Acta Informatica manuscript No.
(will be inserted by the editor)

Precise Parameter Synthesis for Stochastic Biochemical Systems

**Milan Češka · Frits Dannenberg ·
Nicola Paoletti · Marta Kwiatkowska ·
Luboš Brim**

Received: date / Accepted: date

Abstract We consider the problem of synthesising rate parameters for stochastic biochemical networks so that a given time-bounded CSL property is guaranteed to hold, or, in the case of quantitative properties, the probability of satisfying the property is maximised or minimised. Our method is based on extending CSL model checking and standard uniformisation to parametric models, in order to compute safe bounds on the satisfaction probability of the property. We develop synthesis algorithms that yield answers that are precise to within an arbitrarily small tolerance value. The algorithms combine the computation of probability bounds with the refinement and sampling of the parameter space. Our methods are precise and efficient, and improve on existing approximate techniques that employ discretisation and refinement. We evaluate the usefulness of the methods by synthesising rates for three biologically motivated case studies: infection control for a SIR epidemic model; reliability analysis of molecular computation by a DNA walker; and bistability in the gene regulation of the mammalian cell cycle.

1 Introduction

Biochemical reaction networks are a convenient formalism for modelling a multitude of biological systems, including molecular signalling pathways, logic gates built from DNA and DNA walker circuits. For low molecule counts, and assuming a well-mixed and fixed reaction volume, the prevailing approach is to model such

This work has been partially supported by the ERC Advanced Grant VERIWARE, Microsoft Research PhD Scholarship (F. Dannenberg), the Czech Grant Agency grant No. GA15-11089S (L. Brim), and the IT4Innovations Excellence in Science project No. LQ1602 (M. Češka).

M. Češka, F. Dannenberg, M. Kwiatkowska, and N. Paoletti
Department of Computer Science, University of Oxford, Wolfson Building, Parks Road, OX1 3QD Oxford, UK
E-mail: name.surname@cs.ox.ac.uk

M. Češka
Faculty of Information Technology, Brno University of Technology, Božetěchova 1/2, 612 66 Brno, Czech Republic

L. Brim
Faculty of Informatics, Masaryk University, Botanická 68a, 602 00 Brno, Czech Republic
E-mail: brim@fi.muni.cz

networks using continuous-time Markov chains (CTMCs) [20]. Stochastic model checking [31], implemented in programs such as PRISM [32], allows the analysis of the model behaviour against temporal logic properties expressed in Continuous Stochastic Logic (CSL) [3]. For instance, the reliability and performance of DNA walker circuits are evaluated using properties such as "what is the probability that the walker reaches the correct final anchorage within 10 minutes?". We envision biochemical devices that implement biosensors and medical diagnostic systems, and hence ensuring appropriate levels of reliability is important.

Stochastic model checking assumes that the model is fully specified, including reaction rate constants. However, the reaction rates can be unknown or given as estimates that typically include some measurement error. In spite of this uncertainty, one might want to still demonstrate robustness and reliability of a synthetic molecular device. Or, one might be interested in the identification of parameter values that reproduce experimentally observed behaviour. The *parameter synthesis problem*, studied for CTMCs in [13, 23], assumes a formula and a model whose rates are given as functions of parameters, and aims to compute the parameter valuations that guarantee the satisfaction of the formula. Previously the parameter synthesis problem was solved for CTMCs approximately, and only for probabilistic time-bounded reachability [23]. In this paper, we address the parameter synthesis problem for stochastic biochemical reaction networks for the full time-bounded fragment of the (branching-time) logic CSL [3]. We formulate two variants: *threshold synthesis*, which inputs a CSL formula and a probability threshold and identifies the parameter valuations which meet the threshold, and *max synthesis*, where the maximum probability of satisfying the property and the maximizing set of parameter valuations are returned.

We develop efficient synthesis algorithms that yield answers with arbitrary precision. The algorithms exploit a recently published technique that computes safe approximations to the lower and upper bounds for the probability to satisfy a CSL property over a fixed parameter space [11]. Our algorithms automatically derive the satisfying parameter regions through iterative decomposition of the parameter space up to a given tolerance value. We also demonstrate a significant speed-up of the max synthesis algorithm through the use of a sampling-based heuristic. The method is demonstrated using three case studies: the SIR epidemic model [27], where we synthesize infection and recovery rates that maximize the probability of disease extinction; the DNA walker circuit [17], where we derive stepping rates that ensure a predefined level of reliability; and a gene regulation model of the mammalian cell cycle [11], where we investigate degradation rates that lead to bi-stability.

This work is an extended version of [13], where we first introduced parameter synthesis problems and algorithms for CTMCs. In this version, we provide a rigorous treatment of the method to compute safe probability bounds and extend the approach to reward operators. We also include an additional case study on the gene regulation of the mammalian cell cycle.

Structure of the paper. In Section 2 preliminary definitions are given. In Section 3, we introduce the threshold problem and the max synthesis problem. In Section 4, we describe the methods to bound the probability of a formula for a fixed parameter region. These methods are then used in the synthesis algorithms that are described in Section 5. In Section 6, case studies and results of synthesis experiments are discussed. Related work is discussed in Section 7. Concluding remarks are given in Section 8.

2 Background

This section introduces the main concepts relevant for model checking of (parametric) continuous-time Markov chains and stochastic modelling of biochemical reactions.

2.1 Parametric CTMCs

Before we introduce parametric CTMCs, we recall the standard definition of CTMCs and describe the uniformisation procedure that is employed for their model checking based on [31].

Definition 1 (Continuous-time Markov chain (CTMC)) A CTMC is a tuple $\mathcal{C} = (S, \pi_0, \mathbf{R}, L)$ where:

- S is a finite set of *states*;
- $\pi_0 : S \to [0,1]$ is the *initial state distribution* where $\sum_{s \in S} \pi_0(s) = 1$;
- $\mathbf{R} : S \times S \to \mathbb{R}_{\geq 0}$ is the *rate matrix*; and
- $L : S \to 2^{AP}$ is a *labelling function* mapping each state $s \in S$ to the set $L(s) \subseteq AP$ of atomic propositions that hold true in s.

A transition between states $s, s' \in S$ can occur only if $\mathbf{R}(s,s') > 0$ and, in that case, the probability of triggering the transition within time t is $1 - e^{-t\mathbf{R}(s,s')}$. The time spent in state s, before a transition is triggered, is exponentially distributed with *exit rate* $E(s) = \sum_{s' \in S} \mathbf{R}(s,s')$, and when the transition occurs the probability of moving to state s' is given by $\frac{\mathbf{R}(s,s')}{E(s)}$.

A CTMC $\mathcal{C} = (S, \pi_0, \mathbf{R}, L)$ can be extended with a reward structure (ρ, ι). $\rho : S \to \mathbb{R}_{\geq 0}$ is called *state reward* and defines the rate with which a reward is acquired in state $s \in S$, e.g. a reward of $t\rho(s)$ is acquired if $\mathcal{C}$ remains in state s for t time units. The function $\iota : S \times S \to \mathbb{R}_{\geq 0}$ defines the *transition reward*, such that $\iota(s_i, s_j)$ describes the reward acquired each time the transition (s_i, s_j) occurs.

We now describe the computation of transient probabilities for CTMCs, based on standard uniformisation (also called Jensen's method or randomisation). Let $\mathbf{E}$ be a $S \times S$ diagonal matrix such that $\mathbf{E}(s_i, s_i) = E(s_i)$, and define the *generator matrix* by setting $\mathbf{Q} = \mathbf{R} - \mathbf{E}$. Then, the vector $\pi_t : S \to \mathbb{R}_{\geq 0}$ of transient probabilities at time t is given by $\pi_t = \pi_0 e^{\mathbf{Q}t}$, such that $\pi_t(s)$ is the probability of being in state s at time instant t. Below we describe the uniformisation method.

Definition 2 (Uniformised matrix) Let $\mathcal{C} = (S, \pi_0, \mathbf{R}, L)$ be a CTMC and $\mathbf{Q}$ the associated generator matrix. Then, the *uniformised matrix* $\mathbf{P}$ of $\mathcal{C}$ is defined by $\mathbf{P} = \mathbf{I} + \frac{1}{q}\mathbf{Q}$, where $q \geq \max_s\{E(s) - \mathbf{R}(s,s)\}$ is called the *uniformisation rate*.

Definition 3 (Path of a CTMC) Let $\mathcal{C} = (S, \pi_0, \mathbf{R}, L)$ be a CTMC. A path ω of $\mathcal{C}$ is a sequence $\omega = s_0 t_0 s_1 t_1 \ldots$, where for all i, $s_i \in S$ and $t_i \in \mathbb{R}_{\geq 0}$ is the time spent in state s_i. ω is infinite when $\mathbf{R}(s_i, s_{i+1}) > 0$ for all i, and finite of length n when $\mathbf{R}(s_i, s_{i+1}) > 0$ for all $i < n-1$ and $E(s_{n-1}) = 0$.

The set of paths starting in state s is denoted as $\mathrm{Path}(s)$ and a unique probability measure, Pr, exists on $\mathrm{Path}(s)$ [31]. Function $\omega(i) = s_i$ maps a position i of ω to its i-th state. The state at time t in ω is denoted as $\omega@t$, and is equal to $\omega(i)$ for the smallest i such that $\sum_{n=0}^{i} t_n \geq t$.

The transient state-probabilities by time t are obtained by standard uniformisation as a sum of state distributions after i discrete-stochastic steps, weighted by the probability of observing i jumps in a Poisson process.

Definition 4 (Transient probabilities with standard uniformisation) Let $\mathcal{C} = (S, \pi_0, \mathbf{R}, L)$ be a CTMC. Let q and $\mathbf{P}$ be the associated uniformisation rate and uniformised matrix, respectively. The vector of transient probabilities at time t, π_t, is given by standard uniformisation as follows [21,24,38]:

$$\pi_t = \sum_{i=0}^{\infty} \gamma_{i,qt} \tau_i \tag{1}$$

where $\tau_i = \pi_0 \mathbf{P}^i$ is the vector of probabilities in the discretized process at the i-th step; and $\gamma_{i,qt} = e^{-qt} \frac{(qt)^i}{i!}$ denotes the i-th Poisson probability for a process with parameter qt. An approximate value is given by finite summation

$$\hat{\pi}_t = \sum_{i=0}^{k_\epsilon} \gamma_{i,qt} \tau_i \tag{2}$$

when k_ϵ satisfies the convergence bound $\sum_{0}^{k_\epsilon} \gamma_{i,qt} \geq 1 - \epsilon$ for some $\epsilon > 0$. The Poisson terms and the summation bound k_ϵ are computed efficiently using an algorithm due to Fox and Glynn [19].

Parametric continuous-time Markov chains (pCTMCs) [23] extend the notion of CTMCs by allowing transition rates to depend on model parameters. We assume a set K of model parameters. The domain of each parameter $k \in K$ is given by a closed real interval describing the range of possible values, i.e, $[k^{\bot}, k^{\top}] \subseteq \mathbb{R}$. The parameter space $\mathcal{P}$ induced by K is defined as the Cartesian product of the individual intervals, $\mathcal{P} = \bigtimes_{k \in K} [k^{\bot}, k^{\top}]$, so that $\mathcal{P}$ is a hyper-rectangular space. Subsets of the parameter space $\mathcal{P}$ are referred to as *parameter regions* or *subspaces*.

Definition 5 (Parametric CTMC (pCTMC)) Let K be a set of parameters. A pCTMC over K is a tuple $(S, \pi_0, \mathbf{R}, L)$, where:

- S, π_0 and L are as in Definition 1; and
- $\mathbf{R} : S \times S \to \mathbb{R}[K]$ is the *parametric rate matrix*, where $\mathbb{R}[K]$ denotes the set of polynomials over the reals $\mathbb{R}$ with variables $k \in K$.

Given a pCTMC and a parameter space $\mathcal{P}$, we denote with $\mathcal{C}_{\mathcal{P}}$ the set $\{\mathcal{C}_p \mid p \in \mathcal{P}\}$ where $\mathcal{C}_p = (S, \pi, \mathbf{R}_p, L)$ is the instantiated CTMC obtained by replacing the parameters in $\mathbf{R}$ with their valuation in p. The definition restricts the rates to be polynomials, which are sufficient to describe a wide class of biological systems.

2.2 CSL for Parametric CTMCs

To specify properties over pCTMCs, we employ the time-bounded fragment of *Continuous Stochastic Logic (CSL)* [3].

Definition 6 (Time-bounded CSL) The syntax of time-bounded CSL consists of state formulas ($\varPhi$) and path formulas (ϕ) given as

$$\begin{aligned} \varPhi &::= \mathsf{true} \mid a \mid \neg\varPhi \mid \varPhi \wedge \varPhi \mid \mathrm{P}_{\sim r}[\phi] \\ \phi &::= \mathsf{X}\ \varPhi \mid \varPhi\ \mathrm{U}^{I}\varPhi \end{aligned}$$

where $a \in AP$ is an atomic proposition, $\sim\, \in \{<, \leq, \geq, >\}$, $r \in [0,1]$ is a probability threshold and I is an interval of $\mathbb{R}_{\geq 0}$.

$\mathrm{P}_{\sim r}[\phi]$ holds if the probability of the path formula ϕ being satisfied from a given state meets $\sim r$. Path formulas are defined by combining state formulas through temporal operators: $\mathsf{X}\ \varPhi$ is true if $\varPhi$ holds in the next state, $\varPhi_1\ \mathrm{U}^{I}\varPhi_2$ is true if $\varPhi_2$ holds at some time point $t \in I$, and $\varPhi_1$ holds for all time points $t' < t$. The future operator, F, and globally operator, G, are derived from U as follows:

$$\begin{aligned} \mathrm{P}_{\sim r}[\mathrm{F}^{I}\varPhi] &\equiv \mathrm{P}_{\sim r}[\mathsf{true}\ \mathrm{U}^{I}\ \varPhi] \\ \mathrm{P}_{\sim r}[\mathrm{G}^{I}\varPhi] &\equiv \mathrm{P}_{\bar{\sim} 1-r}[\mathrm{F}^{I}\ \neg\varPhi] \end{aligned}$$

where $\bar{<}\, \equiv >$, $\bar{\leq}\, \equiv \geq$, $\bar{\geq}\, \equiv \leq$ and $\bar{>}\, \equiv <$. Informally, $\mathrm{F}^{I}\varPhi$ is true if $\varPhi$ holds at some time instant in the interval I, while $\mathrm{G}^{I}\varPhi$ is true if $\varPhi$ holds for all $t \in I$. The logic can be extended with the following time-bounded reward operators [31]:

$$\mathrm{R}_{\sim r}[\mathrm{C}^{\leq t}] \mid \mathrm{R}_{\sim r}[\mathrm{I}^{=t}] \tag{3}$$

where $t, r \in \mathbb{R}_{\geq 0}$. $\mathrm{R}_{\sim r}[\mathrm{C}^{\leq t}]$ holds if the expected reward cumulated up to time t meets the bound $\sim r$, while $\mathrm{R}_{\sim r}[\mathrm{I}^{=t}]$ holds if the expected reward at time t meets $\sim r$. We now provide the formal semantics of time-bounded CSL with rewards for parametric CTMCs. To this end, we introduce two satisfaction relations, $\models_{\bot}$ and $\models_{\top}$, to describe if a CSL property holds for all and some instantiations, respectively, of a pCTMC.

