mathrm{E}_s^{\mathcal{M}}(\rho_2^{\mathcal{M}})$, where $\mathrm{E}_s^{\mathcal{M}}(\rho^{\mathcal{M}})$ denotes the expectation of random variable $\rho^{\mathcal{M}}$ (if existent).

We write $\mathcal{M} \models \Phi$ iff $s_{\mathit{init}} \models_{\mathcal{M}} \Phi$. 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 $(\Diamond T)$ denotes the set of maximal paths π that eventually visit a T-state.

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. For instance, the inner nodes of the DAG 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. So, 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 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 $\mathrm{Pr}_s^{\mathfrak{M}(\overline{x})}(\Diamond a)$ for reachability conditions are rational functions and computable via Gaussian elimination [2, 3]. Algorithms based on this observation are realised in, e. g., the tools PARAM [5] and Storm [8, 19] together with techniques 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 [10, 7], 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 focus on *polynomial* (augmented) pMCs throughout the remainder of the paper. Generally, a linear equation system containing rational functions as coefficients can be rearranged to one containing only polynomials by multiplying each equation 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. We avoid this blowup by adding variables in the following way. Let $\mathfrak{M} = (S, s_{\mathit{init}}, E, \mathbf{P}, \mathfrak{C})$ 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 $f_{s,t}, g_{s,t} \in \mathbb{Q}[\overline{x}]$. 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 $\mathfrak{M}' = (S, s_{\mathit{init}}, E, \mathbf{P}', \mathfrak{C}')$ is a polynomial augmented pMC.

3.1. Linear equation systems with polynomial coefficients

Let $x_1, \ldots, x_k$ be parameters, $\overline{x} = (x_1, \ldots, x_k)$. We consider linear equation systems of the form $A \cdot p = b$, where $A = (a_{i,j})_{i,j=1,\ldots,n}$ is a non-singular $n \times n$-matrix with $a_{i,j} = a_{i,j}(\overline{x}) \in \mathbb{Q}[\overline{x}]$. Likewise, $b = (b_i)_{i=1,\ldots,n}$ is a vector of length n with $b_i = b_i(\overline{x}) \in \mathbb{Q}[\overline{x}]$. The solution vector $p = (p_i)_{i=1,\ldots,n}$ is a vector of rational functions $p_i = f_i / g_i$ with $f_i, g_i \in \mathbb{Q}[\overline{x}]$. By Cramer's rule, we obtain $p_i = \frac{\det(A_i)}{\det(A)}$, where $\det(A)$ is the determinant of A, and $\det(A_i)$ is 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 $n \cdot d$.

We first consider upper and lower bounds on the number of monomials in the solution for 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 1. *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 and n the number of rows in A.*

Proof. As observed above, 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, if $d = 1$ the polynomials f_i and g_i have degree at most n. The upper bound on the number of monomials is now obtained by simple combinatorics.

Furthermore, an estimate for this upper bound is $\left(2\frac{n}{k}\right)^k \leqslant \binom{n+k}{k} \leqslant \left(3\frac{n+k}{k}\right)^k$. So the number of monomials is at most exponential in k. □

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

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

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

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

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

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 $\mathrm{Pr}_{s_{init}}^{\mathfrak{M}(\overline{x})}(\lozenge\, goal)$ has the form $\sum_{I\subseteq\{1,\ldots,k\}} \alpha_I \cdot \prod_{i\in I} x_i$ with non-zero coefficients $\alpha_I \in \mathbb{Q}$. The number of monomials with non-zero coefficients is therefore in $\mathcal{O}(2^k)$, that is, exponential in the number of parameters. □

3.2. One-step fraction-free Gaussian elimination

Fraction-free Gaussian elimination strives to avoid a fractional representation of the intermediate matrix values during matrix triangulation. For example, when starting with an integer matrix it ensures that the intermediate values are integers as well. Now, when using (naïve) fraction-free Gaussian elimination the new coefficients after the m-th step, $m = 1, \ldots, n-1$, are computed as

$$a_{i,j}^{(m)} = a_{i,j}^{(m-1)} a_{m,m}^{(m-1)} - a_{i,m}^{(m-1)} a_{m,j}^{(m-1)}$$

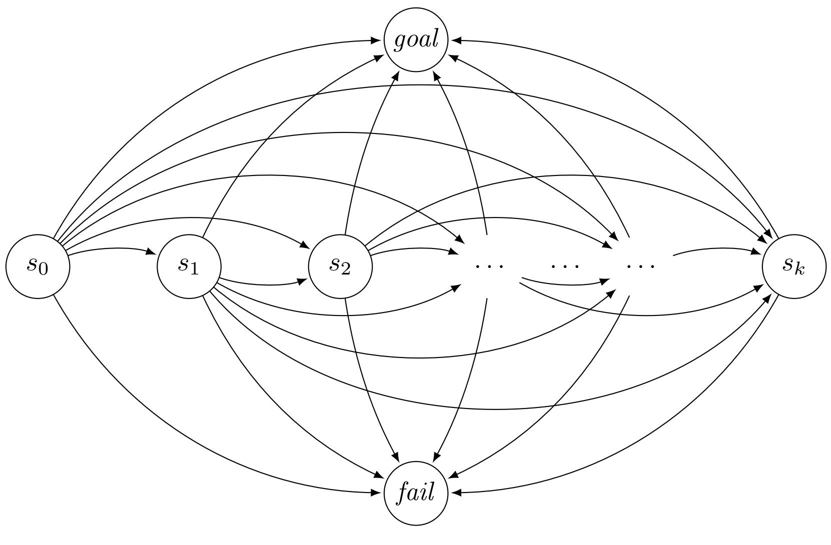


Figure 1 – Graph structure of an acyclic parametric Markov chain on parameters $x_1, \ldots, x_k$, with transition probabilities $\frac{1}{k+2}$ for transitions from s_0 to any other state, x_i for transitions from state s_i, $i = 1, \ldots, k$, to *goal*, $\frac{1}{k-i+1} \cdot (1 - x_i)$ for transitions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

Algorithm 1 One-step fraction-free Gaussian elimination [10]

```
 1: procedure FractionFreeGauss($A = (a_{ij})_{i,j=1,\ldots,n}$, $b = (b_i)_{i=1,\ldots,n}$)
 2:     $a_{0,0} = 1$
 3:     for $m = 1, \ldots, n-1$ do                                ▷ triangulation, assuming $a_{m,m} \neq 0$
 4:         for $i = m+1, \ldots, n$ do
 5:             for $j = m+1, \ldots, n$ do
 6:                 $a_{i,j} = \big(a_{m,m} \cdot a_{i,j} - a_{i,m} \cdot a_{m,j}\big)/a_{m-1,m-1}$    ▷ exploit exact divisibility by $a_{m-1,m-1}$
 7:             $b_i = \big(a_{m,m} \cdot b_i - a_{i,m} \cdot b_m\big)/a_{m-1,m-1}$                ▷ exploit exact divisibility by $a_{m-1,m-1}$
 8:             $a_{i,m} = 0$
 9:     for $m = n-1, \ldots, 1$ do                                ▷ back substitution
10:         $b_m = \big(a_{n,n} \cdot b_m - \sum_{i=m+1}^{n} a_{m,i} \cdot b_i\big)/a_{m,m}$      ▷ exploit exact divisibility by $a_{m,m}$
11:     return $\big(b_i/a_{n,n}\big)_{i=1,\ldots,n}$                          ▷ rational solution functions
```

for $i, j = m+1, \ldots, n$, where $a_{i,j}^{(0)} = a_{i,j}$. The b_i are updated analogously. When applied to systems with polynomial coefficients this results in doubling the degree in each step, so the degree grows exponentially.

In *one-step fraction-free* Gaussian elimination [10] (see Algorithm 1), the computation of the coefficients in step m 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 $a_{0,0}^{(0)} = 1$, analogously for the b_i. 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 of $a_{i,j}^{(m-1)} a_{m,m}^{(m-1)} - a_{i,m}^{(m-1)} a_{m,j}^{(m-1)}$. Here, the application of division limits the growth of the polynomials, and, as an exact divisor is known by construction, the costly computation of the greatest common divisor, which is otherwise used in practice to limit this growth by keeping numerator and denominator of the rational functions in the matrix coprime, is avoided.

Lemma 3. *Let $\mathfrak{M} = (S, s_{\mathit{init}}, E, \mathbf{P})$ be a polynomial pMC on $x_1, \ldots, x_k$, and $T \subseteq S$. Let $n = |S|$, and $d = \max_{s,t \in S} \deg(\mathbf{P}(s,t))$. The rational functions for the reachability probabilities for reaching T, $\Pr_s^{\mathfrak{M}}(\Diamond T)$, the expected accumulated weight until reaching T, $\mathrm{E}_s^{\mathfrak{M}}\big(\oplus T\big)$, if $\Pr_s^{\mathfrak{M}}(\Diamond T) = 1$ for all $s \in S$, and the expected mean payoff for T, $\mathrm{E}_s^{\mathfrak{M}}\big(\mathrm{mp}(T)\big)$ are all computable in $\mathcal{O}\big(\mathrm{poly}(n,d)^k\big)$.*

Note that $\mathrm{Pr}_s^{\mathfrak{M}}(\Diamond T) = 1$ denotes here that $\mathrm{Pr}_s^{\mathcal{M}}(\Diamond T) = 1$ for all MCs $\mathcal{M} \in [\![\mathfrak{M}]\!]$, i.e., for all MCs that arise from admissible parameter valuations. All those Markov chains share the same underlying graph structure $G_{\mathcal{M}}$ and qualitative reachability does not depend on the parameter valuations but only on the graph structure. This assumption can thus be checked using graph algorithms in polynomial time (e.g., cf. [26]).

Proof. The result is obtained via one-step fraction-free Gaussian elimination (Algorithm 1). 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 A and the appropriate vector b, or, when considering the expected mean payoff, two such systems. Thus, they all fall into the same complexity class. Here, we only consider the reachability probabilities.

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 $\mathcal{O}\big(\mathrm{poly}(m, d)^k\big)$, so for the full one-step fraction-free Gaussian elimination it is $\mathcal{O}\big(\mathrm{poly}(n, d)^k\big)$. □

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 $\mathrm{Pr}_s^{\mathfrak{M}}(\Diamond goal) = f_s/g_s$ is in $\mathcal{O}(n \cdot d)$.

Proposition 4.3 in [3] states that the rational functions 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 [3]) are computable in polynomial time. The statement contradicts Lemma 2 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, [3, Proposition 4.3] is correct for any fixed number of variables:

Corollary 4. *Let $\mathfrak{M}$ be a polynomial pMC over k parameters and $T \subseteq S$. The rational functions for the reachability probabilities for reaching T, $\mathrm{Pr}_s^{\mathfrak{M}}(\Diamond T)$, the expected accumulated weight until reaching T, $\mathrm{E}_s^{\mathfrak{M}}\big(\diamondplus T\big)$, if $\mathrm{Pr}_s^{\mathfrak{M}}(\Diamond T) = 1$ for all $s \in S$, and the expected mean payoff for T, $\mathrm{E}_s^{\mathfrak{M}}\big(\mathrm{mp}(T)\big)$ are all computable in polynomial time.*

Another observation concerns the case where only the right-hand side of the linear equation system is parametric. Systems of this form occur, for example, when considering expectation properties for (non-parametric) MCs with parametric weights.