Definition 7 (Semantics of time-bounded CSL for pCTMCs) Let $\mathcal{C}_{\mathcal{P}} = (S, \pi_0, \mathbf{R}, L)$ be a pCTMC over a parameter space $\mathcal{P}$ with reward structure (ρ, ι). For each state $s \in S$ the satisfaction relations $s \models_{\bot} \varPhi$ and $s \models_{\top} \varPhi$ are defined inductively by:

$$\begin{array}{llll} s \models_{\top} \mathsf{true} & \text{for all } s \in S & \qquad s \models_{\bot} \mathsf{true} & \text{for all } s \in S \\ s \models_{\top} a & \Leftrightarrow a \in L(s) & \qquad s \models_{\bot} a & \Leftrightarrow a \in L(s) \\ s \models_{\top} \neg\varPhi & \Leftrightarrow s \not\models_{\bot} \varPhi & \qquad s \models_{\bot} \neg\varPhi & \Leftrightarrow s \not\models_{\top} \varPhi \\ s \models_{\top} \varPhi \wedge \varPsi & \Leftrightarrow s \models_{\top} \varPhi \wedge s \models_{\top} \varPsi & \qquad s \models_{\bot} \varPhi \wedge \varPsi & \Leftrightarrow s \models_{\bot} \varPhi \wedge s \models_{\bot} \varPsi \end{array}$$

$$
\begin{aligned}
&s \models_{\top} \mathrm{P}_{\sim r}[\phi] \Leftrightarrow \exists p \in \mathcal{P}.\ Pr(\omega \in \mathrm{Path}(s) \mid \omega \models \phi) \sim r \text{ in } \mathcal{C}_p\\
&s \models_{\bot} \mathrm{P}_{\sim r}[\phi] \Leftrightarrow \forall p \in \mathcal{P}.\ Pr(\omega \in \mathrm{Path}(s) \mid \omega \models \phi) \sim r \text{ in } \mathcal{C}_p\\
&s \models_{\top} \mathrm{R}_{\sim r}[\mathrm{C}^{\leq t}] \Leftrightarrow \exists p \in \mathcal{P}.\ Exp(s, X_{\mathrm{C}^{\leq t}}) \sim r \text{ in } \mathcal{C}_p\\
&s \models_{\bot} \mathrm{R}_{\sim r}[\mathrm{C}^{\leq t}] \Leftrightarrow \forall p \in \mathcal{P}.\ Exp(s, X_{\mathrm{C}^{\leq t}}) \sim r \text{ in } \mathcal{C}_p\\
&s \models_{\top} \mathrm{R}_{\sim r}[\mathrm{I}^{=t}] \Leftrightarrow \exists p \in \mathcal{P}.\ Exp(s, X_{\mathrm{I}^{=t}}) \sim r \text{ in } \mathcal{C}_p\\
&s \models_{\bot} \mathrm{R}_{\sim r}[\mathrm{I}^{=t}] \Leftrightarrow \forall p \in \mathcal{P}.\ Exp(s, X_{\mathrm{I}^{=t}}) \sim r \text{ in } \mathcal{C}_p
\end{aligned}
$$

where the path formula ϕ is expanded as

$$
\begin{aligned}
&\omega \models \mathtt{X}\Phi && \Leftrightarrow \omega(1) \text{ exists and } \omega(1) \models \Phi\\
&\omega \models \Phi_1 \mathrm{U}^I\ \Phi_2 && \Leftrightarrow \exists t \in I. \text{ such that } [\omega@t \models \Phi_2 \wedge (\forall r \in [0,t).\ \omega(r) \models \Phi_1)]
\end{aligned}
$$

and $Exp(s, X)$ for $X \in \{X_{\mathrm{C}^{\leq t}}, X_{\mathrm{I}^{=t}}\}$ denotes the expectation of the random variable X with respect to the probability measure Pr over paths starting in s, defined for any $\omega = s_0 t_0 s_1 t_1 \ldots \in \mathrm{Path}(s)$ by

$$
\begin{aligned}
X_{\mathrm{C}^{\leq t}} &= \sum_{i=0}^{j_t - 1} (t_i \cdot \rho(s_i) + \iota(s_i, s_{i+1})) + \left(t - \sum_{i=0}^{j_t - 1} t_i\right) \cdot \rho(s_{j_t})\\
X_{\mathrm{I}^{=t}} &= \rho(\omega@t)
\end{aligned}
$$

where $j_t = \min\{j \mid \sum_{i=0}^{j} t_i \leq t\}$.

Note that, for formula $\mathrm{P}_{\sim r}[\phi]$, $\models_{\bot}$ and $\models_{\top}$ are defined by quantifying over $p \in \mathcal{P}$ and evaluating the satisfaction probability of ϕ on the instantiation C_p. This probability can thus be obtained using regular CSL satisfaction relation $\models$. Therefore, relations $\models_{\bot}$ and $\models_{\top}$ reduce to $\models$ when the parameter space contains only a single valuation, i.e. $\mathcal{P} = \{p\}$.

We further define the minimal satisfaction set $\mathrm{Sat}_{\bot}(\Phi)$ and the maximal satisfaction set $\mathrm{Sat}_{\top}(\Phi)$ as follows:

$$
\mathrm{Sat}_{\bot}(\Phi) = \{s \in S \mid s \models_{\bot} \Phi\} \text{ and } \mathrm{Sat}_{\top}(\Phi) = \{s \in S \mid s \models_{\top} \Phi\}. \tag{4}
$$

We now describe the *satisfaction function* to capture how the satisfaction probability of a given property relates the parameters and the initial state. For simplicity we define the function and further describe parameter synthesis only for the P operator: the method also allows a definition based on reward operators, which we describe in Section 4.3.

Definition 8 (Satisfaction function) Let ϕ be a CSL path formula, $\mathcal{C}_{\mathcal{P}}$ be a pCTMC over a space $\mathcal{P}$ and $s \in S$. We denote with $\Lambda_\phi : \mathcal{P} \to S \to [0,1]$ the satisfaction function such that $\Lambda_\phi(p)(s) = Pr(\omega \in \mathrm{Path}(s) \mid \omega \models \phi)$ in $\mathcal{C}_p$.

Since ϕ allows nested probabilistic operators, the satisfaction function is, in general, not continuous.

2.3 Stochastic models of biochemical reaction networks

Biochemical reaction networks provide a convenient formalism for describing various biological processes as a system of well-mixed reactive species in a volume of fixed size. A CTMC semantics can be derived where states describe the number of molecules of each species, and transitions correspond to reactions that consume and produce molecules. The rate matrix is defined as

$$\mathbf{R}(s_i, s_j) = \sum_{r \in \mathsf{reac}(s_i, s_j)} f_r(s_i) \tag{5}$$

where $\mathsf{reac}(s_i, s_j)$ denotes all the reactions changing state s_i into s_j and f_r is the rate function of reaction r. Recalling that the rates of a pCTMC are polynomials over the parameters, f_r can be used to describe, among others, *mass-action kinetics* [20], according to which the rate of a reaction is proportional to the concentrations of its reactants. For instance, a bimolecular chemical reaction of the form $r : A + B \rightarrow \ldots$ has rate $f_r(s_i) = k_r \frac{A(s_i)}{V} \frac{B(s_i)}{V}$, where $A(s_i), B(s_i)$ are the numbers of molecules for species A, B in state s_i, k_r is the rate constant of reaction r and V is the size of the reaction volume.

3 Problem Definition

We consider the problem of synthesizing parameters for pCTMC models of biochemical reaction networks, so that a given specification, expressed in time-bounded CSL, is satisfied. We allow models that are parametric in the rate constants and in the initial state. In contrast to previous approaches that support only specific kinds of properties (e.g. reachability as in [23]), we support the full time-bounded fragment of CSL with rewards, thus enabling generic and more expressive synthesis requirements.

We introduce two parameter synthesis problems: the *threshold synthesis* problem that, given a threshold $\sim r$ and a CSL path formula ϕ, aims to find the parameter region where the probability of ϕ meets $\sim r$; and the *max synthesis* problem that asks for the parameter region where the probability of the input formula attains its maximum, together with an interval bounding that maximum. In the latter case, all the synthesised parameters yield probabilities within this interval, but not all of them are maximising. On the other hand, solutions to the threshold synthesis problem admit parameter points left undecided. Our approach supports precise solutions through an input tolerance that limits the volume of the undecided region for the threshold synthesis problem. For max synthesis, the tolerance determines the precision of the probability interval and in turn, of the returned region. To the best of our knowledge, no other parameter synthesis methods for CTMCs exist that provide guaranteed error bounds. In the remainder of the paper, we omit the min synthesis problem that is defined and solved in a symmetric way to the max case. In addition, we assume there is a single initial state s_0, i.e. $\forall s \in S\ \pi_0[s] = 1$ if $s = s_0$, and 0 otherwise.

Problem 1 (Threshold Synthesis) Let $\mathcal{C}_\mathcal{P}$ be a pCTMC over a parameter space $\mathcal{P}$, s_0 an initial state, ϕ a CSL path formula, $\sim r$ a threshold where $r \in [0, 1]$, $\sim \in \{\leq, <, >, \geq\}$ and $\varepsilon > 0$ be a volume tolerance. The *threshold synthesis* problem is finding a partition $\{\mathcal{T}, \mathcal{U}, \mathcal{F}\}$ of $\mathcal{P}$, such that:

1. $\forall p \in \mathcal{T}.\ \Lambda_\phi(p)(s_0) \sim r$; and
2. $\forall p \in \mathcal{F}.\ \Lambda_\phi(p)(s_0) \not\sim r$; and
3. $\mathrm{vol}(\mathcal{U})/\mathrm{vol}(\mathcal{P}) \leq \varepsilon$

where $\mathrm{vol}(A) = \int_A 1 d\mu$ is the volume of A.

Observe that a Boolean combination of state formulas results in a partition of the parameter space in a natural fashion, by following a three-valued logic interpretation. For example, consider the state formula $\Phi = \mathrm{P}_{\sim r_1}[\phi_1] \ \wedge \ \mathrm{P}_{\sim r_2}[\phi_2]$. Let $\{\mathcal{T}_1, \mathcal{U}_1, \mathcal{F}_1\}$ and $\{\mathcal{T}_2, \mathcal{U}_2, \mathcal{F}_2\}$ be a partition of $\mathcal{P}$ that satisfies the threshold synthesis problem for ϕ_1 and ϕ_2, respectively, and $\varepsilon > 0$ be a tolerance value. The partition $\{\mathcal{T}, \mathcal{U}, \mathcal{F}\}$ of $\mathcal{P}$ for Φ is given as follows:

$$\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2, \quad \mathcal{T} = \mathcal{T}_1 \cap \mathcal{T}_2, \quad \mathcal{U} = \mathcal{P} \backslash (\mathcal{F} \cup \mathcal{T}) \tag{6}$$

The new partition satisfies $\mathrm{vol}(\mathcal{U})/\mathrm{vol}(\mathcal{P}) < 2\varepsilon$.

Problem 2 (Max Synthesis) Let $\mathcal{C}_\mathcal{P}$ be a pCTMC over a parameter space $\mathcal{P}$, s_0 an initial state, ϕ a CSL path formula, and $\epsilon > 0$ a probability tolerance. The *max synthesis* problem is finding a partition $\{\mathcal{T}, \mathcal{F}\}$ of $\mathcal{P}$ and probability bounds $\Lambda_\phi^\perp$, $\Lambda_\phi^\top$ such that:

1. $\Lambda_\phi^\perp - \Lambda_\phi^\top \leq \epsilon$;
2. $\forall p \in \mathcal{T}.\ \Lambda_\phi^\perp \leq \Lambda_\phi(p)(s_0) \leq \Lambda_\phi^\top$; and
3. $\exists p \in \mathcal{T}.\ \forall p' \in \mathcal{F}.\ \Lambda_\phi(p)(s_0) > \Lambda_\phi(p')(s_0)$.

The above formulation implies two important properties of the set $\mathcal{T}$: i) $\mathcal{T}$ contains all the maximising parameters and ii) all the parameters in $\mathcal{T}$ are ϵ-optimal, i.e. $\forall p \in \mathcal{T}.\ |\Lambda_\phi(p)(s_0) - \Lambda^*| \leq \epsilon$, where Λ^* is the optimal value of the satisfaction function. Note that some parameters in $\mathcal{F}$ can be also ϵ-optimal, but the third condition ensures they are not maximising.

Example 1 Figure 1 illustrates a simple birth-death process with an uncertain parameter k_1 representing the birth rate. It depicts the corresponding pCTMC and the satisfaction function Λ for a reachability property. Figure 2 illustrates the results of threshold synthesis (left) and max synthesis (right) for this model.

4 Computing Lower and Upper Probability Bounds

This section presents a generalization of the parameter exploration procedure originally introduced in [11]. The procedure takes a pCTMC $\mathcal{C}_\mathcal{P}$ and CSL path formula ϕ, and provides safe under- and over-approximations for the minimal and maximal probability that $\mathcal{C}_\mathcal{P}$ satisfies ϕ, that is, lower and upper bounds satisfying, for all $s \in S$,

$$\begin{aligned} \Lambda_{\phi,\min}(s) &\leq \inf_{p \in \mathcal{P}} \ \Lambda_\phi(p)(s) \text{ and} \\ \Lambda_{\phi,\max}(s) &\geq \sup_{p \in \mathcal{P}} \ \Lambda_\phi(p)(s). \end{aligned} \tag{7}$$

The accuracy of these approximations is improved by partitioning the parameter space $\mathcal{P}$ into subspaces and re-computing the corresponding bounds, which forms

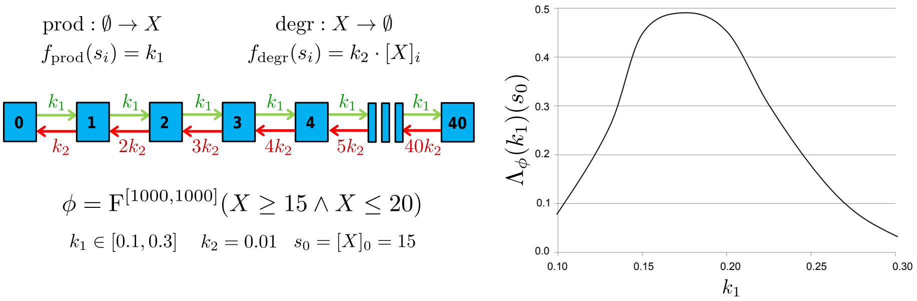


Fig. 1 **Left:** The example model contains one species X (bounded by 40) and two reactions: production of X ($\emptyset \to X$ with parametric rate k_1) and degradation of X ($X \to \emptyset$ with rate $k_2 \cdot [X]$ and $k_2 = 0.01$). $[X]_i$ denotes the number of X molecules in state s_i. The initial state s_0 is given by the population $X = 15$. The corresponding pCTMC has 41 states. Property ϕ indicates that the population of X is between 15 and 20 at time 1000. The parameter space $\mathcal{P}$ is given by the interval of the stochastic rate constant $k_1 \in [0.1, 0.3]$. **Right:** The satisfaction function Λ_ϕ.

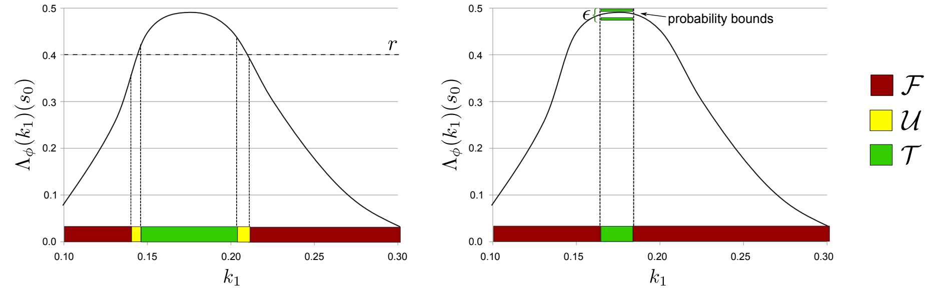


Fig. 2 Synthesis for the birth-death process of Fig. 1. **Left:** threshold synthesis for $\mathrm{P}_{\geq 0.4}[\phi]$ and with volume tolerance $\varepsilon = 5\%$. **Right:** max-synthesis with probability tolerance $\epsilon = 2\%$.

the basis of the synthesis algorithms that we discuss in the next section. We first show how to compute bounds $\Lambda_{\phi,\min}(s), \Lambda_{\phi,\max}(s)$ for unnested path formulas. Then, we extend the method to nested path formulas, by providing under- and over-approximations of the satisfaction sets $\mathrm{Sat}_\perp$ and $\mathrm{Sat}_\top$ (see Eqn 4), and to reward operators. Finally, we analyse the accuracy and consistency of the method, and show that in case of nested properties, the satisfaction function is characterized as a piecewise polynomial function.

4.1 Computing bounds for unnested path formulas

Regular time-bounded CSL model checking for an unnested path formula ϕ reduces to the computation of transient probabilities [4]. A similar reduction is also applicable to the computation of lower and upper bounds $\Lambda_{\phi,\min}$ and $\Lambda_{\phi,\max}$. In the following, we extend standard uniformisation to obtain safe bounds for a class of parametric rate functions.

Definition 9 (Parametric transient probabilities) Let $\mathcal{C}_\mathcal{P} = (S, \pi_0, \mathbf{R}, L)$ be a pCTMC over a parameter space $\mathcal{P}$. The vector of transient probabilities at time t

and for parameter valuation $p \in \mathcal{P}$ is approximated as follows

$$\hat{\pi}_{t,p} = \pi_0 \sum_{i=0}^{k_\epsilon} \gamma_{i,qt} \mathbf{P}_p^i = \sum_{i=0}^{k_\epsilon} \gamma_{i,qt} \tau_{i,p} \tag{8}$$

where π_0, $\gamma_{i,qt}$ and k_ϵ are as in Definition 4, $\tau_{i,p} = \pi_0 \mathbf{P}_p^i$ is the probability evolution in the discretized process, and $\mathbf{P}_p$ is the uniformised matrix obtained from $\mathbf{R}_p$.

We now show how to obtain safe approximations, $\hat{\pi}_t^{\min}$ and $\hat{\pi}_t^{\max}$, of $\hat{\pi}_{t,p}$, such that for all $s \in S$:

$$\begin{aligned} \hat{\pi}_t^{\min}(s) &\leq \min_{p \in \mathcal{P}} \hat{\pi}_{t,p}(s) \text{ and} \\ \hat{\pi}_t^{\max}(s) &\geq \max_{p \in \mathcal{P}} \hat{\pi}_{t,p}(s). \end{aligned} \tag{9}$$

The function $\hat{\pi}_t(s)$, which maps each parameter p to $\hat{\pi}_{t,p}(s)$, is a polynomial of degree $k_\epsilon d$, where d is the maximum degree of the elements of the parametric rate matrix $\mathbf{R}$. Thus, bounding the polynomial expression of $\hat{\pi}_t(s)$ is infeasible due to the large number of uniformisation steps, k_ϵ, and previous approaches have provided only an approximate solution by sampling the value of $\hat{\pi}_t$ over a grid in $\mathcal{P}$ [23].

We overcome this problem through a stepwise and statewise approximation. Specifically, for each uniformisation step i, we derive bounds $\tau_i^{\min}$ and $\tau_i^{\max}$, such that for all $s \in S$:

$$\tau_i^{\min}(s) \leq \min_{p \in \mathcal{P}} \tau_{i,p}(s) \quad \text{and} \quad \tau_i^{\max}(s) \geq \max_{p \in \mathcal{P}} \tau_{i,p}(s). \tag{10}$$

This allows robust approximations to the transient probabilities given by

$$\hat{\pi}_t^{\min} = \sum_{i=0}^{k_\epsilon} \gamma_{i,qt} \tau_i^{\min} \text{ and} \tag{11}$$

$$\hat{\pi}_t^{\max} = \sum_{i=0}^{k_\epsilon} \gamma_{i,qt} \tau_i^{\max} \tag{12}$$

which satisfy Eqn 9. For fixed $p \in \mathcal{P}$ and step i, the vector $\tau_{i,p}$ is given by

$$\tau_{i,p}(s) = \begin{cases} \tau_{i-1,p}(s) + \frac{1}{q} \cdot \text{flux}(\tau_{i-1,p}, s)(p) & \text{if } i > 0 \\ \pi_0(s) & \text{if } i = 0 \end{cases} \tag{13}$$

where q is the uniformisation constant and $\text{flux}(\tau, s)(p)$ is the net probability inflow of s in one step, starting from distribution τ. This is defined as:

$$\text{flux}(\tau, s)(p) = \sum_{s' \in S} \mathbf{R}_p(s', s) \cdot \tau(s') - \sum_{s' \in S} \mathbf{R}_p(s, s') \cdot \tau(s). \tag{14}$$

In the stepwise approximation, $\tau_i^{\min}$ and $\tau_i^{\max}$ are computed from $\tau_{i-1}^{\min}$ and $\tau_{i-1}^{\max}$, respectively, in such a way that:

$$\tau_i^{\min}(s) \leq \tau_{i-1}^{\min}(s) + \frac{1}{q} \cdot \min_{p \in \mathcal{P}} \text{flux}(\tau_{i-1}^{\min}, s)(p) \quad \text{and} \tag{15}$$

$$\tau_i^{\max}(s) \geq \tau_{i-1}^{\max}(s) + \frac{1}{q} \cdot \max_{p \in \mathcal{P}} \text{flux}(\tau_{i-1}^{\max}, s)(p). \tag{16}$$

The above inequalities imply Equation 10, since they establish coarser under- and over-approximations where the parameter valuation p is optimised locally, i.e. at each step and at each state.

It can be shown that the computation of $\tau_i^{\min}(s)$ and $\tau_i^{\max}(s)$ reduces to bounding the range of a polynomial of degree d over the parameters, where d is the maximum degree in $\mathbf{R}$. Henceforth, we restrict the class of allowed rate functions in order to compute $\tau_i^{\min}(s)$ and $\tau_i^{\max}(s)$. Specifically, we consider models where the entries of $\mathbf{R}$ are *multi-affine polynomials*, i.e. multivariate polynomials where each variable has degree at most 1. We remark that this class of models includes biochemical reaction networks with mass-action kinetics. Due to the following proposition, we can optimise the flux terms in a precise and efficient way, thus providing an effective method to compute $\tau_i^{\min}$ and $\tau_i^{\max}$.

Proposition 1 *Let $\mathcal{C}_{\mathcal{R}}$ be a pCTMC over a rectangular space $\mathcal{R}$, with state space S and parametric rate matrix $\mathbf{R}$. If the entries of $\mathbf{R}$ are multi-affine functions, then for any vector $\tau : S \to [0,1]$ and state $s \in S$,*

$$\min_{p\in\mathcal{R}} \mathrm{flux}(\tau,s)(p) = \min_{p\in V_{\mathcal{R}}} \mathrm{flux}(\tau,s)(p) \text{ and } \max_{p\in\mathcal{R}} \mathrm{flux}(\tau,s)(p) = \max_{p\in V_{\mathcal{R}}} \mathrm{flux}(\tau,s)(p) \tag{17}$$

where $V_{\mathcal{R}}$ is the set of vertices of $\mathcal{R}$ and flux *is as in Equation 14.*

Proof The expression $\mathrm{flux}(\tau,s)$ is a linear combination of the entries of $\mathbf{R}$, and thus is, in turn, a multi-affine function. By [7,40], the extrema of a multi-affine function defined over a rectangular domain $\mathcal{R}$ are found in the vertices of $\mathcal{R}$. □

Therefore, the bounds are computed as

$$\tau_i^{\min}(s) = \tau_{i-1}^{\min}(s) + \frac{1}{q} \cdot \min_{p\in V_{\mathcal{R}}} \mathrm{flux}(\tau_{i-1}^{\min}, s)(p) \quad \text{and} \tag{18}$$

$$\tau_i^{\max}(s) = \tau_{i-1}^{\max}(s) + \frac{1}{q} \cdot \max_{p\in V_{\mathcal{R}}} \mathrm{flux}(\tau_{i-1}^{\max}, s)(p) \tag{19}$$

which requires evaluating the flux terms only at the corner points of $\mathcal{R}$.

The above derivation describes forward computation of the probability bounds, i.e. the computation starts with an initial distribution at time 0 and the probability mass is propagated forward in time. Then, the bounds on the satisfaction function $\Lambda_{\phi,\min}(s)$ and $\Lambda_{\phi,\max}(s)$ are computed from $\hat{\pi}_t^{\min}$ and $\hat{\pi}_t^{\max}$, respectively, by setting $\pi_0(s) = 1$. When model checking a CSL formula, the computation of transient probabilities actually proceeds backwards [31]. For a target set $A \subseteq S$, parametric backward analysis computes a series of vectors $\sigma_i^{\min}$ and $\sigma_i^{\max}$ such that for all $s \in S$:

$$\sigma_i^{\min}(s) \le \min_{p\in\mathcal{P}} \sigma_{i,p}(s) \quad \text{and} \quad \sigma_i^{\max}(s) \ge \max_{p\in\mathcal{P}} \sigma_{i,p}(s) \tag{20}$$

where $\sigma_{i,p}(s)$ is the probability that, starting from the state s, a state in A is reached after i steps in the discretised process corresponding to $\mathcal{C}_p$. The computation of $\sigma_i^{\min}$ and $\sigma_i^{\max}$ exploits Proposition 1 and is analogous to that of the forward method. In this way, the vectors $\Lambda_{\phi,\min}, \Lambda_{\phi,\max}$ are obtained as:

$$\Lambda_{\phi,\min}(s) = \sum_{i=0}^{k_\epsilon} \gamma_{i,qt} \sigma_i^{\min}(s) \tag{21}$$

$$\Lambda_{\phi,\max}(s) = \sum_{i=0}^{k_\epsilon} \gamma_{i,qt} \sigma_i^{\max}(s) + e_{\text{f-g}} \tag{22}$$

for all $s \in S$, where the $e_{\text{f-g}}$ error is due to the truncation of the infinite summation in the discretised process, and can be controlled using the Fox and Glynn algorithm [19]. The set of target states A and time-horizon considered in the uniformisation procedure depend on the CSL formula. Note that the uniformised matrix $\mathbf{P}_p$ is modified according to the formula in a similar way to standard non-parametric CSL model checking [31].

4.2 Computing bounds for nested path formulas

To obtain $\Lambda_{\phi,\min}$ and $\Lambda_{\phi,\max}$ for an arbitrary path formula ϕ that contains nested state formulas, we have to correctly approximate the sets $\text{Sat}_\bot(\Phi)$ and $\text{Sat}_\top(\Phi)$ for each sub-formula $\Phi = \text{P}_{\sim r}[\phi]$. The approximated sets, denoted as $\overline{\text{Sat}_\bot}(\Phi)$ and $\overline{\text{Sat}_\top}(\Phi)$ are defined as

$$\overline{\text{Sat}_\bot}(\Phi) = \{s \in S \mid s \mathrel{\overline{\models_\bot}} \Phi\} \text{ and } \overline{\text{Sat}_\top}(\Phi) = \{s \in S \mid s \mathrel{\overline{\models_\top}} \Phi\} \tag{23}$$

where $\overline{\models_\bot}$ and $\overline{\models_\top}$ approximate the satisfaction relations $\models_\bot$ and $\models_\top$ (see Definition 7), respectively. For a pCTMC $\mathcal{C}_\mathcal{P}$, their semantics is defined as follows:

$$s \mathrel{\overline{\models_\top}} \text{P}_{\sim r}[\phi] \Leftrightarrow \exists p \in \mathcal{P}.\ Pr(\omega \in \text{Path}(s) \mid \omega \mathrel{\overline{\models_\top}} \phi) \sim r \text{ in } \mathcal{C}_\mathcal{P}$$
$$s \mathrel{\overline{\models_\bot}} \text{P}_{\sim r}[\phi] \Leftrightarrow \forall p \in \mathcal{P}.\ Pr(\omega \in \text{Path}(s) \mid \omega \mathrel{\overline{\models_\bot}} \phi) \sim r \text{ in } \mathcal{C}_\mathcal{P}$$

where the path formula ϕ is expanded for $+ \in \{\top, \bot\}$ as

$$\omega \mathrel{\overline{\models_+}} \mathtt{X}\Phi \quad \Leftrightarrow \omega(1) \text{ exists and } \omega(1) \mathrel{\overline{\models_+}} \Phi$$
$$\omega \mathrel{\overline{\models_+}} \Phi_1 \text{U}^I\ \Phi_2 \quad \Leftrightarrow \exists t \in I. \text{ such that } [\omega@t \mathrel{\overline{\models_+}} \Phi_2 \wedge (\forall r \in [0,t).\ \omega(r) \mathrel{\overline{\models_+}} \Phi_1)].$$

The semantics of the other state formulas is the same as in Definition 7.

The CSL model checking for pCTMCs proceeds through a bottom-up procedure that computes the sets $\overline{\text{Sat}_\bot}(\Phi)$ and $\overline{\text{Sat}_\top}(\Phi)$ by iteratively replacing the innermost $\text{P}_{\sim r}[\phi]$ operators with the corresponding sets of satisfying states. When ϕ is non-nested, these sets are obtained from the safe bounds of the corresponding satisfaction function Λ_ϕ (computed as per Section 4.1) as follows:

$$s \mathrel{\overline{\models_\bot}} \text{P}_{\sim r}[\phi] \Leftrightarrow \begin{cases} \Lambda_{\phi,\min}(s) \sim r & \text{if } \sim \in \{\geq, >\} \\ \Lambda_{\phi,\max}(s) \sim r & \text{if } \sim \in \{\leq, <\} \end{cases} \tag{24}$$

$$s \mathrel{\overline{\models_\top}} \text{P}_{\sim r}[\phi] \Leftrightarrow \begin{cases} \Lambda_{\phi,\max}(s) \sim r & \text{if } \sim \in \{\geq, >\} \\ \Lambda_{\phi,\min}(s) \sim r & \text{if } \sim \in \{\leq, <\}. \end{cases} \tag{25}$$

The approximations $\overline{\models_\bot}$ and $\overline{\models_\top}$ propagate the bounds on the satisfaction function inductively on the structure of the CSL formula, such that:

$$\overline{\mathrm{Sat}_\bot}(\varPhi) \subseteq \mathrm{Sat}_\bot(\varPhi) \text{ and } \overline{\mathrm{Sat}_\top}(\varPhi) \supseteq \mathrm{Sat}_\top(\varPhi). \tag{26}$$

Correctness follows from the expansion of the satisfaction relations, which we demonstrate for the $\mathrm{P}_{\geq r}$ operator only. The left-hand side of Eqn. 26 follows for $\varPhi = \mathrm{P}_{\geq r}[\phi]$ through:

$$\begin{aligned} s \overline{\models_\bot} \mathrm{P}_{\geq r}[\phi] &\Rightarrow \varLambda_{\phi,\min}(s) \geq r \Rightarrow \forall p \in \mathcal{P}.\ \varLambda_\phi(p)(s) \geq r \\ &\Rightarrow \forall p \in \mathcal{P}.\ Pr(\omega \in \mathrm{Path}(s) \text{ in } \mathcal{C}_p \mid \omega \models \phi) \geq r \Rightarrow s \models_\bot \mathrm{P}_{\geq r}[\phi]. \end{aligned}$$

The right-hand side follows from:

$$s \models_\top \mathrm{P}_{\geq r}[\phi] \Rightarrow \exists p \in \mathcal{P}.\ \varLambda_\phi(p)(s) \geq r \Rightarrow \varLambda_{\phi,\max}(s) \geq r \Rightarrow s \overline{\models_\top} \mathrm{P}_{\geq r}[\phi].$$

An example of synthesis for nested formulas is illustrated in Section 6.1.1.

Complexity. For $\mathcal{C}_\mathcal{R} = (S, \pi_0, \mathbf{R}, L)$, time-bounded path formula ϕ and fixed n-dimensional rectangular space $\mathcal{R}$, the time complexity of the procedure for computing the probability bounds is $\mathcal{O}(t_{CSL} \cdot t_{pCSL})$. The factor $t_{CSL} = |\phi| \cdot M \cdot q \cdot t_{\max}$ is the worst-case time complexity of time-bounded CSL model checking (see [4]), where $|\phi|$ is the number of time-bounded path sub-formulas in ϕ, M is the number of non-zero elements in the rate matrix, $t_{\max}$ is the highest time bound occurring in ϕ and q is the uniformisation rate. The factor t_{pCSL} is due to the parametric analysis. Following Proposition 1 and Eqn 18, for general multi-affine rate functions the bounds $\tau_i^{\min}$ and $\tau_i^{\max}$ are obtained by performing $2 \cdot 2^n$ evaluations of the vector $\tau_{i,p}$ (there are 2^n corner points in $\mathcal{R}$), at each uniformisation step i. Thus, $t_{pCSL} = 2^{n+1}$. On the other hand, for linear rate functions $t_{pCSL} = \mathcal{O}(n)$, as shown in [11].

4.3 Computing bounds for reward operators

The standard model checking algorithm for reward operators is based on the uniformisation procedure [31]. To obtain the sets $\overline{\mathrm{Sat}_\top}(\varPhi)$ and $\overline{\mathrm{Sat}_\bot}(\varPhi)$ for the reward operators, we have to compute for $X \in \{X_{\mathrm{C}^{\leq t}}, X_{\mathrm{I}^{=t}}\}$ bounds $Exp_{\min}(s, X)$ and $Exp_{\max}(s, X)$ on the expected rewards such that:

$$Exp_{\min}(s, X) \leq \inf_{p \in \mathcal{P}} Exp(s, X) \text{ in } \mathcal{C}_p \tag{27}$$

$$Exp_{\max}(s, X) \geq \sup_{p \in \mathcal{P}} Exp(s, X) \text{ in } \mathcal{C}_p. \tag{28}$$

The quantities can be obtained using the forward computation where the initial distribution is defined as $\pi_0(s) = 1$. For a reward structure $(\rho, \boldsymbol{\iota})$, the instantaneous reward is computed as:

$$Exp(s, X_{\mathrm{I}^{=t}}) = \sum_{s' \in S} \rho(s')\pi_t(s'). \tag{29}$$

To find the cumulative reward, the state-transition rewards $\boldsymbol{\iota}$ are additionally taken into account as follows [30]:

$$\begin{aligned} Exp(s, X_{\mathrm{C}^{\leq t}}) &= \sum_{s' \in S} \int_0^t \left(\rho(s')\pi_u(s') + \sum_{s'' \in S} \mathbf{R}(s', s'')\boldsymbol{\iota}(s', s'')\pi_u(s') \right) du &(30) \\ &= \sum_{s' \in S} \int_0^t \left(\rho(s') + \sum_{s'' \in S} \mathbf{R}(s', s'')\boldsymbol{\iota}(s', s'') \right) \pi_u(s')du &(31) \\ &= \sum_{s' \in S} \left(\rho(s') + \sum_{s'' \in S} \mathbf{R}(s', s'')\boldsymbol{\iota}(s', s'') \right) \int_0^t \pi_u(s')du. &(32) \end{aligned}$$

where $\int_0^t \pi_u(s')du$ is the expected amount of time the Markov process spends in state s' up until time t. Following the parametric uniformisation of Section 4.1, safe bounds for the reward operators are found as:

$$\begin{aligned} Exp_{\min}(s, X_{I^{=t}}) &= \sum_{s' \in S} \rho(s')\hat{\pi}_t^{\min}(s') &(33) \\ Exp_{\max}(s, X_{I^{=t}}) &= \sum_{s' \in S} \rho(s') \left(\hat{\pi}_t^{\max}(s') + e_{\text{f-g}} \right) &(34) \\ Exp_{\min}(s, X_{C^{\leq t}}) &= \sum_{s' \in S} \left(\mathrm{rew}(\rho, \boldsymbol{\iota}, s') \frac{1}{q} \sum_{i=0}^{\infty} \overline{\gamma}_{i,qt} \cdot \tau_i^{\min}(s') \right) &(35) \\ Exp_{\max}(s, X_{C^{\leq t}}) &= \sum_{s' \in S} \left(\mathrm{rew}(\rho, \boldsymbol{\iota}, s') \frac{1}{q} \sum_{i=0}^{\infty} \overline{\gamma}_{i,qt} \cdot \tau_i^{\max}(s') \right) &(36) \end{aligned}$$

where the mixed Poisson probabilities and the combined rewards are

$$\begin{aligned} \overline{\gamma}_{i,qt} &= 1 - \sum_{j=i}^{\infty} \gamma_{j,qt} &(37) \\ \mathrm{rew}(\rho, \iota, s) &= \rho(s) + \sum_{s' \in S} \mathbf{R}(s, s')\boldsymbol{\iota}(s, s'). &(38) \end{aligned}$$

The bounds for the cumulative rewards (Equations 35 and 36) are understood as follows: $\overline{\gamma}_{i,qt}$ is the probability to see at least i jumps in the discretised process, which is multiplied by the under- or over-approximation of the probability to be in state s. So $\sum_{i=0}^{\infty} \overline{\gamma}_{i,qt} \cdot \tau_i^{\min}(s)$ is an under-approximation of the number of epochs the discretised process spends in state s. Observe that $\frac{1}{q}$ is the expected time until a jump occurs and $\mathrm{rew}(\rho, \boldsymbol{\iota}, s)$ is the expected reward obtained per time unit spent in s. As discussed in [31], the infinite sums can be approximated using methods based on Fox and Glynn [19].