Lemma 5. *Let $A \cdot p = b$ be a parametric linear equation system as defined above where A is parameter-free. Then the solution vector $p = (p_i)_{i=1,\ldots,n}$ consists of polynomials of the form $p_i = \sum_{j=1}^{n} \beta_j \cdot b_j$ with $\beta_j \in \mathbb{Q}$ and can be computed in polynomial time.*

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

Since

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

$b_i^{(m)}$ 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 $b_i^{(n-1)}$ are of the form $b_i^{(n-1)} = \sum_{j=1}^{n} \sigma_j \cdot b_j$ with $\sigma_j \in \mathbb{Q}$, $j = 1, \ldots, n$.

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 $p_i = \sum_{j=1}^{n} \beta_j \cdot b_j$ with $\beta_j \in \mathbb{Q}$, $j = 1, \ldots, n$. 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. □

3.3. Stratification via SCC-decomposition

It is well known (e. g., [27, 9]) 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*. The topological ordering ensures that the coefficient matrix already has 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 $i = 1, \ldots, n{-}1$, $a^*(\text{SCC at } i) = a^*(\text{SCC at } i{+}1)$ 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 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.

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.

3.4. Implementation and Experiments

To perform an experimental evaluation of the one-step fraction-free Gaussian elimination (*GE-ff*) approach in the context of probabilistic model checking, we have implemented this method (including the SCC decomposition and topological ordering described above)[3] as an alternative solver for parametric linear equation systems in the state-of-the-art probabilistic model checker Storm [8], building upon version 1.2.1.

We compare *GE-ff* against the two solvers provided by Storm for solving parametric equation systems, i. e., the solver based on the *eigen* linear algebra library [28], and on state elimination (*state-elim*) [4]. Both of Storm's solvers use partially factorized representations of the rational functions provided by the CArL library[4]. In this representation, all factors in the numerator polynomial and all factors in the denominator polynomial are guaranteed to share no common divisor. This representation, together with caching, is often beneficial [9], due to improved performance of the gcd-computations during the simplification steps.

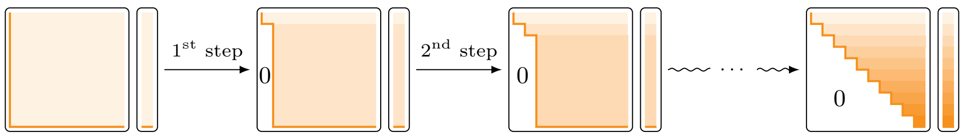
In addition to the fraction-free approach, our solver can also be instantiated to perform a straightforward Gaussian elimination (*GE*), using any of the representations for rational functions provided by the CArL library. We consider here the Gaussian elimination with the (partially) factorized representation (Storm's default, which is also used by the other standard solvers) and with a plain representation using fully expanded polynomials for numerator and denominator, using gcd computations to ensure that those are coprime.

Experimental studies.[5] 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

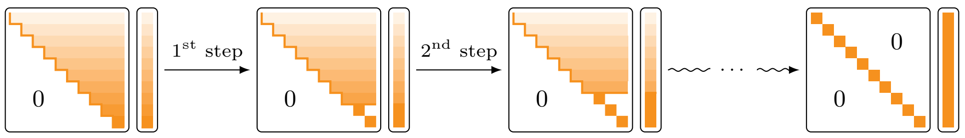
[3]In contrast to our implementation used for the experiments reported in [14], the implementation here has been improved by switching to a sparse instead of a dense matrix representation, which speeds up the processing especially for large systems. In particular, for the benchmarks reported in [14], the *GE-ff* implementation ran into the memory limit of 30GB for several of the instances, while our sparse implementation now stays within the memory limit for all considered instances. In addition, the implementation in [14] was based on version 1.0.1 of Storm. We have observed that, on the same hardware as used in [14], the standard equation solvers in version 1.2.1 of Storm showed some speed-ups over version 1.0.1, likely due to some optimizations in the underlying math library. The statistics for those solvers presented here thus differ from those presented in [14].

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

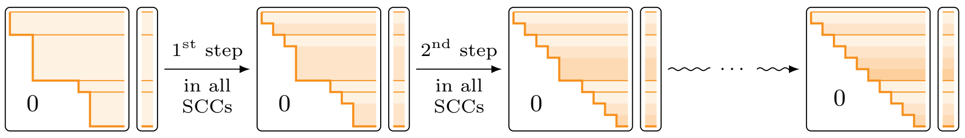
[5]The source code of our extension of Storm and the artifacts of the experiments are available at `https://wwwtcs.inf.tu-dresden.de/ALGI/PUB/Fraction-Free-Gauss/`



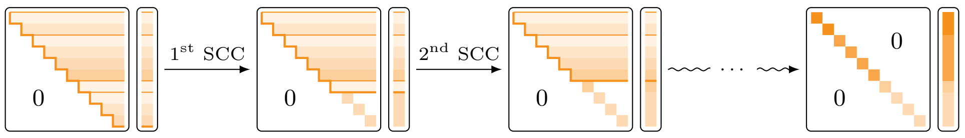
(a) One-step fraction-free elimination without topological ordering of SCCs.



(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.

solvers run single-threaded. We report the time actually spent for solving the parametric equation system by the different solvers. Other parts of model checking (model building, preprocessing) are independent of the chosen solver. We have compared the solutions obtained by the different solvers and verified that they are the same.

As the ordering of rows and columns in the matrix, respectively the order in which the equation system is processed, can play a significant role in the performance of the solution algorithm, we consider multiple variants for the four major approaches (*eigen*, *state-elim*, *GE*, *GE-ff*). We considered the following variants and denote each with a number that allows us to refer to them from the statistics tables: For the *eigen* solver, we consider the default ordering (1) and a variant, where the matrix has been topologically sorted (2). For the *state-elim* solver we consider each of the following elimination order variants (without/with preceding topological ordering of the matrix): "fw" (3/4), "fwRev" (5/6), "bw" (7/8), "bwRev" (9/10), which implement ordering based on the distance between a state and the initial state or the state and the target state, "regex" (11/12), based on the ordering proposed by [2], and "dpen" (13/14) and "spen" (15/16), which employ dynamic or static penalities to determine the ordering. By default, Storm uses the *eigen* solver (variant 1), the default order for the *state-elim* solver is "fwRev", i.e., variant 5.

For the newly implemented "normal" Gaussian elimination solver *GE*, we consider the variant with (partially) factorized representation of the rational functions (17) and with fully expanded, coprime polynomials (18), both with preceding topological ordering.

For our fraction-free Gaussian elimination (*GE-ff*) implementation, we consider a standard variant as described above (19), where topological sorting and separate handling of the SCCs is performed. Another variant (20) uses a single, global denominator for the back substitution step, instead of using per-SCC

Table 2: Statistics for "complete pMC". Matrix rows and number of distinct parameters, as well as best-case/worst-case times for solving the parametric equation system per solver. For $n = 7$, all solvers timed out (30min).

n	rows	param.	*eigen*	*state-elim*	*GE*	*GE-ff*	*red(GE-ff)*
4	4	20	^{1}0.43 s	^{4}0.64 s	^{17}0.53 s	^{21}0.00 s	0.01 s
			^{2}*0.44* s	^{6}*0.66* s	*timeout*	^{20}*0.01* s	*0.01* s
5	5	30	^{1}43.33 s	^{12}39.81 s	^{17}59.15 s	^{21}0.12 s	1.62 s
			^{2}*44.76* s	^{7}*42.29* s	*timeout*	^{19}*0.37* s	*1.64* s
6	6	42	timeout	timeout	timeout	^{21}18.65 s	24.15 s
						^{19}*92.59* s	*23.60* s

denominators as explained in Sec. 3.3. A third variant (21) does not perform any SCC stratification, while a fourth variant (22) does not perform any SCC stratification and does not reorder the matrix topologically. The last variant thus represents a straightforward implementation of one-step fraction-free Gaussian elimination.

In the following tables we report model and timing statistics for the various model instances. First, we report the number of rows of the parametric equation system (matrix) that is passed on to the equation system solvers, as well as the number of distinct parameter variables in the polynomials in the matrix. We then report the best-case run times for solving the equation system, i.e., obtaining a rational function for all states, grouped by the four approaches (*eigen*, *state-elim*, *GE*, *GE-ff*). With a super-script, we report the identifier of the fastest variant (see above). For fraction-free Gaussian elimination, we first report the time until a closed-form solution for all states is obtained, i.e., rational functions with fully expanded numerator and denominator polynomials. As the numerator and denominator of these rational functions are not necessarily coprime, we list as well the time needed for simplification *red(GE-ff)* via division by the gcd. Depending on the use-case, the non-simplified solution might be sufficient, allowing to avoid this additional step relying on potentially costly gcd-computations. The time in the *red(GE-ff)* column is reported for the same *GE-ff* variant as reported in the *GE-ff* column.

We are also interested in the dependency on the chosen configuration, i.e., a sense of the variability of the run times for each approach. In a second row (in gray and italics), we report the run times (and variant identifier) of the variant that terminated last (within the timeout), or timeout if at most one configuration terminated. A small † indicates that there were other variants that had a timeout.

It is well-known that bisimulation quotienting (with strong or weak bisimulation) can have a significant impact on the performance of model checking, in particular for parametric model checking. Where bisimulation quotienting proved to yield beneficial reductions, we note where it has been applied and sometimes report statistics for both the original and the quotiented model.

We have considered 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.

This family of pMCs $\mathfrak{M}_n = (S, s_{\mathit{init}}, E, \mathbf{P})$ has n regular states and a *goal* and *fail* state, i.e., the state space $S = \{s_1, \ldots, s_n, \mathit{fail}, \mathit{goal}\}$, and has 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 $1 \leqslant i, j \leqslant n$. 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 $1 \leqslant i \leqslant n$, and $\mathbf{P}(s_i, \mathit{fail}) = 1 - \mathbf{P}(s_i, \mathit{goal}) - \sum_{1 \leqslant j \leqslant n} \mathbf{P}(s_i, s_j)$. Overall, the pMC is defined on parameters $(x_{1,1}, \ldots, x_{1,n}, x_{1,\mathit{goal}}, \ldots, x_{n,1}, \ldots, x_{n,n}, x_{n,\mathit{goal}})$, 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., $\Pr^{\mathfrak{M}_n}_{s_{\mathit{init}}}(\lozenge \mathit{goal})$.

Table 2 summarizes statistics for the corresponding computations. As can be seen, the fraction-free approach significantly outperforms all of Storm's standard solvers, as well as normal Gaussian elimination, and scales to a higher number of parameters. Using profiling of the invocations of the gcd-computations we have determined that the *eigen*, *state-elim* and *GE* solvers spent more than 99% of their computation time for these instances in gcd-computations. For $n = 5$, the reported best-case variant for *eigen* (1) invoked the

Table 3: Statistics for “Israeli-Jalfon”, with strong bisimulation quotienting. Matrix rows and number of distinct parameters, as well as best-case/worst-case times for solving the parametric equation system per solver.