Note that, also for rewards, backward computation allows obtaining safe bounds for all states $s \in S$, using the vectors $\sigma_i^{\min}$ and $\sigma_i^{\max}$.

4.4 Analysis of satisfaction function and approximation error

When computing bounds $\tau_i^{\min}$ and $\tau_i^{\max}$ on the transient probabilities, an approximation error occurs because the values are obtained by optimizing $\tau_{i,p}$ locally, i.e.

at each step and at each state, and this error accumulates at each uniformisation step. We examine this error for the multi-affine case where Proposition 1 applies. For a fixed state s, let the maximizing argument of the transient probability be (cf. Eqn 8):

$$p^* = \arg\max_{p \in \mathcal{R}} \pi_t(s) \tag{39}$$

Then, the optimal probabilities at step i, τ_i^*, are defined by

$$\tau_i^* = \pi_0 \mathbf{P}_{p^*}^i. \tag{40}$$

For state s, the global error after i uniformisation steps corresponds to the difference between the maximum probability in s and its over-approximation:

$$g_i(s) = |\tau_i^*(s) - \tau_i^{\max}(s)|. \tag{41}$$

This error depends linearly on the size of the parameter space and exponentially on the number of uniformisation steps, which we summarize as follows.

Proposition 2 *Let $\mathcal{C}_\mathcal{R} = (S, \pi_0, \mathbf{R}, L)$ be a pCTMC with multi-affine rates on an n-dimensional rectangular space $\mathcal{R}$, ϕ be an unnested time-bounded CSL path formula and $g_i(s)$ be the global approximation error for the maximum probability of being in state s after i uniformisation steps. Then, there exist $M_1, M_2 < \infty$ such that, for any $s \in S$ and step $i > 0$, an upper bound to the error, $\overline{g}_i \geq \max_{s \in S} g_i(s)$, is given as*

$$\overline{g}_i = \begin{cases} 0 & \textit{if } i = 0 \\ \overline{g}_{i-1} \cdot \left(1 + \frac{M_2}{q}\right) + \frac{M_1}{q} w_\mathcal{R} & \textit{if } i > 0, \end{cases} \tag{42}$$

where $w_\mathcal{R} = \max_{j=1,\ldots,n}(x_j^\top - x_j^\bot)$ is the width of $\mathcal{R}$.

Proof See Appendix A.1.

Let $\hat{\Lambda}_\phi(.)(s_0)$ be the approximation of the satisfaction function $\Lambda_\phi(.)(s_0)$ for initial state s_0 obtained using standard transient analysis and uniformisation [31]. We now provide an important characterization of $\hat{\Lambda}_\phi(.)(s_0)$, which holds for pCTMCs with general polynomial rates.

Theorem 1 *For a pCTMC $\mathcal{C}_\mathcal{P}$ on a bounded parameter space $\mathcal{P}$, an initial state s_0 and a finitely-nested and time-bounded CSL path formula ϕ, the approximate satisfaction function $\hat{\Lambda}_\phi(.)(s_0)$ is piecewise polynomial in $\mathcal{P}$ over a finite number of subdomains.*

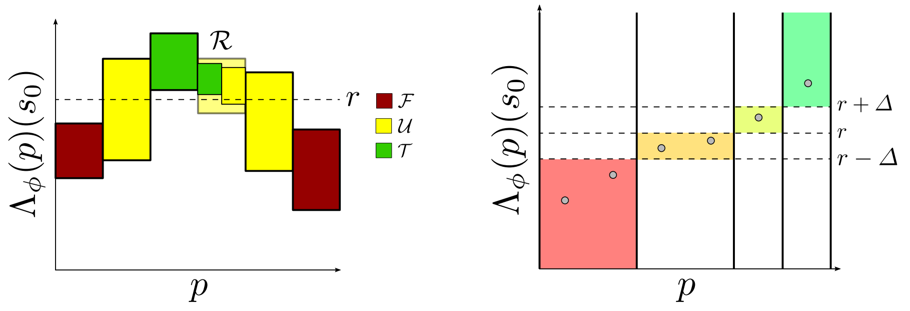
Proof See Appendix A.2.

5 Refinement-based Parameter Synthesis

We present algorithms to solve Problems 1 and 2, utilising the approximation of probability bounds introduced in Section 4. The algorithms iteratively refine the parameter space $\mathcal{P}$ and compute the probability bounds on the satisfaction function for each subspace until a required accuracy is obtained.

Algorithm 1 Threshold Synthesis

```
Require: pCTMC $\mathcal{C}_\mathcal{P}$ over parameter space $\mathcal{P}$, initial state $s_0$, CSL path formula $\phi$, threshold $\geq r$ and volume tolerance $\varepsilon > 0$
Ensure: $\mathcal{T}$, $\mathcal{U}$ and $\mathcal{F}$ as in Problem 1
 1: $\mathcal{T} \leftarrow \emptyset$, $\mathcal{F} \leftarrow \emptyset$, $\mathcal{U} \leftarrow \mathcal{P}$
 2: repeat
 3:     $D \leftarrow \mathsf{decompose}(\mathcal{U})$, $\mathcal{U} \leftarrow \emptyset$
 4:     for each $\mathcal{R} \in D$ do
 5:         $(\Lambda^{\mathcal{R}}_{\min}, \Lambda^{\mathcal{R}}_{\max}) \leftarrow \mathsf{computeBounds}(\mathcal{C}_\mathcal{R}, s_0, \phi)$
 6:         if $\Lambda^{\mathcal{R}}_{\min} \geq r$ then
 7:             $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{R}$
 8:         else if $\Lambda^{\mathcal{R}}_{\max} < r$ then
 9:             $\mathcal{F} \leftarrow \mathcal{F} \cup \mathcal{R}$
10:         else
11:             $\mathcal{U} \leftarrow \mathcal{U} \cup \mathcal{R}$
12: until $\mathrm{vol}(\mathcal{U})/\mathrm{vol}(\mathcal{P}) > \varepsilon$        ▷ where $\mathrm{vol}(A) = \int_A 1 d\mu$
```



(a) (b)

Fig. 3 (a) Refinement in threshold synthesis with $\geq r$. Parameter values are on the x-axis, probabilities on the y-axis. Each box describes a parameter region (width), and its probability bounds (height). The refinement of $\mathcal{R}$ yields regions in $\mathcal{T}$ and in $\mathcal{U}$. (b) Initial sampling-guided refinement of $\mathcal{P}$. Sampled probabilities and a tolerance Δ are used to identify regions that are *likely* to be in $\mathcal{T}$ (green area, samples $\geq r + \Delta$), in $\mathcal{F}$ (red, $\leq r - \Delta$), or close to the threshold r (orange and lime green, $\in (r - \Delta, r + \Delta)$).

5.1 Threshold Synthesis

Algorithm 1 describes the method to solve the threshold synthesis problem with input formula ϕ and threshold $\geq r$. The idea, also illustrated in Figure 3(a), is to iteratively refine the undecided parameter subspace $\mathcal{U}$ (line 3) until the termination condition is met (line 14). At each step, we obtain a partition D of $\mathcal{U}$. For each subspace $\mathcal{R} \in D$, the algorithm computes bounds $\Lambda^{\mathcal{R}}_{\min}$ and $\Lambda^{\mathcal{R}}_{\max}$ on the minimal and maximal probability that $\mathcal{C}_\mathcal{R}$ with the initial state s_0 satisfies ϕ (line 5). We then evaluate if $\Lambda^{\mathcal{R}}_{\min}$ is above the threshold r, in which case the satisfaction of the threshold is guaranteed for the whole region $\mathcal{R}$, which is then added to $\mathcal{T}$. Otherwise, the algorithm tests whether $\mathcal{R}$ can be added to the set $\mathcal{F}$ by checking if $\Lambda^{\mathcal{R}}_{\max}$ is below the threshold r. If $\mathcal{R}$ is neither in $\mathcal{T}$ nor in $\mathcal{F}$, it forms an undecided subspace that is added to the set $\mathcal{U}$. The algorithm terminates when the volume of the undecided subspace is negligible with respect to the volume of the entire parameter space, i.e. $\mathrm{vol}(\mathcal{U})/\mathrm{vol}(\mathcal{P}) \leq \varepsilon$, where ε is the input tolerance. Otherwise, the algorithm continues to the next iteration, where $\mathcal{U}$ is further refined.

5.1.1 Correctness and termination

Correctness of the algorithm follows from the construction of the regions $\mathcal{T}$ (lines 6,7) and $\mathcal{F}$ (lines 8,9). The termination condition guarantees the required bound of the relative volume of $\mathcal{U}$ (line 12). Termination of the algorithm for (possibly nested) CSL properties is stated below.

Proposition 3 *For a pCTMC $\mathcal{C}_{\mathcal{P}}$ over a parameter space $\mathcal{P}$, initial state s_0, CSL path formula ϕ and volume tolerance ε, Algorithm 1 terminates.*

Proof See Appendix A.3.

5.1.2 Initial decomposition

Optionally, a heuristic based on an initial decomposition precedes the refinement procedure. The initial decomposition can speed up the refinement, since it decomposes the parameter space $\mathcal{P}$ in advance. It is guided by a priori uniform sampling of probability values. In particular, we sample points $p_1, p_2, \ldots, p_n \in \mathcal{P}$ and compute $\hat{\Lambda}_\phi(p_i)(s_0)$ for $i = 1, \ldots, n$ using standard CSL model checking. Then, we partition $\mathcal{P}$ into subspaces that set apart samples where $\hat{\Lambda}_\phi(p_i)(s_0) \geq r$ from those where $\hat{\Lambda}_\phi(p_i)(s_0) < r$. As depicted in Figure 3(b), we also use a tolerance $\Delta > 0$ to identify regions close to the threshold that are more likely to be further decomposed. In this case the initial decomposition returns four regions. Our experiments demonstrates that, in some cases, depending on the shape of the satisfaction function and the threshold r, the initial decomposition accelerates the synthesis.

5.2 Max Synthesis

Algorithm 2 is used to solve the max synthesis problem, which returns the set $\mathcal{T}$ containing the parameter valuations that maximize the probability of the path formula ϕ and the set $\mathcal{F}$ not yielding the maximum value. Starting from $\mathcal{T} = \mathcal{P}$, the algorithm iteratively refines $\mathcal{T}$ until the probability tolerance condition at Problem 2 is met (line 14).

Let D be a partition of $\mathcal{T}$. For each subspace $\mathcal{R} \in D$, the algorithm computes bounds $\Lambda^{\mathcal{R}}_{\min}$ and $\Lambda^{\mathcal{R}}_{\max}$ on the minimal and maximal probability that $\mathcal{C}_{\mathcal{R}}$ with the initial state s_0 satisfies ϕ (line 5). The algorithm then rules out subspaces that are guaranteed to be included in $\mathcal{F}$, by deriving an under-approximation (M) to the maximum satisfaction probability (line 7). If $\Lambda^{\mathcal{R}}_{\max}$ is below the under-approximation, the subspace $\mathcal{R}$ can be safely added to the set $\mathcal{F}$ (line 9). Otherwise, it is kept in $\mathcal{T}$.

We consider two approaches for deriving the bound M, namely a naive approach and a sampling-based approach. In the naive method, we set M to the maximum over the least bounds in the partition of $\mathcal{T}$, that is, $M = \max\{\Lambda^{\mathcal{R}'}_{\min} \mid \mathcal{R}' \in D\}$. Let $\overline{\mathcal{R}}$ be the region with highest lower bound. The sampling-based method, illustrated in Algorithm 3, improves on this by sampling a set of parameters $\{p_1, p_2, \ldots\} \subseteq \overline{\mathcal{R}}$ (line 2) and taking the highest value of $\hat{\Lambda}_\phi(p)(s_0)$, that is, $M = \max\left\{\hat{\Lambda}_\phi(p_i)(s_0) \mid p_i \in \{p_1, p_2, \ldots\}\right\}$ (line 3). Each $\hat{\Lambda}_\phi(p)(s_0)$ is computed

Algorithm 2 Max Synthesis

```
Require: pCTMC $\mathcal{C}_\mathcal{P}$ over parameter space $\mathcal{P}$, initial state $s_0$, CSL path formula $\phi$ and
    probability tolerance $\epsilon > 0$
Ensure: $\Lambda^\perp_\phi$, $\Lambda^\top_\phi$, $\mathcal{T}$ and $\mathcal{F}$ as in Problem 2
$\mathcal{F} \leftarrow \emptyset$, $\mathcal{T} \leftarrow \mathcal{P}$
repeat
    $D \leftarrow \mathsf{decompose}(\mathcal{T})$, $\mathcal{T} \leftarrow \emptyset$, $\Lambda^\perp_\phi \leftarrow +\infty$, $\Lambda^\top_\phi \leftarrow -\infty$
    for each $\mathcal{R} \in D$ do
        $(\Lambda^\mathcal{R}_{\min}, \Lambda^\mathcal{R}_{\max}) \leftarrow \mathsf{computeBounds}(\mathcal{C}_\mathcal{R}, s_0, \phi)$
    $M \leftarrow \mathsf{getMaximalLowerBound}(D)$
    for each $\mathcal{R} \in D$ do
        if $\Lambda^\mathcal{R}_{\max} < M$ then
            $\mathcal{F} \leftarrow \mathcal{F} \cup \mathcal{R}$
        else
            $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{R}$
            $\Lambda^\perp_\phi \leftarrow \min\{\Lambda^\perp_\phi, \Lambda^\mathcal{R}_{\min}\}$
            $\Lambda^\top_\phi \leftarrow \max\{\Lambda^\top_\phi, \Lambda^\mathcal{R}_{\max}\}$
until $\Lambda^\top_\phi - \Lambda^\perp_\phi > \epsilon$
```

Algorithm 3 Sampling-guided computation of a maximal lower bound

```
Require: Parameter decomposition $D$ and number of samples $n$
Ensure: $\overline{M}$, an improved lower bound for max probability in $D$
$\overline{\mathcal{R}} = \arg\max_{\mathcal{R}' \in D} \Lambda^{\mathcal{R}'}_{\min}$
$(p_1, \dots, p_n) \leftarrow \mathsf{Uniform}(\overline{\mathcal{R}}, n)$
$\overline{M} \leftarrow \max_{p_i} \hat{\Lambda}(p_i)(s_0)$
```

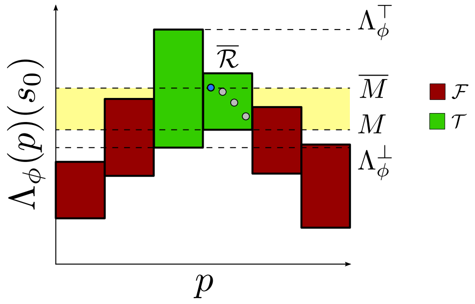


Fig. 4 Refinement in max synthesis. The two outermost regions (in red) cannot contain the maximum, as their upper bound is below the maximum lower bound (M) found at region $\overline{\mathcal{R}}$. The maximum lower bound is improved by sampling several points $p \in \overline{\mathcal{R}}$ and taking the highest value ($\overline{M}$) of the satisfaction function $\hat{\Lambda}_\phi(p)(s_0)$. The yellow area highlights the improvement.

through regular CSL model checking, and is equally expensive as computing the bounds on a fixed pCTMC. The sampling method results in an improved under-approximation to the maximum of the satisfaction function. As a result, the bound rules out more regions, and fewer refinements are required in the next iteration (see Figure 4).