N	K	rows	param.	*eigen*	*state-elim*	*GE*	*GE-ff*	*red(GE-ff)*
4	2	11	4	^{2}0.13 s	^{12}0.16 s	^{17}0.27 s	^{21}0.01 s	0.02 s
				^{1}*0.13* s	^{8}*0.57* s	^{18}*0.97* s	^{22}*0.09* s	*0.02* s
4	3	21	4	^{2}0.43 s	^{12}0.52 s	^{17}0.54 s	^{19}0.04 s	0.19 s
				^{1}*0.86* s	^{9}*4.50* s	^{18}*1.66* s	^{22}*8.89* s	*0.24* s
4	4	15	4	^{2}0.36 s	^{12}0.39 s	^{17}0.47 s	^{19}0.04 s	0.12 s
				^{1}*0.78* s	^{8}*1.64* s	^{18}*1.48* s	^{22}*1.65* s	*0.16* s
5	2	16	5	^{1}15.11 s	^{15}17.55 s	^{17}23.65 s	^{21}1.77 s	0.35 s
				^{2}*23.84* s	^{8}*66.55* s	^{18}*737.59* s	^{20}*2.15* s	*0.35* s
5	3	36	5	^{2}210.40 s	^{11}606.64 s	^{17}126.10 s	^{19}135.18 s	85.00 s
				^{1}*271.33* s	$^{\dagger 6}$*1081.62* s	*timeout*	$^{\dagger 20}$*144.06* s	*104.42* s
5	4	51	5	^{2}242.02 s	^{11}444.22 s	^{17}214.79 s	^{19}175.52 s	667.34 s
				^{1}*361.73* s	$^{\dagger 5}$*1239.27* s	*timeout*	$^{\dagger 20}$*520.08* s	*1240.72* s
5	5	31	5	^{2}217.15 s	^{6}1019.55 s	^{17}177.33 s	^{19}172.61 s	390.15 s
				^{1}*279.58* s	$^{\dagger 12}$*1762.25* s	*timeout*	$^{\dagger 20}$*523.92* s	*752.15* s
6	2	22	6	timeout	timeout	timeout	^{22}565.33 s	70.25 s
							^{20}*894.81* s	*69.44* s
6	3	57	6	timeout	timeout	timeout	timeout	timeout

gcd-computation 139 times, while *state-elim*(12) had 148 and *GE*(17) had 124 gcd-invocations. The maximal time spent for a single gcd-invocation was between 9 and 16 seconds, depending on the solver, indicating that for this model individual gcd-computations are expensive.

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 two. To allow further experiments with an increasing number of parameters, we considered a pMC-variant of the Israeli-Jalfon self-stabilizing protocol [29].

In the Israeli-Jalfon self-stabilizing protocol, with the model taken from the PRISM case study repository,[6] 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 K of tokens, 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 neighbor 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 neighbor. An initial gadget ensures that the K initial tokens are distributed uniformly between the processes. We then compute the expected number of steps until reaching a stable situation.

Table 3 depicts the statistics for computing the rational functions for several instances. As can be seen, the fraction-free approach is competitive against the standard solvers of Storm (*eigen* and *state-elim*). For the larger instances with $N = 5$, the “normal” Gaussian elimination implementation is competitive as well, with slightly better performance for $N = 5, K = 3$. It should be noted that, for $N = 5$, if one is interested in a simplified, coprime representation of the solutions, the simplification step for the *GE-ff* result is quite costly compared to a direct computation of the simplified representation with one of the other methods. However, for $N = 6, K = 2$, fraction-free Gaussian elimination is the only method that succeeded in computing a solution within the time bound of 30 minutes.

Benchmark case studies from [19]. Furthermore, we considered several case study instances that were

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

Table 4: Statistics for the "crowds" (reachability probability) and "zeroconf" (expected accumulated reward) benchmarks of [19]. Matrix rows and number of distinct parameters, as well as best-case/worst-case times for solving the parametric equation system per solver.

model	rows	param.	*eigen*	*state-elim*	*GE*	*GE-ff*	*red(GE-ff)*
Crowds (3,5)	715	2	^{1}0.82 s	^{5}0.67 s	^{17}1.05 s	^{19}2.59 s	65.18 s
			^{2}*0.87*s	^{4}*49.66*s	^{18}*3.02*s	$^{\dagger 20}$*2.98*s	*81.58*s
Crowds (5,5)	2928	2	^{2}5.43 s	^{5}4.95 s	^{17}6.15 s	^{19}342.49 s	timeout
			^{1}*5.51*s	^{4}*533.03*s	^{18}*17.48*s	$^{\dagger 20}$*347.82*s	*timeout*
Crowds (10,5)	25103	2	^{2}118.39 s	^{5}158.14 s	^{17}120.56 s	timeout	timeout
			^{1}*126.93*s	$^{\dagger 16}$*1735.32*s	^{18}*417.97*s		
Crowds (3,5), w-bisim	40	2	^{1}0.06 s	^{12}0.04 s	^{17}0.07 s	^{19}0.00 s	0.12 s
			^{2}*0.08*s	^{4}*0.95*s	^{18}*0.17*s	^{22}*0.31*s	*0.32*s
Crowds (5,5), w-bisim	40	2	^{1}0.06 s	^{12}0.04 s	^{17}0.07 s	^{19}0.01 s	0.10 s
			^{2}*0.08*s	^{4}*0.86*s	^{18}*0.18*s	^{22}*0.31*s	*0.31*s
Crowds (10,5), w-bisim	40	2	^{1}0.06 s	^{12}0.04 s	^{17}0.07 s	^{19}0.01 s	0.10 s
			^{2}*0.08*s	^{3}*0.86*s	^{18}*0.18*s	^{21}*0.29*s	*0.32*s
Zeroconf (1000)	1002	2	^{2}39.33 s	^{14}33.52 s	^{17}128.72 s	^{20}6.39 s	12.20 s
			^{1}*77.77*s	$^{\dagger 7}$*479.45*s	^{18}*320.84*s	^{22}*17.88*s	*11.46*s
Zeroconf (10000)	10002	2	timeout	timeout	timeout	timeout	timeout

used in [19] to benchmark parametric model checkers, namely the *brp*, *crowds*, *egl*, *nand*, *zeroconf* models. Of those, *brp*, *egl* and *nand* are acyclic models. As those models are polynomial parametric Markov chains, the rational function solutions (as well as the intermediate results) have denominator polynomials of degree zero, making the occuring gcd-computations very efficient. Solving time differences between the various approaches are thus mostly influenced by the order of processing of the rows in the matrix and the size of the intermediate rational functions.

The *crowds* and *zeroconf* case studies, however, are cyclic, as they contain non-trivial strongly connected components. Table 4 depicts statistics for instances of both case studies. For *crowds*, bisimulation quotienting (weak bisimulation, marked with "w-bisim") was particularly effective, with all considered instances having a very small state space and negligible solving times. For the non-quotiented *crowds* instances, Storm's standard solvers as well as the "normal" Gaussian elimination outperform the fraction-free *GE-ff* variants. For the smaller *zeroconf* instance in Table 4, *GE-ff* significantly outperforms the standard approaches, while the larger benchmark instance could not be solved by any of the variants within the allotted time frame.

In the same vein as the case studies considered in [19], we also consider a parametrized variant of Herman's self-stabilizing protocol [30, 31], *herman* with the model obtained from [32]. Here, the uniform coin flips in the model are replaced with a biased coin, i.e., with a single parameter that represents the probability of the coin flip succeeding. In an initial phase, each configuration (every process can either start with or without a token) is chosen uniformly. We then compute the expected number of steps of the protocol (with N processes) until a stable token configuration is reached. We report here on the model with weak bisimulation applied. As can be seen from Table 5, our fraction-free Gaussian elimination implementation significantly outperforms Storm's standard solvers, as well as the "normal" Gaussian elimination implementation. This result is a bit surprising, as these instances lead to a univariate equation system, where the gcd-computations are generally not as expensive as those for the multivariate case. Profiling of the gcd-invocations showed that the overhead indeed can be largely attributed to the gcd-computations. E.g., for the $N = 9$ instance, the reported three best-performing solvers using *eigen*, *state-elim* and *GE* spent 85% to 90% of the computation time within gcd-computations. For this instance and *eigen*(1), the gcd-computations were invoked 35440 times, for *state-elim*(16) there were 28494 and for *GE*(17) there were 28929 gcd-invocations. The maximal time spent during a single gcd-invocation was between 75ms and 130ms, i.e., the computation time was spent computing a large number of relatively simple gcd-computations.

Table 5: Statistics for “herman” (expected accumulated reward), with weak bisimulation. Matrix rows and number of distinct parameters, as well as best-case/worst-case times for solving the parametric equation system per solver.

N	rows	param.	*eigen*	*state-elim*	*GE*	*GE-ff*	*red(GE-ff)*
7	15	1	^{1}0.36 s	^{16}0.81 s	^{17}0.43 s	^{20}0.03 s	0.06 s
			^{2}*0.41* s	^{14}*1.45* s	^{18}*0.87* s	^{21}*0.15* s	*0.06* s
9	54	1	^{1}472.22 s	^{16}204.90 s	^{17}203.36 s	^{20}28.23 s	4.22 s
			^{2}*484.64* s	^{5}*376.70* s	^{18}*788.63* s	^{22}*150.60* s	*4.19* s
11	181	1	timeout	timeout	timeout	timeout	timeout

For the sake of completeness, we provide statistics for the other, acyclic models of the benchmarks in [19] as well. As noted above, due to their acyclic nature, we cannot expect significant benefits from the fration-free approach for those models. For the “egl” instances (marked with “s-bisim” where strong bisimulation quotienting was applied), *GE-ff* is competitive with the standard approaches. For the largest considered instance (8,4), the Gaussian elimination processing strategy (*GE* as well as *GE-ff*) outperforms the standard approaches of Storm. However, bisimulation quotienting is highly effective for this model, with all approaches having neglible computation times in the equation system of the quotiented model. For the “brp” instances, bisimulation (weak bisimulation, marked with “w-bisim”) again yields significant reductions in the size of the matrix and thus allows more efficient solving. Here, *GE-ff* is comparable to the standard approaches.

For the “nand” case study, we consider the computation of a reachability probability and an expected accumulated reward (Table 7). For the probability computation, we also applied weak bisimulation (marked with “w-bisim”), for the expected reward strong bisimulation (marked with “s-bisim”). *GE-ff*'s performance is again in the range of the standard variants, but can – due to the acyclic nature of the model – not demonstrate its strengths.

Overall, the experiments have shown that there are instances where the fraction-free approach can indeed have a positive impact on performance, in some cases quite dramatically. In several cases, the fraction-free implementation provided the only approach that was able to solve the equation system at all within the given time bound. It thus can be seen as a beneficial addition to the standard approaches. We still see several avenues for additional optimizations, in particular in the interplay between acyclic and cyclic parts of the models. Likewise, it might be beneficial to perform additional preprocessing steps on the parametric model to reduce its size. Additionally, as can be seen in particular in the performance of the different variants of the *state-elim* solver, the order of processing plays a paramount role in the computation times. Here, additional heuristics and support for different order variants in *GE-ff* seem to be a promising avenue for further research.

Table 6: Statistics for the benchmarks of [19], "egl" (expected accumulated reward) and "brp" (reachability probability). Matrix rows and number of distinct parameters, as well as best-case/worst-case times for solving the parametric equation system per solver.

model	rows	param.	*eigen*	*state-elim*	*GE*	*GE-ff*	*red(GE-ff)*
EGL(5,2)	33789	1	^{2}0.25 s	^{5}0.04 s	^{18}0.01 s	^{20}0.02 s	0.01 s
			^{1}*0.26 s*	^{3}*8.90 s*	^{17}*0.01 s*	$^{\dagger 19}$*9.86 s*	*0.01 s*
EGL(5,4)	74749	1	^{2}0.51 s	^{5}0.07 s	^{18}0.03 s	^{20}0.05 s	0.03 s
			^{1}*0.52 s*	^{4}*20.46 s*	^{17}*0.03 s*	$^{\dagger 19}$*48.52 s*	*0.03 s*
EGL(8,2)	3342333	1	^{2}25.16 s	^{5}3.52 s	^{17}1.50 s	^{20}2.45 s	1.18 s
			^{1}*27.01 s*	$^{\dagger 13}$*791.12 s*	^{18}*1.54 s*	*timeout*	
EGL(8,4)	7536637	1	^{2}52.56 s	^{12}10.37 s	^{17}3.09 s	^{20}5.75 s	2.73 s
			^{1}*56.90 s*	$^{\dagger 13}$*1087.59 s*	^{18}*3.39 s*	*timeout*	
EGL(5,2), s-bisim	238	1	^{1}0.00 s	^{5}0.00 s	^{17}0.00 s	^{19}0.00 s	0.00 s
			^{2}*0.00 s*	^{4}*0.02 s*	^{18}*0.00 s*	^{22}*0.07 s*	*0.00 s*
EGL(5,4), s-bisim	478	1	^{1}0.00 s	^{5}0.00 s	^{17}0.00 s	^{20}0.00 s	0.00 s
			^{2}*0.00 s*	^{4}*0.04 s*	^{18}*0.00 s*	^{22}*0.30 s*	*0.00 s*
EGL(8,2), s-bisim	466	1	^{1}0.00 s	^{5}0.00 s	^{17}0.00 s	^{20}0.00 s	0.00 s
			^{2}*0.00 s*	^{4}*0.05 s*	^{18}*0.00 s*	^{22}*0.40 s*	*0.00 s*
EGL(8,4), s-bisim	946	1	^{1}0.00 s	^{5}0.00 s	^{17}0.00 s	^{20}0.00 s	0.00 s
			^{2}*0.00 s*	^{4}*0.11 s*	^{18}*0.00 s*	^{22}*1.68 s*	*0.00 s*
BRP(128,2)	3964	2	^{1}1.76 s	^{11}0.71 s	^{17}0.95 s	^{20}0.67 s	0.20 s
			^{2}*1.77 s*	$^{\dagger 10}$*1316.94 s*	^{18}*1.20 s*	^{22}*18.45 s*	*0.21 s*
BRP(128,5)	8950	2	^{1}16.30 s	^{11}4.88 s	^{17}6.81 s	^{20}4.89 s	1.50 s
			^{2}*48.62 s*	$^{\dagger 16}$*1201.16 s*	^{18}*8.58 s*	^{22}*150.64 s*	*1.73 s*
BRP(256,2)	7932	2	^{1}7.78 s	^{11}3.15 s	^{17}4.38 s	^{20}3.11 s	0.97 s
			^{2}*7.83 s*	$^{\dagger 14}$*16.24 s*	^{18}*5.56 s*	^{22}*75.07 s*	*1.04 s*
BRP(256,5)	17910	2	^{1}76.98 s	^{11}23.68 s	^{17}34.90 s	^{20}25.04 s	8.50 s
			^{2}*362.40 s*	$^{\dagger 13}$*217.95 s*	^{18}*42.85 s*	^{22}*654.95 s*	*9.68 s*
BRP(128,2), w-bisim	768	2	^{2}0.43 s	^{12}0.27 s	^{17}0.32 s	^{20}0.19 s	0.03 s
			^{1}*0.43 s*	^{3}*443.20 s*	^{18}*0.33 s*	^{22}*1.37 s*	*0.04 s*
BRP(128,5), w-bisim	1536	2	^{1}2.42 s	^{11}1.69 s	^{17}2.13 s	^{20}1.22 s	0.21 s
			^{2}*2.43 s*	$^{\dagger 9}$*147.24 s*	^{18}*2.14 s*	^{22}*10.41 s*	*0.21 s*
BRP(256,2), w-bisim	1536	2	^{2}1.92 s	^{12}1.20 s	^{17}1.46 s	^{20}0.86 s	0.16 s
			^{1}*1.93 s*	$^{\dagger 10}$*242.88 s*	^{18}*1.51 s*	^{22}*8.28 s*	*0.16 s*
BRP(256,5), w-bisim	3072	2	^{2}11.61 s	^{12}8.20 s	^{18}10.37 s	^{20}6.02 s	1.13 s
			^{1}*11.63 s*	$^{\dagger 9}$*1333.13 s*	^{17}*10.87 s*	^{22}*84.06 s*	*1.08 s*

Table 7: Statistics for the benchmarks of [19], “NAND”, for a reachability probability (above) and an expected accumulated reward (below).

model	rows	param.	*eigen*	*state-elim*	*GE*	*GE-ff*	*red(GE-ff)*
NAND (10,1)	4660	2	^{2}0.44 s	^{9}0.19 s	^{18}0.21 s	^{20}0.17 s	0.03 s
			^{1}0.47 s	*^{8}79.33 s*	*^{17}0.22 s*	*^{22}187.04 s*	*0.03 s*
NAND (10,3)	18520	2	^{2}3.21 s	^{5}1.29 s	^{17}1.55 s	^{20}1.80 s	0.26 s
			^{1}3.29 s	*$^{\dagger 13}$124.51 s*	*^{18}1.93 s*	*$^{\dagger 19}$4.81 s*	*0.25 s*
NAND (10,5)	32380	2	^{1}8.14 s	^{5}3.13 s	^{17}3.80 s	^{20}6.34 s	0.69 s
			^{2}8.58 s	*$^{\dagger 13}$330.74 s*	*^{18}5.64 s*	*$^{\dagger 19}$15.70 s*	*0.67 s*
NAND (20,1)	49040	2	^{1}10.71 s	^{5}3.54 s	^{17}4.14 s	^{20}4.05 s	0.61 s
			^{2}10.76 s	*$^{\dagger 10}$3.72 s*	*^{18}4.36 s*	*$^{\dagger 19}$24.95 s*	*0.60 s*
NAND (20,3)	202260	2	^{2}93.42 s	^{5}26.43 s	^{17}32.03 s	^{20}60.21 s	6.15 s
			^{1}97.51 s	*$^{\dagger 12}$29.10 s*	*^{18}49.31 s*	*$^{\dagger 19}$434.74 s*	*6.27 s*
NAND (20,5)	355480	2	^{2}249.46 s	^{9}68.43 s	^{17}85.56 s	^{20}198.86 s	16.97 s
			^{1}372.04 s	*$^{\dagger 11}$74.34 s*	*^{18}145.66 s*	*$^{\dagger 19}$1350.30 s*	*16.25 s*
NAND (10,1), w-bisim	1610	2	^{2}0.22 s	^{12}0.11 s	^{18}0.11 s	^{20}0.09 s	0.01 s
			^{1}0.24 s	*^{4}52.36 s*	*^{17}0.13 s*	*^{22}21.21 s*	*0.01 s*
NAND (10,3), w-bisim	6050	2	^{2}1.64 s	^{12}0.75 s	^{17}0.83 s	^{20}0.93 s	0.09 s
			^{1}1.68 s	*$^{\dagger 8}$1481.40 s*	*^{18}1.02 s*	*$^{\dagger 21}$145.13 s*	*0.09 s*
NAND (10,5), w-bisim	10490	2	^{1}4.39 s	^{12}1.82 s	^{17}1.97 s	^{20}3.25 s	0.23 s
			^{2}5.12 s	*$^{\dagger 14}$41.48 s*	*^{18}2.95 s*	*$^{\dagger 21}$449.85 s*	*0.23 s*
NAND (20,1), w-bisim	20870	2	^{2}6.56 s	^{11}2.37 s	^{18}2.68 s	^{20}2.36 s	0.27 s
			^{1}7.57 s	*$^{\dagger 16}$730.47 s*	*^{17}2.72 s*	*$^{\dagger 19}$6.21 s*	*0.28 s*
NAND (20,3), w-bisim	84550	2	^{2}59.39 s	^{11}18.02 s	^{17}20.42 s	^{20}36.31 s	2.50 s
			^{1}66.61 s	*$^{\dagger 14}$1487.06 s*	*^{18}30.73 s*	*$^{\dagger 19}$98.87 s*	*2.53 s*
NAND (20,5), w-bisim	148230	2	^{2}188.31 s	^{11}46.07 s	^{17}53.06 s	^{20}126.13 s	7.07 s
			^{1}191.44 s	*$^{\dagger 5}$445.97 s*	*^{18}91.87 s*	*$^{\dagger 19}$325.24 s*	*6.91 s*
NAND (10,1)	7381	2	^{2}0.50 s	^{9}0.13 s	^{18}0.09 s	^{20}0.06 s	0.02 s
			^{1}0.53 s	*^{8}454.52 s*	*^{17}0.14 s*	*^{22}519.03 s*	*0.02 s*
NAND (10,3)	21241	2	^{2}2.26 s	^{5}0.74 s	^{17}0.85 s	^{20}0.77 s	0.14 s
			^{1}2.28 s	*$^{\dagger 14}$109.65 s*	*^{18}0.91 s*	*$^{\dagger 19}$4.74 s*	*0.15 s*
NAND (10,5)	35101	2	^{1}5.99 s	^{5}2.17 s	^{17}2.56 s	^{20}3.76 s	0.45 s
			^{2}6.32 s	*$^{\dagger 14}$294.63 s*	*^{18}3.56 s*	*$^{\dagger 19}$14.60 s*	*0.45 s*
NAND (20,1)	78311	2	^{1}10.36 s	^{9}1.45 s	^{18}0.92 s	^{20}0.59 s	0.23 s
			^{2}10.49 s	*$^{\dagger 12}$1.66 s*	*^{17}1.48 s*	*$^{\dagger 19}$53.77 s*	*0.23 s*
NAND (20,3)	231531	2	^{2}50.01 s	^{9}9.79 s	^{17}11.82 s	^{20}14.11 s	2.06 s
			^{1}52.67 s	*$^{\dagger 12}$11.03 s*	*^{18}14.15 s*	*$^{\dagger 19}$506.92 s*	*2.20 s*
NAND (20,5)	384751	2	^{1}150.25 s	^{9}35.05 s	^{17}44.04 s	^{20}90.07 s	8.60 s
			^{2}152.39 s	*$^{\dagger 12}$39.43 s*	*^{18}68.19 s*	*$^{\dagger 19}$1421.21 s*	*8.25 s*
NAND (10,1), s-bisim	5381	2	^{1}0.32 s	^{9}0.09 s	^{18}0.06 s	^{20}0.04 s	0.01 s
			^{2}0.33 s	*^{8}170.46 s*	*^{17}0.09 s*	*^{22}274.14 s*	*0.02 s*
NAND (10,3), s-bisim	15461	2	^{1}1.57 s	^{5}0.51 s	^{17}0.58 s	^{20}0.52 s	0.11 s
			^{2}1.98 s	*$^{\dagger 14}$27.68 s*	*^{18}0.64 s*	*$^{\dagger 21}$1045.39 s*	*0.10 s*
NAND (10,5), s-bisim	25541	2	^{2}4.10 s	^{5}1.46 s	^{17}1.75 s	^{20}2.59 s	0.34 s
			^{1}4.31 s	*$^{\dagger 14}$66.60 s*	*^{18}2.48 s*	*$^{\dagger 19}$8.38 s*	*0.35 s*
NAND (20,1), s-bisim	63311	2	^{2}7.19 s	^{9}1.15 s	^{18}0.71 s	^{20}0.43 s	0.19 s
			^{1}8.39 s	*$^{\dagger 14}$351.69 s*	*^{17}1.13 s*	*$^{\dagger 19}$35.20 s*	*0.19 s*
NAND (20,3), s-bisim	187371	2	^{2}37.58 s	^{5}7.69 s	^{17}9.32 s	^{20}10.69 s	1.68 s
			^{1}38.57 s	*$^{\dagger 12}$8.49 s*	*^{18}10.87 s*	*$^{\dagger 19}$332.94 s*	*1.72 s*
NAND (20,5), s-bisim	311431	2	^{1}143.16 s	^{9}27.24 s	^{17}33.79 s	^{20}68.68 s	7.00 s
			^{2}250.48 s	*$^{\dagger 12}$30.03 s*	*^{18}52.88 s*	*$^{\dagger 19}$950.92 s*	*6.97 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 dual to each other, i.e., the answer for the universal variant (MC-U) is obtained by negating the answer for (MC-E) with formula $\neg\Phi$, and vice versa. We concentrate on (All) in Section 4.1 and the existential model-checking problem (MC-E) for general pMCs and PCTL, and subclasses thereof, in Sections 4.2 through 4.4, respectively.