5.2.1 Correctness and termination

The correctness of the algorithm derives from the construction of the sets $\mathcal{T}$ (lines 12,13) and $\mathcal{F}$ (lines 8,9), and from the termination condition (line 14).

We remark that, for nested properties, the satisfaction function is in general discontinuous, which allows $\Lambda^{\mathcal{R}}_{\max} - \Lambda^{\mathcal{R}}_{\min} > \epsilon$ when $\mathcal{T}$ contains a jump discontinuity. This prevents the algorithm from terminating. For this reason a volume-based stopping criterion, such as $\mathrm{vol}(\mathcal{T}) \leq \epsilon$ for $\epsilon > 0$, which replaces the condition on line 14, should be used when analysing nested properties. Indeed, the volume of any region containing such a discontinuity can be made arbitrarily small in a finite number of refinement steps, as discussed in the proof of Proposition 3. With unnested properties, the following proposition ensures termination.

Proposition 4 *For a pCTMC $\mathcal{C}_{\mathcal{P}}$ over a parameter space $\mathcal{P}$, initial state s_0, non-nested CSL path formula ϕ and tolerance ϵ, Algorithm 2 terminates.*

Proof See Appendix A.4.

5.3 Complexity

The time complexity of the procedure computing the probability bounds for a fixed region has been discussed in Section 4.2. The overall runtime of both algorithms further depends on the number of subspaces that are required to obtain the desired precision. This number scales exponentially in the number of parameters and linearly in the volume of the parameter space. However, in practice, the number of required subspaces also depends on the shape of the satisfaction function and the type of synthesis.

6 Results

We implemented the synthesis algorithms on top of the tool PRISM 4.0 [32]. Currently, a prototype command-line version is available at `https://github.com/Palmik/prism-pse/`. Models and properties are specified using the native specification languages of PRISM. Note that the online version of the tool only supports linear rate functions and non-nested formulas.

We demonstrate the applicability and efficiency of the developed algorithms on three case studies. We run all experiments on a Linux workstation with an AMD Phenom$^{\text{TM}}$ II X4 940 Processor @ 3GHz, 8 GB DDR2 @ 1066 MHz RAM.

6.1 Epidemic model

The SIR model [27] describes the epidemic dynamics in a well-mixed and closed population of susceptible (S), infected (I) and recovered (R) individuals. In the model, a susceptible individual is infected after a contact with an infected individual with rate k_i. Infected individuals recover with rate k_r, after which they are immune to the infection. We can describe this process with the following biochemical reaction model with mass action kinetics (i.e. the rate functions are linear with respect to the parameters):

$$S + I \xrightarrow{k_i} I + I \qquad \text{(Infection)}$$

$$I \xrightarrow{k_r} R \qquad \text{(Recovery)}$$

We represent the model as a pCTMC with parameters $k_i \in [0.005, 0.3]$ and $k_r \in [0.005, 0.2]$, and initial populations $S = 95$, $I = 5$, $R = 0$.

We consider the time-bounded CSL path formula $\phi = (I > 0)\mathrm{U}^{[100,120]}(I = 0)$, specifying behaviour where the infection lasts for at least 100 time units, and dies out before 120 time units. Property and parameters are taken from [9], where the authors estimate the satisfaction function for ϕ following a Bayesian approach[1].

First, we perform threshold synthesis experiments to find infection and recovery rates for which ϕ is satisfied with probability at least $r = 10\%$. Figure 5 illustrates the solutions for one-dimensional parameter spaces (plots a,b) obtained by fixing $k_i = 0.12$ and $k_r = 0.05$, respectively; and for the two-dimensional parameter space (plots c,d). Results evidence that a significantly higher number of refinement steps is required for parameter subspaces where the satisfaction function Λ is close to the probability threshold r.

Second, we perform max and min synthesis experiments for property ϕ over one-dimensional and two-dimensional parameter spaces. Results are summarized in Figure 6. For the experiments in Figure 6b and 6c we observe that, in order to meet the desired probability tolerance, a high number of refinement steps is required due to two local extrema close to the minimizing region and a bell-shaped Λ with the maximizing region at the top, respectively.

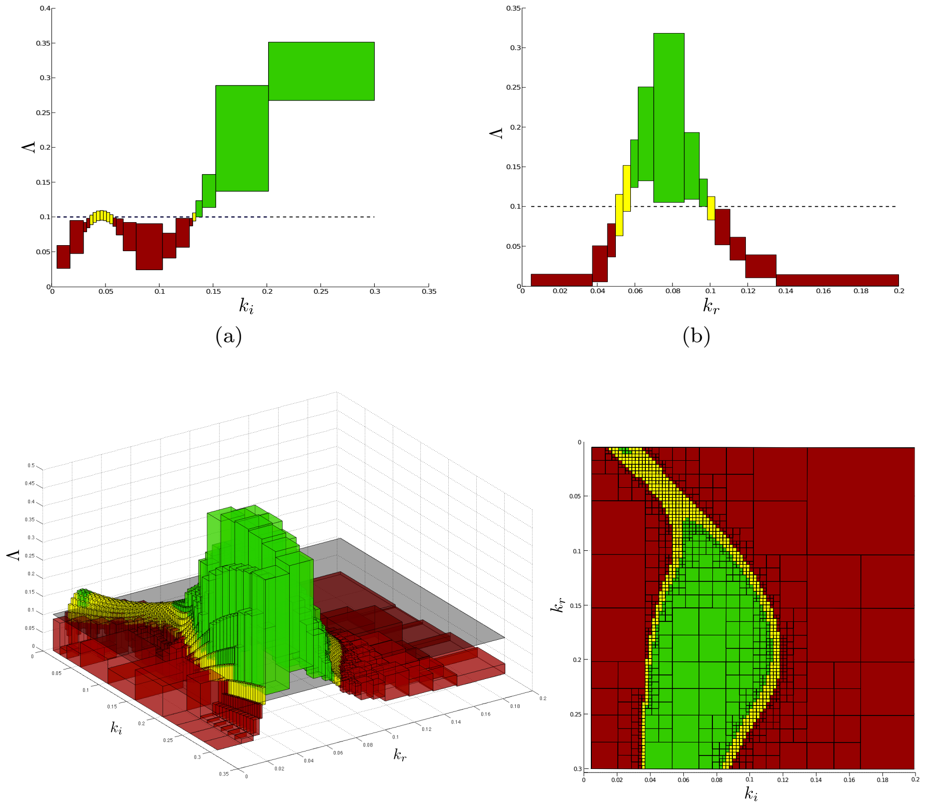
Our precise results for the problem in Fig. 6a improve on the estimation in [9], where in a similar experiment the maximal satisfaction probability is imprecisely registered at $k_i = 0.25$.

We also compare the solutions to the max synthesis problem over the two-dimensional parameter space obtained by applying Algorithm 2 with sampling (Figure 6e) and without (Figure 6f). In the former case, a more precise $\mathcal{T}$ region is obtained (with volume 2.04 times smaller than in the approach without sampling), hence giving a more accurate approximation of the max probability. Sampling also allows us to rule out earlier those parameter regions that are outside the final solution, thus avoiding unnecessary decompositions and improving the runtime (1.72 times faster than in the approach without sampling). This is visible by the coarser approximations of probabilities in the $\mathcal{F}$ region.

6.1.1 Parameter synthesis for nested CSL formulas

We use the SIR model to illustrate parameter synthesis for a nested CSL formula. We consider initial populations of $S = 3$, $I = 1$ and $R = 0$ to obtain a small model with only 14 reachable states, which allows us visualise the main steps of the synthesis algorithm. We modify the original path formula ϕ to specify behaviour where the infection lasts at least 100 time units and, before 200 time units, the system reaches a state where the infection becomes extinct before time 100 with probability higher than 90%. Such a property can be formalised as the nested CSL path formula $\phi' = (I > 0)\mathrm{U}^{[100,120]}(\Psi)$, where $\Psi = \mathrm{P}_{>0.9}[F^{[0,100]}(I = 0)]$.

[1] In [9], a linear-time specification equivalent to ϕ is given.



(a)

(b)

(c)

(d)

	$k_i \times k_r$	Runtime	Subspaces
a)	$[0.005, 0.3] \times 0.05$	42.2 s	23
b)	$0.12 \times [0.005, 0.2]$	26.7 s	15
c,d)	$[0.005, 0.3] \times [0.005, 0.2]$	29.3 min	1320

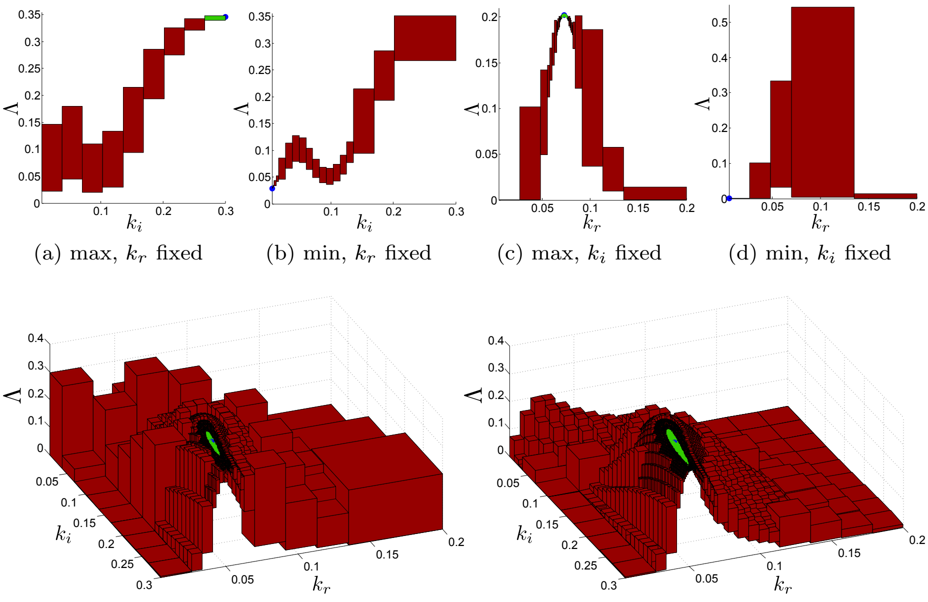
Fig. 5 Solution to threshold synthesis problems for the SIR model and the property $\mathrm{P}_{\geq 0.1}[(I > 0)\mathrm{U}^{[100,120]}(I = 0)]$. Plots c) and d) depict the same result with two different angles. Runtime and number of subspaces in the final partition of $\mathcal{P}$ are listed. Volume tolerance is $\varepsilon = 10\%$. Colour code is as in Fig. 3 a).

We show threshold synthesis for property $\mathrm{P}_{>0.1}[\phi']$ and parameter space $k_i \times k_r = 0.12 \times [0.02, 0.04]$. Recall that the probability bounds $\Lambda_{\phi',\min}$ and $\Lambda_{\phi',\max}$ for a nested formula ϕ' are computed as:

$$\Lambda_{\phi',\min}(s) = \Lambda_{\phi^{\perp},\min}(s) \text{ where } \phi^{\perp} = (I > 0)\mathrm{U}^{[100,120]}(\mathrm{Sat}_{\perp}(\Psi))$$

$$\Lambda_{\phi',\max}(s) = \Lambda_{\phi^{\top},\min}(s) \text{ where } \phi^{\top} = (I > 0)\mathrm{U}^{[100,120]}(\mathrm{Sat}_{\top}(\Psi))$$

Figure 7 depicts four steps of the refinement algorithm. In the first step, $\mathrm{Sat}_{\perp}(\Psi)$ contains only states where the population of I is 0 and $\mathrm{Sat}_{\top}(\Psi)$ contains all reachable states. The threshold synthesis algorithm for $\mathrm{P}_{>0.1}[\phi']$ partitions the parameter space into a single undecided region. In the second step, the parameter space refinement yields a refinement of the satisfaction sets. In particular, for $k_r = [0.02, 0.03]$, we observe that some states no longer appear in



(a) max, k_r fixed (b) min, k_r fixed (c) max, k_i fixed (d) min, k_i fixed

(e) max, sampling-based (f) max, no-sampling

	$k_i \times k_r$	Runtime	Subspaces	$\Lambda_\phi^\perp$	$\Lambda_\phi^\top$	$\mathcal{T}$
a)	$[0.005, 0.3] \times 0.05$	16.5 s	9	33.94%	34.85%	$[0.267, 0.3] \times 0.05$
b)	$[0.005, 0.3] \times 0.05$	49.5 s	21	2.83%	2.91%	$[0.005, 0.0054] \times 0.05$
c)	$0.12 \times [0.005, 0.2]$	99.7 s	57	19.94%	20.42%	$0.12 \times [0.071, 0.076]$
d)	$0.12 \times [0.005, 0.2]$	10.4 s	5	†	0.005%	$0.12 \times [0.005, 0.026]$
e)	$[0.005, 0.3] \times [0.005, 0.2]$	3.6 h	5817	35.01%	35.72%	$[0.217, 0.272] \times [0.053, 0.059]$
f)	$[0.005, 0.3] \times [0.005, 0.2]$	6.2 h	10249	34.77%	35.72%	$[0.209, 0.29] \times [0.051, 0.061]$

Fig. 6 Solution to max (a,c,e,f) and min (b,d) synthesis for the SIR model and the formula $\phi = (I > 0)\mathrm{U}^{[100,120]}(I = 0)$. Sampling-based refinement is used for all experiments but e). Colour code is as in Fig. 4. In the table, we report runtime, number of subspaces, approximation ($\Lambda_\phi^\perp$ and $\Lambda_\phi^\top$) of min or max probability, and bounding box of $\mathcal{T}$. Probability tolerance is $\epsilon = 1\%$ for max synthesis and $\epsilon = 0.1\%$ for min synthesis. †: The found value, 3.05×10^{-10}, is of the same order of magnitude as the precision used during uniformisation.

$\mathrm{Sat}_\top(\Psi)$ (shown red-coloured), while, for $k_r = [0.03, 0.04]$, some states are added to $\mathrm{Sat}_\perp(\Psi)$ (shown green-coloured). Note that, although $\mathrm{Sat}_\perp(\Psi) \neq \mathrm{Sat}_\top(\Psi)$, the approximation is precise enough to decide that the latter parameter subspace is false, since $\Lambda_{\phi',\max}(s) < 0.1$. In the third refinement step, we manage to refine only $\mathrm{Sat}_\top(\Psi)$ for region $k_r = [0.02, 0.025]$. The region remains still undecided as well as region $k_r = [0.025, 0.03]$. Finally, in the fourth step, the satisfaction sets for $k_r = [0.02, 0.0225]$ collapse, i.e. $\mathrm{Sat}_\perp(\Psi) = \mathrm{Sat}_\top(\Psi)$, which allows us to conclude that the region is true. Region $k_r = [0.0275, 0.03]$ can also be decided, despite the fact that $\mathrm{Sat}_\perp(\Psi) \neq \mathrm{Sat}_\top(\Psi)$ here.

This example demonstrates the key aspects of the parameter synthesis method for nested CSL formulas. In particular, it shows that the refinement of the parameter space yields the refinement of the satisfaction sets and can result in parameter subspaces where, for a given state formula Φ, $\mathrm{Sat}_\perp(\Phi) = \mathrm{Sat}_\top(\Phi)$. Note that, if additional refinements are needed for such subspaces, the corresponding probability bounds can be computed in the same way as non-nested formulas. This example

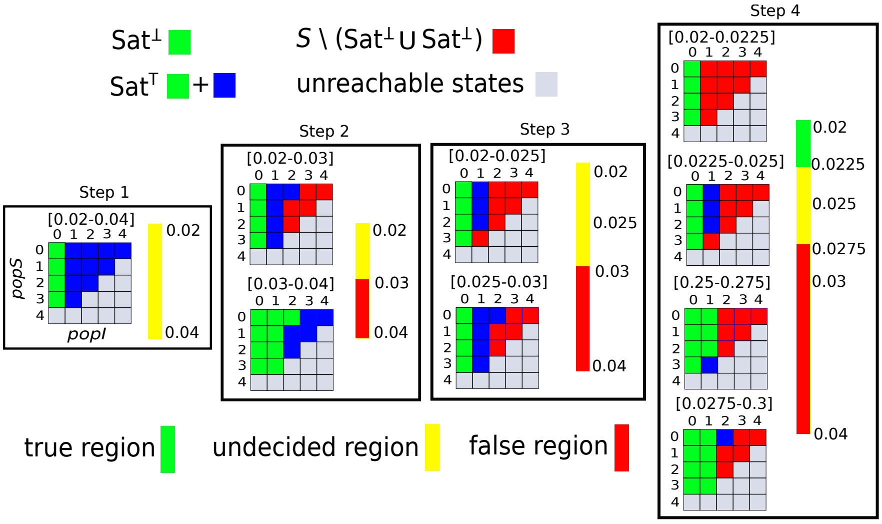


Fig. 7 Visualisation of four steps of the threshold synthesis algorithm for a small variant of the SIR model, nested CSL formula $\mathrm{P}_{>0.1}[(I > 0)\mathrm{U}^{[100,120]}(\Psi)]$ $(\Psi = \mathrm{P}_{>0.9}[F^{[0,100]}(I = 0)])$ and parameter space $k_i \times k_r = 0.12 \times [0.02, 0.04]$. The two-dimensional grids represent the state space. Since the total population is preserved by the model dynamics, each state is unambiguously given by the population of S (*popS*) and population of I (*popI*). For a given parameter region, the satisfaction sets $\mathrm{Sat}_{\perp}(\Psi)$ and $\mathrm{Sat}_{\top}(\Psi)$ for Ψ are depicted using different colouring of the grid cells. For each step, the vertical coloured bar illustrates the current partitioning of the parameter space into regions $\mathcal{T}, \mathcal{U}$ and $\mathcal{F}$.

also demonstrates that a region can be decided even when $\mathrm{Sat}_{\perp}(\Phi) \neq \mathrm{Sat}_{\top}(\Phi)$. This considerably reduces the number of required refinements for the subformula Φ. In general, the synthesis algorithms for the nested formulas have a higher complexity, since the nesting of probabilistic operators increases the number of regions to analyse.

6.2 DNA walkers

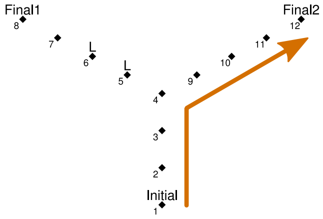
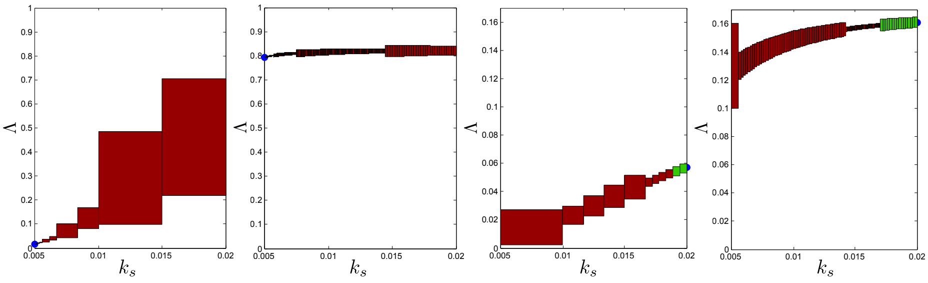


Fig. 8 Single-junction DNA walker circuit. In orange: the walker starts on the Initial anchorage and moves right at the junction, eventually quenching the fluorophore at the Final2 anchorage.

We revisit models of a DNA walker, a man-made molecular motor that traverses a track of anchorages and can take directions at junctions in the track [43], which can be used to create circuits that evaluate Boolean functions. PRISM models of the walker stepping behaviour were developed previously [17] based on rate estimates in the experimental work. The walker model is modified here to allow uncertainty in the stepping rate, and we consider its behaviour over a single-junction circuit, see Figure 8. Following [17], the stepping rate k is parameterised by d, the distance between the walker-anchorage complex and an uncut anchorage, and d_a, the distance between consecutive anchorages, and is



(a) Min, $T = 15$ (b) Min, $T = 200$ (c) Max, $T = 15$ (d) Max, $T = 200$

			Runtime		Subspaces	
Time bound	Min. correct	Max. incorrect	$\emptyset$	Sampling	$\emptyset$	Sampling
$T = 15$	1.68%	5.94%	0.55 s	0.51 s	22	11
$T = 30$	14.86%	10.15%	1.43 s	1.35 s	35	15
$T = 45$	33.10%	12.25%	3.53 s	2.14 s	61	21
$T = 200$	79.21%	16.47%	213.57s	88.97 s	909	329

Fig. 9 Experiments for the DNA walker model: min-synthesis for $\phi = \mathrm{F}^{[T,T]}$ finish-correct and max-synthesis for $\phi = \mathrm{F}^{[T,T]}$ finish-incorrect using $k_s \in [0.005, 0.020], c = 1$ and probability tolerance $\epsilon = 1\%$. Plots show the solutions and the decompositions for the min (a,b) and max (c,d) synthesis and $T = 15, 200$ minutes. Colour code is as in Fig. 4. In the table, results are reported also for $T = 30, 45$. The runtime and subspaces are listed only for min-synthesis (the results for max-synthesis are similar).

defined as:

$$k = \begin{cases} k_s & \text{when } d \leq 1.5 d_a \\ c \cdot k_s/50 & \text{when } 1.5 d_a < d \leq 2.5 d_a \\ c \cdot k_s/100 & \text{when } 2.5 d_a < d \leq 24\text{nm} \\ 0 & \text{otherwise.} \end{cases} \tag{43}$$

where the base stepping rate $k_s \in [0.005, 0.020]$ is now defined as an interval, as opposed to the original value of 0.009. We have also added factor c for steps between anchorages that are not directly adjacent, but we will assume $c = 1$ for now. The base stepping rate may depend on buffer conditions and temperature, and we want to verify the robustness of the walker with respect to the uncertainty in the value of k_s.

We compute the minimal probability of the walker making it onto the correct final anchorage by time T (min synthesis for the formula $\phi = \mathrm{F}^{[T,T]}$ finish-correct) and the maximum probability of the walker making it onto the incorrect anchorage by time T (max synthesis for the formula $\phi = \mathrm{F}^{[T,T]}$ finish-incorrect). In Figure 9, we list the probabilities at $T = 15, 30, 45, 200$ minutes and depict the solutions and the parameter space decompositions for both experiments at $T = 15, 200$ minutes. For time $T = 30, 45, 200$, we note that the walker is robust, as the minimal guaranteed probability for the correct outcome is greater than the maximum possible probability for the incorrect outcome. For time $T = 15$ this is not the case. From plots (a-d), we observe that minimum and maximum probability values are obtained for minimum and maximum values of k_s, respectively.

We also consider a property that provides bounds on the *ratio* between the walker finishing on the correct versus the incorrect anchorage. The rates $c \cdot k_s/50$

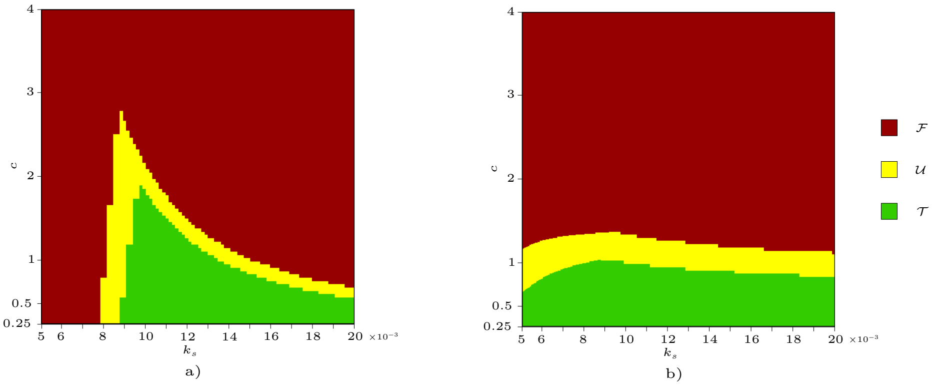
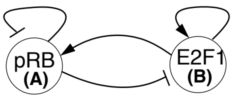


Fig. 10 The computation and results of the threshold synthesis for the DNA walker model, $k_s \times c \in [0.005, 0.020] \times [0.25, 4]$ and different formulas, using volume tolerance $\varepsilon = 10\%$. a) $\mathrm{P}_{\geq 0.4}[\mathrm{F}^{[30,30]}$ finish-correct$] \wedge \mathrm{P}_{\leq 0.08}[\mathrm{F}^{[30,30]}$ finish-incorrect$]$, runtime 282.5 s, 3489 subspaces$*$. b) $\mathrm{P}_{\geq 0.8}[\mathrm{F}^{[200,200]}$ finish-correct$] \wedge \mathrm{P}_{\leq 0.16}[\mathrm{F}^{[200,200]}$ finish-incorrect$]$, runtime 8.2 h, 47229 subspaces$*$. Results are obtained by solving the synthesis problem for each sub-formula and by combining the output regions as in Eqn 6. $*$: derived as the sum of runtimes and subspaces for each sub-formula.

and $c \cdot k_s/100$ correspond to the walker stepping onto anchorages that are not directly adjacent, which affects the probability for the walker to end up on the unintended final anchorage. For higher values of c, the walker is more likely to reach the unintended final anchorage more often. Now we add uncertainty on the value of c, so that $c \in [0.25, 4]$, and define the performance related property by $\mathrm{P}_{\geq 0.4}[\mathrm{F}^{[30,30]}$ finish-correct$] \wedge \mathrm{P}_{\leq 0.08}[\mathrm{F}^{[30,30]}$ finish-incorrect$]$, that is, the probability of the walker to make it onto the correct anchorage is at least 40% by time $T = 30$ min, while the probability for it to make it onto the incorrect anchorage is no greater than 8%. In other words, we require a correct signal of at least 40% and a correct-to-incorrect ratio of at least 5 by time $T = 30$ min. We define a similar property at time $T = 200$ min, this time requiring a signal of at least 80%: $\mathrm{P}_{\geq 0.8}[\mathrm{F}^{[200,200]}$ finish-correct$] \wedge \mathrm{P}_{\leq 0.16}[\mathrm{F}^{[200,200]}$ finish-incorrect$]$. The synthesized ranges of k_s and c where the properties hold are shown in Figure 10. Note that in this case the rate function is a multi-affine polynomial and for fixed c (Figure 9) the function is linear.

6.3 Gene Regulation of Mammalian Cell Cycle

We consider the gene regulation model published in [42]. The model is shown in Fig. 11a and explains the regulation of a transition between the early phases of the mammalian cell cycle. In particular, it targets the transition from the control G_1-phase to S-phase (the synthesis phase). The G_1-phase makes an important checkpoint controlled by a *bistable regulatory circuit*, based on an interplay of the retinoblastoma protein *pRB*, denoted by A (the so-called tumour suppressor, HumanCyc:HS06650) and the retinoblastoma-binding transcription factor E_2F_1, denoted by B (a central regulator of a large set of human genes, HumanCyc:HS02261). At high concentration levels (*high mode*), B activates the G_1/S transition mechanism. On the other hand, a low concentration of B (*low mode*)



(a)

$A < M_A \wedge B > 0 \rightarrow$	$k_1 \frac{B}{(K_{m1}+B)} \frac{J_{11}}{J_{11}+A} : A' = A + 1$
$B < M_B \rightarrow$	$k_2 \frac{a^2+B^2}{(K_{m2}^2+B^2)} \frac{J_{12}}{J_{12}+A} : B' = B + 1$
$B < M_B \rightarrow$	$k_p : B' = B + 1$
$A > 0 \rightarrow$	$\gamma_A \cdot A : A' = A - 1$
$B > 0 \rightarrow$	$\gamma_B \cdot B : B' = B - 1$

(b)

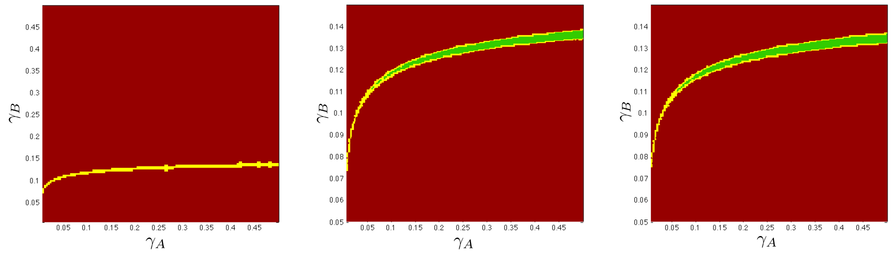
Fig. 11 (a) Two-gene regulatory circuit controlling G_1/S transition in mammalian cell cycle ($\dashv$ indicates inhibition, → activation). (b) Stochastic Michaelis-Menten model of the G_1/S regulatory circuit adapted from [42] in the guarded command notation (*guard* → *rate* : *update*). A and B are the number of proteins pRB and E_2F_1, respectively. Rates are expressed in s^{-1}. Production rate parameters are as in [42]: $k_1 = 1$, $k_2 = 1.6$, $k_p = 0.05$, $K_{m1} = 0.5$, $K_{m2} = 4$, $a = 0.04$, $J_{11} = 0.5$ and $J_{12} = 5$. $M_A = M_B = 30$ are the bounds on A and B, estimated through stochastic simulations. Synthesis parameters are: $\gamma_A \in [0.005, 0.5]$ (degradation rate of pRB), and $\gamma_B \in [0.005, 0.5]$ (degradation rate of E_2F_1).

prevents committing to S-phase. Depending on the parameters, the positive autoregulation of B causes bistability of its concentration, i.e. the existence of two stable states. Of specific interest are the degradation rates of A (γ_A) and B (γ_B). When mitogenic stimulation increases under conditions of active growth, rapid phosphorylation of A starts and makes the degradation of unphosphorylated A stronger (i.e. the rate γ_A increases). This causes B to lock in the high stable mode implying the cell cycle commits to the S-phase. Since mitogenic stimulation influences the degradation rate of A, our goal is to study the population distribution around the low and high steady states and to explore the effect of γ_A and γ_B.

The ODE model in [42] describes this system using Michaelis-Menten (MM) kinetics [35], which provides a deterministic approximation for enzyme-catalysed reactions. In this work, we derive a discrete stochastic translation of the ODE model by directly using the MM rates. The model is illustrated in Fig. 11b. The corresponding CTMC has 961 states and 3690 transitions. For details on the adequacy of the MM approximation in the stochastic settings, we refer the reader to [37,39].

We apply threshold synthesis to find the degradation rates that lead to bistability. In particular, synthesis parameters are $\gamma_A \in [0.005, 0.5]$ and $\gamma_B \in [0.005, 0.5]$ and thus the rate functions are linear with respect to the parameters. We formalize stability using time-bounded properties with time horizon 1000 seconds, which reflects the time scale of the gene regulation response. The stabilization of the model's dynamics within this time horizon was also confirmed by a steady-state analysis for different values of the parameters. We use atomic propositions H and L to denote the high and low mode of B, respectively. In the following experiments, we assume L is true if $B < 2$ and H is true if $B > 4$.

Bistability is commonly expressed as the property of reaching and staying in either one of two different regions of the state space. In time-bounded CSL, this can be expressed using the formula $\mathrm{P}_{\geq r_L}[\mathrm{G}^I L] \wedge \mathrm{P}_{\geq r_H}[\mathrm{G}^I H]$, where I spans the final time window and the probability of resting in the two modes is at least r_L and r_H, respectively. However, threshold synthesis for this property, $I = [900, 1000]$ and different combinations of r_L and r_H resulted in empty $\mathcal{T}$ regions and negligible $\mathcal{U}$ regions, indicating that no parameters can be found that meet this formulation of bistability. Note that this result agrees with the analysis performed in [11] using



(a) (b) (c)

	$\gamma_A \times \gamma_B$	Runtime$*$	Subspaces$*$
a)	$[0.005, 0.5] \times [0.005, 0.5]$	618 s	809
b)	$[0.005, 0.5] \times [0.05, 0.15]$	1256 s	1817
c)	$[0.005, 0.5] \times [0.05, 0.15]$	1882 s	1742

Fig. 12 Bistability analysis in the G_1/S regulatory circuit model: threshold synthesis for property $\mathrm{P}_{\geq 0.4}[\mathrm{F}^t L] \wedge \mathrm{P}_{\geq 0.4}[\mathrm{F}^t H]$ (a,b) and $\mathrm{R}_{\geq 400}[\mathrm{C}^{\leq 1000}](B < 2) \wedge \mathrm{R}_{\geq 400}[\mathrm{C}^{\leq 1000}](B > 4)$ (c) with volume tolerance $\varepsilon = 1\%$. In b) and c) we improve precision by restricting the range of γ_B. Colour code is as in Fig. 3 a). Results are obtained by solving the synthesis problem for each sub-formula and by combining the output regions as in Eqn 6. $*$: derived as the sum of runtimes and subspaces for each sub-formula.

a different stochastic translation of the ODE model and parameter exploration techniques.

Therefore, we analyse whether there is at least a weaker type of bistability in the form of a bi-modal distribution at time $t = 1000$. The existence of the bi-modal distribution is formalized using the future operator F as:

$$\mathrm{P}_{\geq r_L}[\mathrm{F}^t L] \wedge \mathrm{P}_{\geq r_H}[\mathrm{F}^t H].$$

Results for $r_L = r_H = 0.4$ are summarised in Figure 12. In plot a), we observe that most of the degradation rates violate the formula, with the exception of an undecided region that spans the domain of γ_A and $\gamma_B \in [0.05, 0.15]$. Although this result is consistent with input volume tolerance ε, it does not provide a precise answer to whether satisfiable parameters exist or not.