The results primarily rely upon the following result from [33]: The existential theory of the reals is known to be in PSPACE and NP-hard, and the upper bound on its time-complexity is $\ell^{k+1} \cdot d^{\mathcal{O}(k)}$, where ℓ is the number of constraints, d the maximum degree of the polynomials in the constraints, and k the number of parameters.

4.1. Computing all satisfying parameter valuations (All)

As before, let $X = X_{\mathfrak{M}}$ denote 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:

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

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 represent 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

$$T = \mathrm{Sat}_{\mathfrak{M}}(\Phi)(\overline{\xi}) \text{ iff } \overline{\xi} \models \gamma \text{ for some } (\gamma, T) \in \mathtt{Sat}_{\mathfrak{M}}(\Phi).$$

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 of Φ 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 `true`, that is, the leaf nodes of the DAG:

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

Consider now an inner node v of the DAG labelled by formula Ψ. Under the assumption that the children of v have already been treated, i. e., the satisfaction sets of the proper subformulae 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:

$$\begin{array}{lll} \mathrm{Pr}_s^{\mathfrak{M}}(\bigcirc T) & \bowtie & c \quad \text{for each state } s \in R, \\ \mathrm{Pr}_s^{\mathfrak{M}}(\bigcirc T) & \not\bowtie & c \quad \text{for each state } s \in S' \setminus R. \end{array}$$

- If $\Psi = \mathbb{P}_{\bowtie c}(\Psi_1 \,\mathsf{U}\, \Psi_2)$ then

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

  where $\delta_{\gamma_1,T_1,\gamma_2,T_2}$ is the conjunction of the following constraints:

  $$\begin{array}{lll} \mathrm{Pr}_s^{\mathfrak{M}}(T_1 \,\mathsf{U}\, T_2) & \bowtie & c \quad \text{for each state } s \in R, \\ \mathrm{Pr}_s^{\mathfrak{M}}(T_1 \,\mathsf{U}\, T_2) & \not\bowtie & c \quad \text{for each state } s \in S \setminus R. \end{array}$$

  Here, $\mathrm{Pr}_s^{\mathfrak{M}}(T_1 \,\mathsf{U}\, T_2)$ is the rational function that has been computed using (i) a graph analysis to determine the set U of states s with $s \models \exists(T_1 \,\mathsf{U}\, T_2)$ and (ii) fraction-free Gaussian elimination (Section 3) to compute the rational functions $\mathrm{Pr}_s^{\mathfrak{N}}(\Diamond T_2)$ 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 $\mathrm{Pr}_s^{\mathfrak{M}}(T_1 \,\mathsf{U}\, T_2) = f_s/g_s$, then $\mathrm{Pr}_s^{\mathfrak{M}}(T_1 \,\mathsf{U}\, T_2) \bowtie c$ is a shorthand notation for

  $$\Big(g_s > 0 \;\;\wedge\;\; f_s - c \cdot g_s \bowtie 0\Big) \vee \Big(g_s < 0 \;\;\wedge\;\; 0 \bowtie f_s - c \cdot g_s\Big). \tag{1}$$

  The case $g_s = 0$ does not occur, as we consider only admissible valuations.

- If $\Psi = \mathbb{C}_{\mathrm{Pr}}(\psi_1, \bowtie, \psi_2)$ then the further computation depends on ψ_1, and ψ_2. Here, we only deal with the case $\psi_i = \bigcirc\, \Psi_i$, $i = 1, 2$. The cases for U, and combinations of both work similarly.

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

  where $\delta_{\gamma_1,T_1,\gamma_2,T_2,R}$ is the conjunction of the following constraints:

  $$\begin{array}{lll} \mathrm{Pr}_s^{\mathfrak{M}}(\bigcirc T_1) & \bowtie & \mathrm{Pr}_s^{\mathfrak{M}}(\bigcirc T_2) \quad \text{for each state } s \in R, \\ \mathrm{Pr}_s^{\mathfrak{M}}(\bigcirc T_1) & \not\bowtie & \mathrm{Pr}_s^{\mathfrak{M}}(\bigcirc T_2) \quad \text{for each state } s \in S \setminus R, \end{array}$$

  where $\mathrm{Pr}_s^{\mathfrak{M}}(\bigcirc T) = \sum_{t \in T} \mathbf{P}(s,t)$. Analogous to the probability operator, for cases where one of the operands uses U, $\mathrm{Pr}_s^{\mathfrak{M}}(\psi_1) \bowtie \mathrm{Pr}_s^{\mathfrak{M}}(\psi_2)$ is a shorthand form: We transform it to $\mathrm{Pr}_s^{\mathfrak{M}}(\psi_1) - \mathrm{Pr}_s^{\mathfrak{M}}(\psi_2) \bowtie 0$, and then, as in (1), construct a case distinction based on the sign of the denominators to obtain a polynomial constraint.

- The expectation and expectation comparison operators are dealt with in an analogous fashion to the probability and probability comparison operator, using the solutions to the appropriate linear equation systems.

We simplify the set $\mathtt{Sat}_{\mathfrak{M}}(\Psi)$ as follows. We first remove all pairs (γ, T) where the formula γ is not satisfiable. This can be checked using algorithms for the existential theory of the reals. Furthermore, we aggregate $\mathtt{Sat}_{\mathfrak{M}}(\Psi)$ by combining pairs with the same T-component: Instead of m pairs $(\gamma_1, T), \ldots, (\gamma_m, T) \in \mathtt{Sat}_{\mathfrak{M}}(\Psi)$, we consider a single pair $(\gamma_1 \vee \ldots \vee \gamma_m, T)$. Finally, we simplify (1) whenever, for all $\overline{\xi} \in X$, we either have $\overline{\xi} \models g_s > 0$ or $\overline{\xi} \models g_s < 0$.

To answer question (All), the algorithm finally returns the disjunction of all formulas γ with $s_{\mathit{init}} \in T$ for $(\gamma, T) \in \mathtt{Sat}_{\mathfrak{M}}(\Phi)$.

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. Together with the algorithm above, and assuming that the number of constraints in $\mathfrak{C}$ is at most polynomial in the size of S, we obtain:

Theorem 6 (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.*

4.2. Complexity bounds for (MC-E)

Combining both, 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.

Theorem 7 (PSPACE upper bound for problem (MC-E))**.** *The existential PCTL+EC model-checking problem (MC-E) for augmented 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 $\mathtt{Sat}_{\mathfrak{M}}(\Phi)$ in Section 4.1. 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 $(\chi, \{s \in S : a \in \mathcal{L}(s)\})$ 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 $\mathcal{O}(n)$ 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 $\Psi = \mathbb{P}_{\bowtie c}(\Psi_1 \,\mathsf{U}\, \Psi_2)$ 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. □

NP- and coNP-hardness of (MC-E) follow from results for IMCs [12, 13]. More precisely, [13] provides a polynomial reduction from SAT to the (existential and universal) PCTL model-checking problem for IMCs. In fact, the reduction of [13] 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 8 (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 $\Pr^{\mathfrak{M}(\overline{\xi})}_{s_{\mathit{init}}}(\Diamond a) > c$ is NP-hard, even for acyclic pMCs with the transition probabilities being linear in one parameter, and where the polynomial constraints for the parameters $x_1, \ldots, x_k$ are of the form $f(x_i) \geqslant 0$ with $f \in \mathbb{Q}[x_i]$, $\deg(f) \leqslant 2$.*

The probabilities are linear in one parameter if $\mathbf{P}(s,t) \in \bigcup_{i=1}^{k} \mathbb{Q}[x_i]$, $\deg(\mathbf{P}(s,t)) \leqslant 1$, for all $(s,t) \in E$.

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 $L_{i,j} \in \{\kappa_h, \neg\kappa_h : h = 1, \ldots, k\}$ for $i = 1, \ldots, m$, $j = 1,2,3$. Consider the pMC $\mathfrak{M} = (S, s_{\mathit{init}}, E, \mathbf{P}, \mathfrak{C})$ over the parameters $x_1, \ldots, x_k$:

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

The graph $G = (S, E)$ of $\mathfrak{M}$ is shown in Figure 3. We consider the PCTL formula $\Phi = \mathbb{P}_{>c}(\Diamond\, s_{m+1})$ with $c = \frac{1}{3^m}$. Next, we prove the correctness of the construction. We will prove that

> there exists an admissible parameter valuation $\overline{\xi}$ such that $\mathfrak{M}(\overline{\xi}) \models \Phi$ if and only if the 3CNF formula α is satisfiable.

($\Longleftarrow$): Firstly, assume that there exists a satisfying assignment $\mu : \{\kappa_1 \ldots, \kappa_k\} \to \{0,1\}$ for α. We then choose the following valuation $\xi_1, \ldots, \xi_k$ for the parameters $x_1, \ldots, x_k$ 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, both for the transition probabilities and for $\mathfrak{C}$. 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 $\mathfrak{M}$:

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

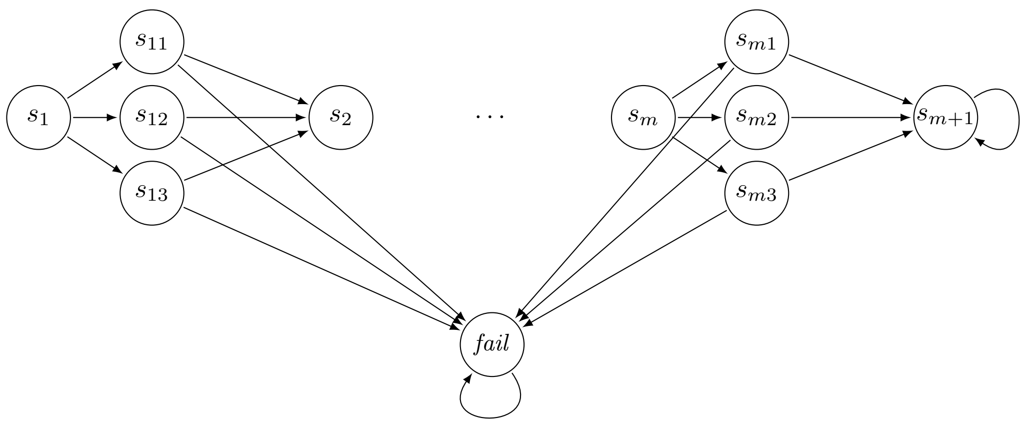


Figure 3 – Graph of the pMC constructed for reduction of 3SAT to PCTL model checking on pMCs

Hence, $\mathfrak{M}(\overline{\xi}) \models \Phi$.
($\Longrightarrow$): Suppose now that there exists an admissible valuation $\overline{\xi}$ with $\mathfrak{M}(\overline{\xi}) \models \Phi$. By admissibility and the set of constraints $\mathfrak{C}$, 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:

$$\begin{array}{rcccccccc} c & < & \mathrm{Pr}_{s_1}^{\mathfrak{M}(\overline{\xi})}\big(\Diamond\, s_{m+1}\big) & = & \mathrm{Pr}_{s_1}^{\mathfrak{M}(\overline{\xi})}\big(\Diamond\, s_{\iota}\big) \cdot \mathrm{Pr}_{s_\iota}^{\mathfrak{M}(\overline{\xi})}\big(\Diamond\, s_{\iota+1}\big) \cdot \mathrm{Pr}_{s_{\iota+1}}^{\mathfrak{M}(\overline{\xi})}\big(\Diamond\, s_{m+1}\big) \\ & < & \mathrm{Pr}_{s_\iota}^{\mathfrak{M}(\overline{\xi})}\big(\Diamond\, s_{\iota+1}\big) & \leqslant & 3 \cdot \frac{1}{3} \cdot (1{-}\wp) \;\; = \;\; 1{-}\wp \;\; = \;\; \frac{1}{3^m} \;\; = \;\; c \end{array}$$

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

The additional constraints in $\mathfrak{C}$ in the pMC used in the proof above ensure that the valuation assigns either very small or very large values to each parameter. The same effect can be obtained via an additional reachability operator:

Theorem 9 (NP-hardness for two probabilistic operators, multivariate case)**.** *Given a polynomial pMC $\mathfrak{M}$ on parameters $\overline{x}$ with initial state s_{init} and atomic propositions a_1 and a_2, and probability thresholds $c_1, c_2 \in \mathbb{Q} \cap]0, 1[$, the problem to decide whether there exists $\overline{\xi} \in X$ such that the PCTL formula*

$$\Phi = \mathbb{P}_{>c_1}(\Diamond\ a_1) \wedge \mathbb{P}_{\geq c_2}(\Diamond\ a_2)$$

is satisfied is NP-hard, even for acyclic pMCs with the transition probabilities being linear in one parameter.

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 $L_{i,j} \in \{\kappa_h, \neg\kappa_h : h = 1, \ldots, k\}$ for $i = 1, \ldots, m$, $j = 1, 2, 3$.

We consider a pMC $\mathfrak{M}$ with two structures, connected as depicted in Figure 4. We first define the blocks and then their union. The blocks ensure the following:

- B_1 encodes the 3SAT-formula, analogous to the proof for Theorem 8 and depicted in Figure 3. The structure of B_1 is relevant for the probability to reach states labelled a_1, i.e., only state s_{m+1}.

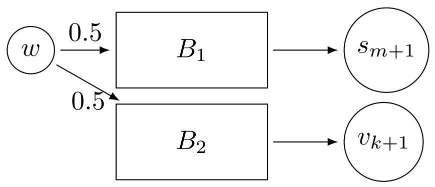


Figure 4 – General structure for the pMC constructed for the reduction of 3SAT to PCTL model checking on pMCs without additional constraints

- B_2 restricts the values for each variable x_i to $\xi_i \notin \,]l,u[$. The restriction is deduced from a necessary condition for the probability of reaching a_2, i.e., only state v_{k+1}.

We assume here $u = 1 - l$, and in particular, we choose $l = \frac{1}{3^m}$.

Block B_1 is as discussed before; formally consider the pMC $\mathfrak{M}_1 = (S_1, s^1_{\mathit{init}}, E_1, \mathbf{P}_1)$ over the parameters $x_1, \ldots, x_k$:

$$\begin{aligned} S_1 &= \{s_i, s_{i,1}, s_{i,2}, s_{i,3} : i = 1, \ldots, m\} \cup \{s_{m+1}, \mathit{fail}\}, \\ s^1_{\mathit{init}} &= s_1, \\ E_1 &= \big\{(s_i, s_{i,j}), (s_{i,j}, s_{i+1}), (s_{i,j}, \mathit{fail}) : i = 1, \ldots, m, j = 1, 2, 3\big\} \\ &\quad \cup \big\{(\mathit{fail}, \mathit{fail})\big\} \cup \big\{(s_{m+1}, s_{m+1})\big\} \end{aligned}$$

$$\mathbf{P}_1(s,t) = \begin{cases} \frac{1}{3} & \text{if } s = s_i,\ t = s_{i,j},\ 1 \leqslant i \leqslant m,\ j \in \{1,2,3\} \\ x_h & \text{if } s = s_{i,j},\ t = s_{i+1},\ 1 \leqslant i \leqslant m,\ j \in \{1,2,3\},\ L_{i,j} = \kappa_h \\ 1 - x_h & \text{if } s = s_{i,j},\ t = s_{i+1},\ 1 \leqslant i \leqslant m,\ j \in \{1,2,3\},\ L_{i,j} = \neg\kappa_h \\ x_h & \text{if } s = s_{i,j},\ t = \mathit{fail},\ 1 \leqslant i \leqslant m,\ j \in \{1,2,3\},\ L_{i,j} = \neg\kappa_h \\ 1 - x_h & \text{if } s = s_{i,j},\ t = \mathit{fail},\ 1 \leqslant i \leqslant m,\ j \in \{1,2,3\},\ L_{i,j} = \kappa_h \\ 1 & \text{if } s = t = s_{m+1} \text{ or } s = t = \mathit{fail} \\ 0 & \text{otherwise} \end{cases},$$

The graph $G = (S, E)$ of $\mathfrak{M}_1$ is shown in Figure 3.

Block B_2 is as in Figure 5. Formally, the pMC for B_2 is given by $\mathfrak{M}_2 = (S_2, s^2_{\mathit{init}}, E_2, \mathbf{P}_2)$ over the parameters $x_1, \ldots, x_k$:

$$\begin{aligned} S_2 &= \{v_i, v^l_i, v^u_i : i = 1, \ldots, k\} \cup \{v_{k+1}, \mathit{fail}\}, \\ s^2_{\mathit{init}} &= v_1, \\ E_2 &= \big\{(v_i, v^d_i), (v^d_i, v_{i+1}), (v^d_i, \mathit{fail}) : i = 1, \ldots, m, d \in \{l, u\}\big\} \end{aligned}$$

$$\mathbf{P}_2(s,t) = \begin{cases} x_i & \text{if } s = v_i,\ t = v^u_i,\ 1 \leqslant i \leqslant k \\ 1 - x_i & \text{if } s = v_i,\ t = v^l_i,\ 1 \leqslant i \leqslant k \\ x_i & \text{if } s = v^u_i,\ t = v_{i+1},\ 1 \leqslant i \leqslant k \\ 1 - x_i & \text{if } s = v^u_i,\ t = \mathit{fail},\ 1 \leqslant i \leqslant k \\ 1 - x_i & \text{if } s = v^l_i,\ t = v_{i+1},\ 1 \leqslant i \leqslant k \\ x_i & \text{if } s = v^l_i,\ t = \mathit{fail},\ 1 \leqslant i \leqslant k \\ 1 & \text{if } s = t = v_{k+1} \\ 0 & \text{otherwise} \end{cases},$$

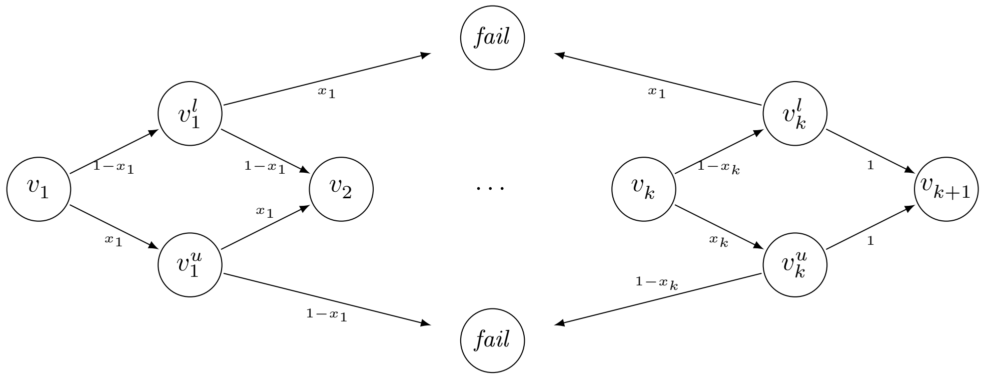


Figure 5 – Block B_2, where *fail* is duplicated to avoid clutter.

Let $f(\overline{x})$ describe the probability mass reaching v_{k+1}, i.e,

$$f(\overline{x}) := \Pr_{v_1}^{\mathfrak{M}(\overline{\xi})}(\Diamond\, v_{k+1}) = \prod_{i=1}^{k}\Big(x_i^2 + (1-x_i)^2\Big)$$

Based on this definition of $f(\overline{x})$, we observe:

- $f(\overline{a}) = f(1-\overline{a})$, and
- for $\overline{l} \leqslant \overline{a} \leqslant \overline{0.5}$ (point-wise comparison) it holds that $f(\overline{a}) \leqslant f(\overline{l})$.

The pMC $\mathfrak{M}$ is then given as $\mathfrak{M} = (S_1 \cup S_2 \cup \{w\}, w, E_1 \cup E_2 \cup \{(w,s_1),(w,v_1)\}, \mathbf{P})$ with $\mathbf{P}(s,t) = \mathbf{P}_i(s,t)$ if $(s,t) \in E_i$ for some i, and $\mathbf{P}(s,t) = 0.5$ if $(s,t) \in \{(w,s_1),(w,v_1)\}$.

Let $\Phi = \mathbb{P}_{>c_1}(\Diamond\, a_1) \wedge \mathbb{P}_{\geq c_2}(\Diamond\, a_2)$, with $c_1 = 0.5 \cdot \frac{1}{3^m}$ and $c_2 = 0.5 \cdot \Big((\frac{1}{3^m})^2 + (1-\frac{1}{3^m})^2\Big)$. State s_{m+1} is the only state labelled a_1, and state v_{k+1} is the only state labelled a_2. We can encode c_1 and c_2 with polynomially many bits (in m and k). Next, we prove the correctness of the construction. We will prove that

> there exists an admissible parameter valuation $\overline{\xi}$ such that $\mathfrak{M}(\overline{\xi}) \models \Phi$ if and only if the 3CNF formula α is satisfiable.