In order to resolve the undecided region, we perform the same experiment by restricting the parameter space to the area of interest (plot b). Indeed, note that, by the volume tolerance criterion, a lower volume of $\mathcal{P}$ implies a lower volume of $\mathcal{U}$. An alternative (less efficient) solution would be keeping $\mathcal{P}$ unchanged and decreasing ε. The results demonstrate that a bi-modal distribution is reached for a relevant set of parameters. In particular, we show that, for any value of γ_A approximately ranging in $[0.08, 0.5]$, there exists a value of γ_B that satisfy the property.

To complement the above analysis, we further analyse how long the system remains in the low and high mode. To this purpose, we synthesise parameters such that the following cumulative reward property is satisfied:

$$\mathrm{R}_{\geq 400}[\mathrm{C}^{\leq 1000}](B < 2) \wedge \mathrm{R}_{\geq 400}[\mathrm{C}^{\leq 1000}](B > 4)$$

where, for subformula $\mathrm{R}_{\geq 400}[\mathrm{C}^{\leq 1000}](B \sim X)$, the state reward ρ is such that: $\forall s \in S.\ \rho(s) = 1 \Leftrightarrow B \sim X$ in s. Figure 12 c) illustrates the results of parameter synthesis restricted to the parameter space as in the previous experiment. The parameters that satisfy the reward property are aligned with the parameters leading to the bimodal distribution. This experiment confirms the existence of a certain form of the bistability in the stochastic version of the model. Moreover, our results are comparable with those obtained using numerical simulations of the ODE model and bifurcation analysis in [42], where bistability is registered[2] for $\gamma_A \in [0.007, 0.03]$ and fixed $\gamma_B = 0.1$.

7 Related work

The work by Brim et al. [11] first introduced extensions of CSL model checking and of the uniformisation method for computing safe probability bounds in parametric models of stochastic reaction networks. The methods are applied to the problem of parameter exploration, i.e. finding bounds that approximate the satisfaction function arbitrarily well. Our work extends [11] with the following five main contributions. (1) Definition of the threshold and max synthesis problems. (2) Synthesis algorithms that combine iterative refinement of the parameter space with sampling of probability values. In contrast to [11], where every parameter region needs to be analyzed, our algorithms focus only on the regions relevant to solve the synthesis problem, so avoiding unnecessary computation. (3) More general class of supported rate functions. Specifically, we allow multi-affine dependency of the rate functions in the parameters, while the method in [11] supports only rate functions linear in the parameters. (4) Convergence analysis of the approximation error (see Proposition 2). (5) Theorem 1, which provides a precise mathematical characterization for the satisfaction function of generic time-bounded and finitely-nested CSL formulas. Furthermore, the gene regulation model in Section 6.3 is also studied by Brim et al. [11]. They translate the original ODE model [42] into the framework of stochastic mass action kinetics, while we consider a stochastic Michaelis-Menten approximation. In [11], no parameters are found that meet the bistability requirement, expressed using 'globally' and cumulative reward formulas. Our analyses confirm that the model does not exhibit bistability if defined with 'globally' formulas, but that bistability occurs if formulated as a bounded reachability or a cumulative reward formula. This is explained by the different stochastic encodings of the original ODE model.

Parameter synthesis for CTMCs and bounded reachability specifications is considered in [23]. The authors show that the problem can be reduced to the analysis of the polynomial function describing the reachability probability of a given target state. As illustrated in Section 4.1, the main limitation is the high degree of the polynomials, which is determined by the number of uniformisation steps. Thus, in contrast to our work, only an approximate solution can be obtained using discretization of the parameter space and the experimental evaluation is limited to one simple case study (a CTMC with 19 states and 54 transitions).

When considering linear-time specifications, specific restrictions can be placed on the rate function to result in a smooth satisfaction function (i.e. having derivatives of all orders). In that case the function can be approximated using statistical methods such as Gaussian Process regression, which leverage smoothness [9]. Such restrictions require the rate function to be smooth in the parameters and polynomial in the state vector. Instead, the synthesis method we presented imposes

[2] In [42], we believe that there is a typo in the figure illustrating bistability (Figure 2B). The range of ϕ_{pRB} (corresponding to γ_A) should read 0.005-0.035.

limitations only on the parameter dependency. In contrast to our approach, the statistical estimation of the satisfaction function cannot provide guaranteed results. Moreover, the precision of the estimation strongly depends on a number of experimental design choices, e.g. how many and which parameter values to sample, while our algorithms provide results with arbitrary precision and do not require any information except the inputs to the synthesis problem.

A concept similar to smoothness, uniform continuity, can be used to obtain an unbiased statistical estimator for the satisfaction function [25]. Inference of parameter values in probabilistic models from time-series measurements is a well studied area of research [2, 10], but different from the problem we consider.

Parameter synthesis problems have been studied for discrete-time Markovian models in [22, 14]. These approaches apply to unbounded temporal properties and are based on constructing a rational function by performing state elimination [22].

Interval CTMCs, where transition rates are given as intervals, have been employed to obtain a three-valued abstraction for CTMCs [26]. In contrast to the parametric models we work with, the transition rates in interval CTMCs are chosen nondeterministically and schedulers are introduced to compute lower and upper probability bounds. Along the same lines, [41] introduce models and model checking algorithms for interval DTMCs and MDPs.

Synthesis approaches for non-stochastic biological models include the work by Batt et al. [6], where parameters of ODE-based gene regulatory network (GRN) models are synthesized using discrete abstractions and LTL model checking. In [5], the method of [18] based on sensitivity analysis is applied to the automated design of synthetic biological devices from basic ODE modules. In [28], parameter synthesis for discrete GRNs is reduced to coloured LTL model checking and solved through a distributed algorithm. Methods based on SMT are presented in [29, 36] to synthesize Boolean network models from time-series and perturbation experiments data.

8 Conclusion and Discussion

We have developed efficient algorithms for synthesising rate parameters for biochemical networks so that a given requirement, expressed as a time-bounded CSL formula, is guaranteed to be satisfied. The techniques are based on the computation of lower and upper probability bounds of [11], in conjunction with iterative refinement and sampling of parameters. In this work we focus on biological systems that, being characterised by complex and non-linear dynamics, are typically hard to analyse. However, our synthesis algorithms can be equivalently applied also to the performance and reliability analysis of computer systems.

We remark that improved performance can be easily achieved through parallel processing of individual subspaces and, within each subspace, of the parametric uniformisation method. Other techniques can also be integrated to speed up the synthesis process, including fast adaptive uniformisation [34, 16], state aggregation [44, 1], and abstraction [33]. Finally, we plan to include the synthesis algorithms in the `param` module of the `PRISM` model checker [14, 32], and to extend the method to general non-linear rate functions.

Acknowledgements.

We thank David Šafránek for useful discussions about the stochastic models of gene regulation of mammalian cell cycle and Nicolas Basset for helping with the termination of the threshold synthesis algorithm. We also thank Andrej Tokarčík and Petr Pilař for implementing the prototype version of the synthesis algorithms. We finally thank the anonymous reviewers for their insightful feedback.

References

1. Abate, A., Brim, L., Češka, M., Kwiatkowska, M.: Adaptive aggregation of markov chains: Quantitative analysis of chemical reaction networks. In: Computer Aided Verification (CAV), *LNCS*, vol. 9206, pp. 195–213. Springer (2015)
2. Andreychenko, A., Mikeev, L., Spieler, D., Wolf, V.: Parameter Identification for Markov Models of Biochemical Reactions. In: Computer Aided Verification (CAV), LNCS, pp. 83–98. Springer (2011)
3. Aziz, A., Sanwal, K., Singhal, V., Brayton, R.: Verifying Continuous Time Markov Chains. In: Computer Aided Verification (CAV), *LNCS*, vol. 1102, pp. 269–276. Springer (1996)
4. Baier, C., Haverkort, B., Hermanns, H., Katoen, J.: Model-Checking Algorithms for Continuous-Time Markov Chains. IEEE Trans. on Soft. Eng. **29**(6), 524–541 (2003)
5. Bartocci, E., Bortolussi, L., Nenzi, L.: A temporal logic approach to modular design of synthetic biological circuits. In: Computational Methods in Systems Biology (CMSB), pp. 164–177. Springer (2013)
6. Batt, G., Yordanov, B., Weiss, R., Belta, C.: Robustness analysis and tuning of synthetic gene networks. Bioinformatics **23**(18), 2415–2422 (2007)
7. Belta, C., Habets, L.: Controlling a class of nonlinear systems on rectangles. IEEE Trans. on Automatic Control **51**(11), 1749–1759 (2006)
8. Billingsley, P.: Probability and measure. John Wiley & Sons (2008)
9. Bortolussi, L., Milios, D., Sanguinetti, G.: Smoothed model checking for uncertain continuous time markov chains. CoRR ArXiv **1402.1450** (2014)
10. Bortolussi, L., Sanguinetti, G.: Learning and designing stochastic processes from logical constraints. In: Quantitative Evaluation of Systems (QEST), *LNCS*, vol. 8054, pp. 89–105. Springer (2013)
11. Brim, L., Češka, M., Dražan, S., Šafránek, D.: Exploring parameter space of stochastic biochemical systems using quantitative model checking. In: Computer Aided Verification (CAV), *LNCS*, vol. 8044, pp. 107–123. Springer (2013)
12. Caron, R., Traynor, T.: The zero set of a polynomial. Tech. rep., University of Windsor (2005)
13. Češka, M., Dannenberg, F., Kwiatkowska, M., Paoletti, N.: Precise parameter synthesis for stochastic biochemical systems. In: Computational Methods in Systems Biology (CMSB), pp. 86–98. Springer (2014)
14. Chen, T., Hahn, E.M., Han, T., Kwiatkowska, M., Qu, H., Zhang, L.: Model repair for Markov decision processes. In: Theoretical Aspects of Software Engineering (TASE), pp. 85–92. IEEE (2013)
15. Courant, R., John, F.: Introduction to calculus and analysis, vol. 2. Springer Science & Business Media (2012)
16. Dannenberg, F., Hahn, E.M., Kwiatkowska, M.: Computing cumulative rewards using fast adaptive uniformisation. In: Computational Methods in Systems Biology (CMSB), *LNCS*, vol. 8130, pp. 33–49. Springer (2013)
17. Dannenberg, F., Kwiatkowska, M., Thachuk, C., Turberfield, A.: DNA walker circuits: Computational potential, design, and verification. Natural Computing (2014). To appear
18. Donzé, A.: Breach, A Toolbox for Verification and Parameter Synthesis of Hybrid Systems. In: Computer Aided Verification (CAV), *LNCS*, vol. 6174, pp. 167–170. Springer (2010)
19. Fox, B.L., Glynn, P.W.: Computing Poisson Probabilities. CACM **31**(4), 440–445 (1988)
20. Gillespie, D.T.: Exact Stochastic Simulation of Coupled Chemical Reactions. Journal of Physical Chemistry **81**(25), 2340–2381 (1977)
21. Grassmann, W.: Transient Solutions in Markovian Queueing Systems. Computers & Operations Research **4**(1), 47 – 53 (1977)

22. Hahn, E.M., Hermanns, H., Zhang, L.: Probabilistic reachability for parametric Markov models. International Journal on Software Tools for Technology Transfer (STTT) **13**(1), 3–19 (2011)
23. Han, T., Katoen, J., Mereacre, A.: Approximate parameter synthesis for probabilistic time-bounded reachability. In: Real-Time Systems Symposium (RTSS), pp. 173–182. IEEE (2008)
24. Jensen, A.: Markoff chains as an aid in the study of Markoff processes. Skand. Aktuarietidskr. **36**, 87–91 (1953)
25. Jha, S.K., Langmead, C.J.: Synthesis and infeasibility analysis for stochastic models of biochemical systems using statistical model checking and abstraction refinement. Theor. Comput. Sci. **412**(21), 2162–2187 (2011)
26. Katoen, J.P., Klink, D., Leucker, M., Wolf, V.: Three-valued abstraction for continuous-time markov chains. In: Computer Aided Verification (CAV), *LNCS*, vol. 4590, pp. 311–324. Springer (2007)
27. Kermack, W., McKendrick, A.: Contributions to the mathematical theory of epidemicsii. the problem of endemicity. Bulletin of mathematical biology **53**(1), 57–87 (1991)
28. Klarner, H., Streck, A., Šafránek, D., Kolčák, J., Siebert, H.: Parameter identification and model ranking of thomas networks. In: Computational Methods in Systems Biology (CMSB), pp. 207–226. Springer (2012)
29. Koksal, A.S., Pu, Y., Srivastava, S., Bodik, R., Fisher, J., Piterman, N.: Synthesis of biological models from mutation experiments. SIGPLAN Not. **48**(1), 469–482 (2013)
30. Kwiatkowska, M., Norman, G., Pacheco, A.: Model checking expected time and expected reward formulae with random time bounds. Computers & Mathematics with Applications **51**, 305–316 (2006)
31. Kwiatkowska, M., Norman, G., Parker, D.: Stochastic Model Checking. In: Formal Methods for Performance Evaluation (SFM), *LNCS*, vol. 4486, pp. 220–270. Springer (2007)
32. Kwiatkowska, M., Norman, G., Parker, D.: PRISM 4.0: Verification of Probabilistic Real-time Systems. In: CAV 2011, *LNCS*, vol. 6806, pp. 585–591. Springer (2011)
33. Madsen, C., Myers, C., Roehner, N., Winstead, C., Zhang, Z.: Utilizing stochastic model checking to analyze genetic circuits. In: Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), pp. 379–386. IEEE (2012)
34. Mateescu, M., Wolf, V., Didier, F., Henzinger, T.A.: Fast Adaptive Uniformization of the Chemical Master Equation. IET Systems Biology **4**(6), 441–452 (2010)
35. Michaelis, L., Menten, M.L.: Die kinetik der invertinwirkung. Biochem. z **49**(333-369), 352 (1913)
36. Paoletti, N., Yordanov, B., Hamadi, Y., Wintersteiger, C.M., Kugler, H.: Analyzing and synthesizing genomic logic functions. In: Computer Aided Verification (CAV), pp. 343–357. Springer (2014)
37. Rao, C.V., Arkin, A.P.: Stochastic chemical kinetics and the quasi-steady-state assumption: application to the gillespie algorithm. The Journal of chemical physics **118**(11), 4999–5010 (2003)
38. Reibman, A.: Numerical transient analysis of Markov models. Computers & Operations Research **15**(1), 19–36 (1988)
39. Sanft, K.R., Gillespie, D.T., Petzold, L.R.: Legitimacy of the stochastic Michaelis-Menten approximation. Systems Biology, IET **5**(1), 58–69 (2011)
40. Sassi, B., Amin, M., Girard, A.: Control of polynomial dynamical systems on rectangles. In: European Control Conference (ECC), pp. 658–663. IEEE (2013)
41. Sen, K., Viswanathan, M., Agha, G.: Model-checking markov chains in the presence of uncertainties. In: Tools and Algorithms for the Construction and Analysis of Systems (TACAS), *LNCS*, vol. 3920, pp. 394–410. Springer (2006)
42. Swat, M., Kel, A., Herzel, H.: Bifurcation Analysis of the Regulatory Modules of the Mammalian G1/S transition. Bioinformatics **20**(10), 1506–1511 (2004)
43. Wickham, S.F.J., Bath, J., Katsuda, Y., Endo, M., Hidaka, K., Sugiyama, H., Turberfield, A.J.: A DNA-based molecular motor that can navigate a network of tracks. Nature nanotechnology **7**, 169–73 (2012)
44. Zhang, J., Watson, L.T., Cao, Y.: Adaptive aggregation method for the chemical master equation. Int. J. of Computational Biology and Drug Design **2**(2), 134–148 (2009)

A Proofs

A.1 Proposition 2

Consider a parameter region $\mathcal{R} = [x_1^{\perp}, x_1^{\top}] \times \ldots \times [x_n^{\perp}, x_n^{\top}] \subseteq \mathcal{P}$. Fix an arbitrary state s and let the maximizing argument of the transient probability in s be (cf. Eqn 8):

$$p^* = \arg\max_{p \in \mathcal{R}} \pi_t(s) \tag{44}$$

and let $\tau_i^* = \pi_0 \mathbf{P}_{p^*}^i$. The local error introduced during a single discrete time step for state s, in the multi-affine case, is given by

$$e_i(s) = \frac{1}{q} |\mathrm{flux}(\tau_{i-1}^*, s)(p^*) - \max_{p \in V_{\mathcal{R}}} \mathrm{flux}(\tau_{i-1}^*, s)(p)|. \tag{45}$$

Note that in this analysis the local error is expressed for the discrete distribution $\tau_i^{\max}$, from which the solution $\pi_i^{\max}$ is derived as a linear combination. We now analyse the global error, which for $i > 1$ is given as:

$$g_i(s) = |\tau_i^*(s) - \tau_i^{\max}(s)| \tag{46}$$

$$= |\tau_{i-1}^*(s) + \frac{1}{q}\mathrm{flux}(\tau_{i-1}^*, s)(p^*) - \tau_{i-1}^{\max}(s) - \frac{1}{q} \max_{p \in V_{\mathcal{R}}} \mathrm{flux}(\tau_{i-1}^{\max}, s)(p)| \tag{47}$$

where we now use the definition of $g_i(s)$ and employ triangle inequality:

$$\leq g_{i-1}(s) + \frac{1}{q}\big|\mathrm{flux}(\tau_{i-1}^*, s)(p^*) - \max_{p \in V_{\mathcal{R}}} \mathrm{flux}(\tau_{i-1}^{\max}, s)(p)\big| \tag{48}$$

$$\leq g_{i-1}(s) + \frac{1}{q}\big|\mathrm{flux}(\tau_{i-1}^*, s)(p^*) - \max_{p \in V_{\mathcal{R}}} \mathrm{flux}(\tau_{i-1}^* + g_{i-1}, s)(p)\big| \tag{49}$$

where the overall global error g_i is the vector-wise equivalent of $g_i(s)$ and we continue

$$g_i(s) \leq g_{i-1}(s) + \frac{1}{q} \max_{p \in V_{\mathcal{R}}} \mathrm{flux}(g_{i-1}, s)(p) \tag{50}$$

$$+ \frac{1}{q}\big|\mathrm{flux}(\tau_{i-1}^*, s)(p^*) - \max_{p \in V_{\mathcal{R}}} \mathrm{flux}(\tau_{i-1}^*, s)(p)\big| \tag{51}$$

$$\leq g_{i-1}(s) + \frac{1}{q} \max_{p \in V_{\mathcal{R}}} \mathrm{flux}(g_{i-1}, s)(p) + e_i(s) \tag{52}$$

The form of $g_i(s)$ is understood as follows. The error in the current step is less than the error in the previous step plus the maximal local error plus the worst-case additional flux resulting from the approximation error in the previous step. Let the width of the parameter space $\mathcal{R}$ be given as $w_{\mathcal{R}} = \max_j (x_j^{\top} - x_j^{\perp})$.