($\Longleftarrow$): Firstly, assume that there exists a satisfying assignment $\mu\colon \{\kappa_1 \ldots, \kappa_k\} \to \{0,1\}$ for α. We then choose the following valuation $\xi_1, \ldots, \xi_k$ for the parameters $x_1, \ldots, x_k$ Let $z = \sqrt[6m]{1-\frac{1}{3^m}}$.

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

This valuation is admissible. The probability to reach v_{k+1} from v_1 in $\mathfrak{M}$ is given by $f(\overline{\xi})$. Due to the symmetry of f, $f(\xi)$ is independent of the assignment μ. Due to the construction of block B_2, the function f is independent of α:

$$f(\xi) = \Big(z^2 + (1-z)^2\Big)^k \geq (z^2)^k \geq z^{6m} \geq 1 - \frac{1}{3^m} \geq \Big(\frac{1}{3^m}\Big)^2 + \Big(1 - \frac{1}{3^m}\Big)^2.$$

We use that $k \leqslant 3m$ as there are at most $3m$ literals in formula α. v_1 is reached from s_{init} with probability 0.5, v_{k+1} is reached only via v_1 with probability greater than $\Big((\frac{1}{3^m})^2 + (1-\frac{1}{3^m})^2\Big)$, thus v_{k+1} is reached with probability at least c_2 $(*)$.

For the probability to reach s_{m+1} from s_1 in $\mathfrak{M}$, 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:

$$\mathrm{Pr}^{\mathfrak{M}(\overline{\xi})}_{s_1}\big(\Diamond\, s_{m+1}\big) \;\geqslant\; \left(\frac{1}{3}z + \frac{2}{3}(1-z)\right)^m \;=\; \frac{1}{3^m}\cdot(z+(2{-}2z))^m \;=\; \frac{1}{3^m}\cdot\underbrace{(2{-}z)}_{>1}{}^m \;>\; \frac{1}{3^m} \;=\; 2c_1.$$

The probability to reach s_1 from w is 0.5, and s_{m+1} is only reached via s_1, thus s_{m+1} is reached with probability larger than c_1 $(**)$. $(*)$ and $(**)$ together yield $\mathfrak{M}(\overline{\xi}) \models \Phi$.
$(\Longrightarrow)$: Suppose now that there exists an admissible valuation $\xi_1, \ldots, \xi_k$ of $x_1, \ldots, x_k$ with $\mathfrak{M}(\overline{\xi}) \models \Phi$. We first show that either $0 < \xi_i < l$ or $u < \xi_i < 1$.

For that, consider $\mathfrak{M}(\overline{\xi}) \models \Phi$ implies $\mathfrak{M}(\overline{\xi}) \models \mathbb{P}_{\geq c_2}(\Diamond\, a_2)$. As above, the probability to reach v_{k+1} from v_1 must be at least $\big(\frac{1}{3^m}\big)^2 + \big(1-\frac{1}{3^m}\big)^2$. Towards a contradiction, assume $\xi_j \in \;]l,u[$ for some j. Due to admissibility, $\xi_i \in (0,1)$ for all i. Now

$$f(\overline{\xi}) = \prod_{i=1}^{k}\Big(x_i^2 + \big(1-x_i\big)^2\Big) \leqslant x_j^2 + \big(1-x_j\big)^2 < \Big(\frac{1}{3^m}\Big)^2 + \Big(1-\frac{1}{3^m}\Big)^2,$$

thus $\mathfrak{M}(\overline{\xi}) \not\models \mathbb{P}_{\geq c_2}(\Diamond\, a_2)$, a contradiction. Hence, either $0 < \xi_i < l$ or $u < \xi_i < 1$.

Now, analogous to the proof of Theorem 8, suppose by contradiction that α is not satisfiable. Then, the assignment μ given by $\mu(\kappa_h) = 0$ if $\xi_h < l$ and $\mu(\kappa_h) = 1$ if $\xi_h > u$ 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, $\mathbf{P}(s_{\iota,j}, s_{\iota+1})(\overline{\xi}) \leqslant l$ for $j = 1, 2, 3$. Hence:

$$\begin{array}{rcl} c_1 & < & \mathrm{Pr}^{\mathfrak{M}(\overline{\xi})}_{s_1}\big(\Diamond\, s_{m+1}\big) \;=\; \mathrm{Pr}^{\mathfrak{M}(\overline{\xi})}_{s_1}\big(\Diamond\, s_{\iota}\big)\cdot \mathrm{Pr}^{\mathfrak{M}(\overline{\xi})}_{s_\iota}\big(\Diamond\, s_{\iota+1}\big)\cdot \mathrm{Pr}^{\mathfrak{M}(\overline{\xi})}_{s_{\iota+1}}\big(\Diamond\, s_{m+1}\big) \\ & < & \mathrm{Pr}^{\mathfrak{M}(\overline{\xi})}_{s_\iota}\big(\Diamond\, s_{\iota+1}\big) \;\leqslant\; 3\cdot\frac{1}{3}\cdot l \;=\; l \;=\; \frac{1}{3^m} \;=\; c_1 \end{array}$$

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

4.3. (MC-E) for PCTL on univariate pMCs

In many scenarios, the number of parameters is fixed, instead of increasing with the model size.

Theorem 10 (PCTL+EC model checking without nesting in P, fixed parameter case)**.** *Let Φ be a PCTL+EC formula without nested probability, expectation or comparison operators, and let $\mathfrak{M}$ be a polynomial pMC with k parameters $x_1 \ldots x_k$. The problem to decide whether there exists an admissible parameter valuation $\overline{\xi} \in X$ such that $\mathfrak{M}(\overline{\xi}) \models \Phi$ is in P.*

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 $s_{\mathit{init}} \models_{\mathfrak{M}(\xi)} \Psi$. We finally check the existence of a value ξ for the parameter x such that $\xi \models \gamma_\Phi$. This check can be done in polynomial time [33].

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 in Section 4.1. □

Theorem 11 (NP-completeness for full PCTL+EC, fixed parameter case)**.** *Let Φ be a PCTL+EC formula, and let $\mathfrak{M}$ be a polynomial pMC for a fixed set of parameters $\overline{x}$. 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.*

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 7). We use that the polynomial constraint contains only a fixed number of variables, and can therefore be checked in polynomial time [33]. NP-hardness follows from Lemma 12, given below. □

Lemma 12 (NP-hardness 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-hard.*

The hardness even holds for (1) 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 (2) for (cyclic) polynomial pMC in combination with PCTL.

Proof. 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:

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

In a first reduction (1), we reduce to the PCTL+C fragment of PCTL+EC that uses the comparison operator $\mathbb{C}_{\mathrm{Pr}}$, but not the probability operator $\mathbb{P}$, yielding NP-hardness for PCTL+EC:

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

The initial state of $\mathfrak{M}$ 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.

Consequently, $\mathfrak{M}$ 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. For $W = \{w_1, \ldots, w_h\} \subseteq S$, we slightly abuse notation, and use W to refer to the formula $w_1 \vee \ldots \vee w_h$.

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

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

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:

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

Therefore:

$$\Pr^{\mathfrak{M}(\xi)}_{v_j}(\,\Diamond ok_j\,) \;=\; \frac{\xi}{2} \;=\; \sum_{k=1}^{\infty} \frac{\xi[k]}{2^{k+1}}$$

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 $s \in Good_j$:

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

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

$$\Pr^{\mathfrak{M}(\xi)}_{v_j}(\,\Diamond Good_j\,) \;\leqslant\; \Pr^{\mathfrak{M}(\xi)}_{v_j}(\,\Diamond ok_j\,) \quad \text{iff} \quad j \in I_\xi \quad \text{iff} \quad \xi[j] = 1$$

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

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

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 $\mathbb{P}_{=1}(\Box(v_j \to \Xi_j))$.

The length of the formulae Ψ_j is 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 $\mu\colon \{\kappa_1, \ldots, \kappa_\ell\} \to \{0,1\}$. 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 $\mu \models \alpha$.

In a second reduction (2), we reduce to the PCTL fragment of PCTL+EC without the comparison operator $\mathbb{C}_{\Pr}$:

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:

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

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

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

Hence:

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

Moreover, we have:

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

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

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

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. □

4.4. (MC-E) for monotonic PCTL on univariate pMCs

The parameters in pMC typically have a fixed meaning, e. g., the 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 observation motivates the consideration of univariate pMCs and PCTL formulas that are *monotonic* in the following sense.

Given a univariate polynomial pMC $\mathfrak{M} = (S, s_{\mathit{init}}, E, \mathbf{P})$ with variable x, 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 $\xi_1(x) \leqslant \xi_2(x)$ then $\mathbf{P}(s,t)(\xi_1) \leqslant \mathbf{P}(s,t)(\xi_2)$. 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 [34]. 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 E_+ is computable in polynomial time. Let S_+ denote the set of states that are reachable only via edges in E_+. Formally, $s \in S_+$ if 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_+ are called *monotonic* states. Given the set E_+, the set S_+ can be determined by simple graph algorithms (in polynomial time).

We observe that the probabilities for path formulas along monotonic states exhibit monotonic behavior:

Lemma 13. *Let $A, B \subseteq S_+$ and φ be one of the path formulas $A \,\mathsf{U}\, B$, $A \,\mathsf{R}\, B$, $\Diamond B$, $\Box B$, or $\bigcirc B$. Then, for all ξ_1, $\xi_2 \in X$ with $\xi_1(x) < \xi_2(x)$ and all states $s \in S$:*

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

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

Let PCTL (state) formula Φ be *monotonic* if it is obtained by the following grammar:

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

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

Lemma 14. *Let $\mathfrak{M}$ be a univariate polynomial pMC and Φ a monotonic PCTL formula. Then,* $\mathrm{Sat}_{\mathfrak{M}(\xi_1)}(\Phi) \subseteq \mathrm{Sat}_{\mathfrak{M}(\xi_2)}(\Phi)$ *for any two valuations ξ_1 and ξ_2 of x with $\xi_1(x) < \xi_2(x)$.*

Proof. We prove the lemma by structural induction on Φ. Consider two arbitrary but fixed valuations ξ_1 and ξ_2 of x satisfying $\xi_1(x) < \xi_2(x)$. The claim is obvious for the case where Φ is an atomic formula $a \in S_+$. In the induction step we consider a monotonic PCTL-formula Φ 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,

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

Analogous statements hold 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 13 will now be used to establish the monotonicity property of the satisfaction functions of monotonic formulas.

Consider the case $\Phi = \mathbb{P}_{\geqslant c}(\varphi)$ where $\varphi = \Psi_1 \,\mathsf{U}\, \Psi_2$. Let $\xi_1, \xi_2 \in X$, $\xi_1(x) < \xi_2(x)$ 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 $\mathrm{Sat}_{\mathfrak{M}(\xi)}(\Phi) \subseteq S_+$. So in particular, $\mathrm{Sat}_{\mathfrak{M}(\xi_1)}(\Phi) \subseteq \mathrm{Sat}_{\mathfrak{M}(\xi_2)}(\Phi) \subseteq S_+$. $\Box$

Hence, if Φ is monotonic then the satisfaction function $X \to 2^S$, $\xi \mapsto \mathrm{Sat}_{\mathfrak{M}}(\Phi)(\xi) = \mathrm{Sat}_{\mathfrak{M}(\xi)}(\Phi)$ is monotonic, and we obtain:

Corollary 15. *Let Φ be a monotonic PCTL formula. Then there exist $\xi_\Phi \in \mathbb{R}$ and $S_\Phi \subseteq S$, called the* maximal satisfaction set of Φ, *such that* $\mathrm{Sat}_{\mathfrak{M}(\xi)}(\Phi) = S_\Phi$ *for all $\xi \in X$ with $\xi(x) \geqslant \xi_\Phi(x)$, and* $\mathrm{Sat}_{\mathfrak{M}(\xi)}(\Phi) \subseteq S_\Phi$ *for all $\xi \in X$ with $\xi(x) < \xi_\Phi(x)$.*

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 16 ((MC-E) for monotonic PCTL on univariate pMC)**.** *Let $\mathfrak{M} = (S, s_{init}, E, \mathbf{P})$ be a univariate polynomial pMC on x, and Φ a monotonic PCTL formula. Then the 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 $\Phi = \Psi_1 \vee \Psi_2$, and $\Phi = \Psi_1 \wedge \Psi_2$. 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 $\Pr^{\mathfrak{M}}_s(A \mathsf{U} B)$. Using a polynomial time algorithm for the univariate theory of the reals [34], 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. □

4.5. Model checking PCTL+EC on MCs with parametric weights

We finally consider the case where $\mathcal{M}$ is an ordinary Markov chain augmented with a parametric weight function $wgt\colon S \to \mathbb{Q}[\overline{x}]$. Given a set $T \subseteq S$ such that $\Pr^{\mathcal{M}}_s(\Diamond T) = 1$ for all states $s \in S$, the vector of the expected accumulated weights $e = (\mathrm{E}^{\mathcal{M}}_s(\diamondsuit T))_{s \in S}$ 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 $\overline{x}$. By Lemma 5, $\mathrm{E}^{\mathcal{M}}_s(\diamondsuit T)$ 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 $\mathrm{E}^{\mathcal{M}}_s(\mathrm{mp}(T)) = \sum_{\mathrm{BSCC}\ B} \Pr^{\mathcal{M}}_s(\Diamond B) \cdot \mathrm{mp}(B)(T)$ 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 17 (Time complexity of PCTL+EC model checking with parametric weights)**.** *Let $\mathcal{M}$ be an MC with parametric weights over k parameters, and Φ a PCTL+EC formula. The problem (MC-E) is solvable in time $\mathcal{O}\big(|\Phi| \cdot \mathrm{poly}\big(size(\mathcal{M}), d\big)^{k \cdot |\Phi|_{\mathbb{E},\mathbb{C}_\mathrm{E}}}\big)$, where $|\Phi|_{\mathbb{E},\mathbb{C}_\mathrm{E}}$ is the number of expectation and expectation comparison operators in Φ, and d the maximum degree of the polynomials assigned as weights in $\mathcal{M}$.*

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 10).

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 show that an implementation using one-step fraction-free Gaussian elimination has superior performance for some benchmarks, and may be beneficial in practice.

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 (cf. Table 1 in the introduction for a summary). 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. Furthermore, we have shown that an exponential blow-up in the number of parameters for a closed-form representation cannot be avoided in general, even for acyclic pMCs.

References

[1] B. Jonsson, K. G. Larsen, Specification and refinement of probabilistic processes, in: 6th Annual Symposium on Logic in Computer Science (LICS), IEEE, 1991, pp. 266–277. doi:10.1109/LICS.1991.151651.

[2] C. Daws, Symbolic and parametric model checking of discrete-time Markov chains, in: 1st Int. Colloquium on Theoretical Aspects of Computing (ICTAC), Vol. 3407 of LNCS, Springer, 2005, pp. 280–294. doi:10.1007/978-3-540-31862-0_21.

[3] R. Lanotte, A. Maggiolo-Schettini, A. Troina, Parametric probabilistic transition systems for system design and analysis, Formal Aspects of Computing 19 (1) (2007) 93–109. doi:10.1007/s00165-006-0015-2.

[4] E. M. Hahn, H. Hermanns, L. Zhang, Probabilistic reachability for parametric Markov models, Int. Journal on Software Tools for Technology Transfer 13 (1) (2011) 3–19. doi:10.1007/s10009-010-0146-x.

[5] E. M. Hahn, H. Hermanns, B. Wachter, L. Zhang, PARAM: A model checker for parametric Markov models, in: 22nd Int. Conference on Computer Aided Verification (CAV), Vol. 6174 of LNCS, Springer, 2010, pp. 660–664. doi:10.1007/978-3-642-14295-6_56.

[6] M. Z. Kwiatkowska, G. Norman, D. Parker, PRISM 4.0: Verification of probabilistic real-time systems, in: 23rd Int. Conference on Computer Aided Verification (CAV), Vol. 6806 of LNCS, Springer, 2011, pp. 585–591. doi:10.1007/978-3-642-22110-1_47.

[7] K. O. Geddes, S. R. Czapor, G. Labahn, Algorithms for Computer Algebra, Kluwer, 1993.

[8] C. Dehnert, S. Junges, J. Katoen, M. Volk, A Storm is coming: A modern probabilistic model checker, in: 29th Int. Conference on Computer Aided Verification (CAV), Vol. 10427 of LNCS, Springer, 2017, pp. 592–600. doi:10.1007/978-3-319-63390-9_31.

[9] N. Jansen, F. Corzilius, M. Volk, R. Wimmer, E. Ábrahám, J. Katoen, B. Becker, Accelerating parametric probabilistic verification, in: 11th Conference on Quantitative Evaluation of Systems (QEST), Vol. 8657 of LNCS, Springer, 2014, pp. 404–420. doi:10.1007/978-3-319-10696-0_31.

[10] E. H. Bareiss, Computational solutions of matrix problems over an integral domain, IMA Journal of Applied Mathematics 10 (1) (1972) 68–104. doi:10.1093/imamat/10.1.68.

[11] H. Hansson, B. Jonsson, A logic for reasoning about time and reliability, Formal Aspects of Computing 6 (5) (1994) 512–535. doi:10.1007/bf01211866.

[12] K. Sen, M. Viswanathan, G. Agha, 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), Vol. 3920 of LNCS, Springer, 2006, pp. 394–410. doi:10.1007/11691372_26.

[13] K. Chatterjee, K. Sen, T. A. Henzinger, Model-checking omega-regular properties of interval Markov chains, in: 11th Int. Conference on Foundations of Software Science and Computational Structures (FoSSaCS), Vol. 4962 of LNCS, Springer, 2008, pp. 302–317. doi:10.1007/978-3-540-78499-9_22.

[14] L. Hutschenreiter, C. Baier, J. Klein, Parametric Markov chains: PCTL complexity and fraction-free Gaussian elimination, in: 8th International Symposium on Games, Automata, Logics and Formal Verification (GandALF), Vol. 256 of EPTCS, 2017, pp. 16–30. doi:10.4204/EPTCS.256.2.

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

[16] R. Kannan, Solving systems of linear equations over polynomials, Theoretical Computer Science 39 (1985) 69–88. doi:10.1016/0304-3975(85)90131-8.

[17] W. Y. Sit, An algorithm for solving parametric linear systems, Journal of Symbolic Computation 13 (4) (1992) 353–394. doi:10.1016/S0747-7171(08)80104-6.

[18] G. Nakos, P. R. Turner, R. M. Williams, Fraction-free algorithms for linear and polynomial equations, ACM SIGSAM Bulletin 31 (3) (1997) 11–19. doi:10.1145/271130.271133.

[19] C. Dehnert, S. Junges, N. Jansen, F. Corzilius, M. Volk, H. Bruintjes, J. Katoen, E. Ábrahám, PROPhESY: A probabilistic parameter synthesis tool, in: 27th Int. Conference on Computer Aided Verification (CAV), Vol. 9206 of LNCS, Springer, 2015, pp. 214–231. doi:10.1007/978-3-319-21690-4_13.

[20] A. Filieri, C. Ghezzi, G. Tamburrelli, Run-time efficient probabilistic model checking, in: 33rd Int. Conference on Software Engineering (ICSE), ACM, 2011, pp. 341–350. doi:10.1145/1985793.1985840.

[21] M. Benedikt, R. Lenhardt, J. Worrell, LTL model checking of interval Markov chains, in: 19th Int. Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), Vol. 7795 of LNCS, Springer, 2013, pp. 32–46. doi:10.1007/978-3-642-36742-7_3.

[22] V. Chonev, Reachability in augmented interval Markov chains, CoRR abs/1701.02996.

[23] T. Quatmann, C. Dehnert, N. Jansen, S. Junges, J. Katoen, Parameter synthesis for Markov models: Faster than ever, in: 14th Int. Symposium on Automated Technology for Verification and Analysis (ATVA), Vol. 9938 of LNCS, Springer, 2016, pp. 50–67. doi:10.1007/978-3-319-46520-3_4.

[24] M. Cubuktepe, N. Jansen, S. Junges, J. Katoen, I. Papusha, H. A. Poonawala, U. Topcu, Sequential convex programming for the efficient verification of parametric MDPs, in: TACAS (2), Vol. 10206 of LNCS, 2017, pp. 133–150. `doi:10.1007/978-3-662-54580-5_8`.

[25] V. G. Kulkarni, Modeling and analysis of stochastic systems, Chapman & Hall, 1995.

[26] C. Baier, J.-P. Katoen, Principles of Model Checking, The MIT Press, 2008.

[27] F. Ciesinski, C. Baier, M. Größer, J. Klein, Reduction techniques for model checking Markov decision processes, in: 5th Int. Conference on Quantitative Evaluation of Systems (QEST), IEEE, 2008, pp. 45–54. `doi:10.1109/QEST.2008.45`.

[28] G. Guennebaud, B. Jacob, et al., Eigen v3, http://eigen.tuxfamily.org (2010).

[29] A. Israeli, M. Jalfon, Token management schemes and random walks yield self-stabilizing mutual exclusion, in: 9th ACM Symposium on Principles of Distributed Computing (PODC), ACM, 1990, pp. 119–131. `doi:10.1145/93385.93409`.

[30] T. Herman, Probabilistic self-stabilization, Information Processing Letters 35 (2) (1990) 63–67. `doi:10.1016/0020-0190(90)90107-9`.

[31] M. Z. Kwiatkowska, G. Norman, D. Parker, Probabilistic verification of Herman's self-stabilisation algorithm, Formal Aspects of Computing 24 (4-6) (2012) 661–670. `doi:10.1007/s00165-012-0227-6`.

[32] S. Aflaki, M. Volk, B. Bonakdarpour, J. Katoen, A. Storjohann, Automated fine tuning of probabilistic self-stabilizing algorithms, in: 36th IEEE Symposium on Reliable Distributed Systems (SRDS), IEEE Computer Society, 2017, pp. 94–103. `doi:10.1109/SRDS.2017.22`.

[33] S. Basu, R. Pollack, M.-F. Roy, Algorithms in Real Algebraic Geometry, Springer, 2008.

[34] M. Ben-Or, D. Kozen, J. Reif, The complexity of elementary algebra and geometry, Journal of Computer and System Sciences 32 (2) (1986) 251–264. `doi:10.1016/0022-0000(86)90029-2`.