We now show how to derive bounds $M_1, M_2 < \infty$ such that

$$e_i \leq \frac{M_1}{q} w_{\mathcal{R}} \tag{53}$$

$$\max_{p \in V_{\mathcal{R}}} \mathrm{flux}(g_i, s)(p) \leq M_2 \cdot \max_{s \in S} g_{i-1}(s) \tag{54}$$

Observe that, if the two inequalities above hold, we get the same over-approximation as in Eqn 42, which would prove the proposition true. For the local error we find

$$e_i(s) \leq \frac{1}{q} \max_{p \in V_{\mathcal{R}}} |\mathrm{flux}(\tau_{i-1}^*, s)(p^*) - \mathrm{flux}(\tau_{i-1}^*, s)(p)| \tag{55}$$

and in this case a Lipschitz constant $M_{1,s}$ exists such that

$$e_i(s) \leq \frac{M_{1,s}}{q} w_{\mathcal{R}} \tag{56}$$

Thus, $M_1 = \max_{s \in S} M_{1,s}$ is such that Eqn 53 holds. The maximum flux that propagates due to the approximation error in the previous step is given as

$$\max_{p \in V_{\mathcal{R}}} \text{flux}(g_{i-1}, s)(p). \tag{57}$$

Now regard $\text{flux}(g_{i-1}, s)(p)$ as a function of g_{i-1}, with p, s fixed. Note that the domain of the function is given by $\mathcal{R}' = \bigtimes_{s \in S} [0, g_{i-1}(s)]$, and thus $w_{\mathcal{R}'} = \max_{s \in S} g_{i-1}(s)$. Also, $\text{flux}(\mathbf{0}, s)(p) = 0$ for $\mathbf{0} : S \to 0$. Thus, another Lipschitz constant $M_{2,p,s}$ exists such that

$$\text{flux}(g_{i-1}, s)(p) = \text{flux}(g_{i-1}, s)(p) - \text{flux}(\mathbf{0}, s)(p) \leq M_{2,p,s} \cdot \max_{s \in S} g_{i-1}(s) \tag{58}$$

By taking $M_2 = \max_{p \in \mathcal{R}, s \in S} M_{2,p,s}$, Eqn 54 holds. Summarizing, the global error on $\tau_i^{\max}(s)$ is bounded by $\overline{g}_i$, under the assumption of multi-affine rate functions, as

$$\overline{g}_i = \begin{cases} 0 & \text{if } i = 0 \\ \overline{g}_{i-1} \cdot \left(1 + \frac{M_2}{q}\right) + \frac{M_1}{q} w_{\mathcal{R}} & \text{if } i > 0, \end{cases} \tag{59}$$

A.2 Theorem 1

The probability of satisfying an unnested and time-bounded CSL property given a CTMC $\mathcal{C}_p = (S, s_0, \mathbf{R}_p, L)$ reduces to the computation of transient-state probabilities over a CTMC $\mathcal{C}'_p$ for which the transition relation $\mathbf{R}'_p$ is easily derived from the original CTMC [4,31]. Recalling Definition 4, the transient-state probability is given by standard uniformisation:

$$\hat{\pi}_{t,p} = \pi_0 \sum_{i=0}^{k_\epsilon} \gamma_{i,qt} \mathbf{P}_p^i. \tag{60}$$

Provided the entries in $\mathbf{P}$ are polynomials of finite degree, the expression $\hat{\pi}_t(s)$ is also a finite-degree polynomial over domain $\mathcal{P}$. We now prove that the approximate satisfaction function $\hat{\Lambda}_\phi$ of any time-bounded finitely-nested path formula can be expressed as a piecewise polynomial function with a finite number of subdomains, using the above as the base case in our induction. Note that only the path operators until (U) and next (X) allow nesting. We prove the induction step only for the until operator, since the proof for the next operator can be obtained in a similar way.

Let ϕ_1, ϕ_2 be time-bounded CSL path formulas such that $\text{P}_{=?}[\phi_1]$, $\text{P}_{=?}[\phi_2]$ are piecewise polynomial with a finite number of subdomains. Consider the nested formula:

$$\text{P}_{=?}\left[\text{P}_{\sim_1 r_1}[\phi_1] \ \text{U}^I \ \text{P}_{\sim_2 r_2}[\phi_2]\right] \tag{61}$$

for a subdomain $I \in \mathbb{R}_{\geq 0}$, bounds r_1, r_2 and $\sim_i \in \{<, \leq, \geq, >\}$.

Observe that, if the satisfaction sets $v_i = \{s \in S \mid s \models \text{P}_{\sim_i r_i}[\phi_i]\}$ for $i = 1, 2$ are constant over a subspace $\mathcal{R} \subseteq \mathcal{P}$, then the expression in Eqn 61 is given by a polynomial function over $\mathcal{R}$, cf. Eqn 60. We will demonstrate that $\hat{\Lambda}_{\text{P}_{\sim_1 r_1}[\phi_1] \ \text{U}^I \ \text{P}_{\sim_2 r_2}[\phi_2]}$ is piecewise polynomial over finitely many subdomains by constructing a partition of $\mathcal{P}$ that is conditioned on the truth assignment of each state. Given a state s, allow the partition $\mathcal{T}_1(s) \cup \mathcal{F}_1(s) = \mathcal{P}$ where

$$\forall p \in \mathcal{T}_1(s) : s \models \text{P}_{\sim_1 r_1}[\phi_1] \tag{62}$$

$$\forall p \in \mathcal{F}_1(s) : s \not\models \text{P}_{\sim_1 r_1}[\phi_1]. \tag{63}$$

By the induction hypothesis $\text{P}_{=?}[\phi_1] - r_1$ is piecewise polynomial over finite many subspaces, so that $\mathcal{T}_1(s), \mathcal{F}_1(s)$ are unions of finitely many subspaces of $\mathcal{P}$. Assume a similar partition $\mathcal{T}_2(s) \cup \mathcal{F}_2(s) = \mathcal{P}$. We now wish to construct a partition $\bigcup_{v_1, v_2 \in 2^S} \mathcal{R}(v_1, v_2) = \mathcal{P}$ that is conditioned on the truth assignment of all states, so that $\forall p \in \mathcal{R}(v_1, v_2)$:

$$s \in v_1 \Leftrightarrow s \models \text{P}_{\sim_1 r_1}[\phi_1] \quad \wedge \quad s \in v_2 \Leftrightarrow s \models \text{P}_{\sim_2 r_2}[\phi_2] \tag{64}$$

in which case the expression of Eqn 61 is a polynomial function over $\mathcal{R}(v_1, v_2)$ because the truth-assignment of the nested formulas is fixed. We provide a constructive definition for $\mathcal{R}(v_1, v_2)$ by finite intersection of the sets $\mathcal{T}_i(s)$ and $\mathcal{F}_i(s)$, that is:

$$\mathcal{R}(v_1, v_2) = [\cap_{s \in v_1} \mathcal{T}_1(s)] \cap [\cap_{s \notin v_1} \mathcal{F}_1(s)] \cap [\cap_{s \in v_2} \mathcal{T}_2(s)] \cap [\cap_{s \notin v_2} \mathcal{F}_2(s)] \,. \tag{65}$$

Then $\mathcal{R}(v_1, v_2)$ is a union of a finite number of subdomains of $\mathcal{P}$. □

A.3 Proposition 3

We first show termination of the algorithm for an unnested property $P_{\geq r}[\phi]$. Define

$$f = \hat{\Lambda}_\phi(\cdot)(s_0) - r$$

and let the zero-set of f be given as $Z(f) = \{p \in \mathcal{P} \mid f(p) = 0\}$. In other words, $Z(f)$ is the set of parameters p yielding satisfaction probability equal to r, i.e. $\hat{\Lambda}_\phi(p)(s_0) = r$. Now note that, at a generic step of the algorithm, the undecided space is composed by those regions $\mathcal{R}_i$ such that $\Lambda^{\mathcal{R}_i}_{\min} < r$ and $\Lambda^{\mathcal{R}_i}_{\max} \geq r$. Assuming infinite precision, $\Lambda^{\mathcal{R}_i}_{\min}$ and $\Lambda^{\mathcal{R}_i}_{\max}$ can be made arbitrarily tight, i.e. the approximation error made arbitrarily small (cf Prop 2). Therefore, any such $\mathcal{R}_i$ intersects the zero-set of f and the undecided space covers the zero-set. Formally, given a decomposition $\cup_i \mathcal{R}_i = \mathcal{P}$, the undecided region is given as:

$$\mathcal{U} = \cup_i \mathcal{R}_i \text{ s.t. } \mathcal{R}_i \cap Z(f) \neq \emptyset. \tag{66}$$

Excluding the trivial case that f is identically zero ($Z(f) = \mathcal{P}$), we prove termination in two steps. First, we show that $Z(f)$ can be covered by finitely many rectangular regions whose total volume can be made arbitrarily small. Call such cover C. Second, we show that our algorithm can reach in a finite number of steps a decomposition where $\mathcal{U}$ covers C and has volume no larger than the tolerance ε. Finally, we extend the termination proof to nested CSL properties.

(1) We state now two useful properties of zero-sets: i) the zero-set of a multi-variate polynomial is negligible, i.e. it has Lebesgue measure (volume) 0 (a proof is available in [12]). ii) the zero-set $Z(f')$ of any continuous function f' is closed. This follows from the fact that $Z(f')$ is the pre-image of f' on the closed set $\{0\}$. Now note that the parameter space $\mathcal{P}$ is compact (bounded and closed). It follows that $Z(f) \subset \mathcal{P}$ is compact too. $Z(f)$ meeting condition i) corresponds to saying that for each $\epsilon' > 0$ there exists a finite or countable collection $R_1, R_2, \ldots$ of (possibly overlapping) open rectangles such that

$$Z(f) \subseteq \bigcup_{k \in K} R_k \text{ and } \sum_{k \in K} \mathrm{vol}(R_k) < \epsilon' \text{ (see e.g. [8], Section 1).}$$

Finally, by compactness every open cover of $Z(f)$ has finite a subcover, i.e. there exists a finite index set $K' \subseteq K$ such that:

$$Z(f) \subseteq \bigcup_{k' \in K'} R_{k'} \subseteq \bigcup_{k \in K} R_k.$$

It follows that:

$$\mathrm{vol}\left(\bigcup_{k' \in K'} R_{k'} \right) \leq \sum_{k' \in K'} \mathrm{vol}(R_{k'}) \leq \sum_{k \in K} \mathrm{vol}(R_k) < \epsilon'.$$

The first inequality holds since rectangles in the cover can overlap. Thus, $C \overset{\text{def}}{=} \bigcup_{k' \in K'} R_{k'}$ is the required finite rectangular cover of $Z(f)$ with arbitrarily small volume. Note that these properties hold also if we replace in C each rectangle $R_{k'}$ with its closure.

(2) Since any finite union of overlapping rectangles can be rewritten as a finite union of almost disjoint rectangles (i.e. intersecting only at their extrema) [15], we rewrite the cover C as

$$C = \bigcup_{j \in J} R_j, \text{ such that } \forall i, j \in J.\ \mathsf{int}(R_i) \cap \mathsf{int}(R_j) = \emptyset$$

where $\mathsf{int}(R)$ is the interior of rectangle R. In particular, this transformation can be done in a such way that each R_j is a box of width δ, for some $\delta > 0$. We can hence derive the following:

$$|J| \cdot \delta^n = \sum_{j \in J} \mathrm{vol}(R_j) = \mathrm{vol}(C) < \epsilon' \tag{67}$$

where n is the number of dimensions/parameters.

Without loss of generality assume that the parameter space $\mathcal{P}$ is the unit cube, meaning that the algorithm terminates for $\mathrm{vol}(\mathcal{U}) \leq \varepsilon$. Assume also that at each step undecided parameter regions are decomposed by bisection. Consider a number of refinement steps i such that each undecided region at the i-th step has width $w \in [\delta, 2\delta]$, yielding $i = \lfloor -\log_2 \delta \rfloor$. From Eqn. 66 and C being a cover of $Z(f)$, we derive that:

$$\mathcal{U} \subseteq \cup_i \mathcal{R}_i \text{ s.t. } \mathcal{R}_i \cap C \neq \emptyset.$$

Observe that each rectangle in the cover C is intersected by at most 2^n undecided regions of width w. Let $N = |J|$ be the number of boxes in the cover C. Then,

$$\mathrm{vol}(\mathcal{U}) \leq 2^n \cdot N \cdot w^n \leq 2^n \cdot N \cdot (2\delta)^n < \epsilon' \cdot 4^n \tag{68}$$

where the last inequality holds by Eqn 67.

Since the bound ϵ' on the volume of C is arbitrary, termination follows. Indeed, to satisfy the termination condition ($\mathrm{vol}(\mathcal{U}) \leq \varepsilon$), we can chose $\epsilon' = \varepsilon \cdot 4^{-n}$ which implies the existence of a suitable δ and, in turn, of a finite number of steps $i = \lfloor -\log_2 \delta \rfloor$.

(3) By Theorem 1, we know that the satisfaction function of a nested CSL property is piecewise polynomial, with a finite number of (bounded) subdomains. We proceed by borrowing from steps *(1)* and *(2)* of this proof. Let D be an index set identifying the subdomains and f_d be the satisfaction function at the d-th subdomain. Then, there exist $\delta > 0$, arbitrary positive constants $\{\epsilon'_d\}_{d \in D}$ and a set of finite covers $\{C_d\}_{d \in D}$ such that for all $d \in D$, $\mathrm{vol}(C_d) < \epsilon'_d$ and C_d is composed of pairwise-disjoint boxes of width δ. We can now prove termination by showing that Eqn 68 holds also for the nested case if we set $N = \sum_{d \in D} |C_d|$ to the total number of boxes and $\epsilon' = \sum_{d \in D} \epsilon'_d$.

However, there is an important caveat to discuss. In this case, the satisfaction function might exhibit jump discontinuities characterized by coarse probability bounds that cannot be "mitigated" by the iterative refinement procedure. It follows that we need to take into account all the additional undecided regions that contain jump discontinuity points. Fortunately, since piecewise continuous functions are continuous almost everywhere, the set of such discontinuities has measure 0, and hence, by a similar argument to the above proof, the total volume of the undecided regions containing discontinuities can be made arbitrarily small in a finite number of steps. □

A.4 Proposition 4

We prove the proposition by first showing that the safe bounds can be made arbitrarily small in a finite number of steps. Second, we derive the number of steps for which the termination condition is met.

(1) Define $f = \hat{\Lambda}_\phi(\cdot)(s_0)$ and let D^k be the set of regions at the k-th decomposition step. Fix a region $\mathcal{R}_j^k \in D^k$. Let $[f(\mathcal{R}_j^k)]$ be the interval describing the range of f over $\mathcal{R}_j^k$. Since f is a polynomial function over a bounded domain, then it is also Lipschitz continuous, implying that there exists a constant m_j^k s.t. $\mathbf{w}([f(\mathcal{R}_j^k)]) \leq m_j^k \cdot \mathbf{w}(\mathcal{R}_j^k)$, where $\mathbf{w}$ denotes the width of the rectangle. The safe approximation we compute is such that $[f(\mathcal{R}_j^k)] \subset [\Lambda_{\min}^{\mathcal{R}_j^k}, \Lambda_{\max}^{\mathcal{R}_j^k}]$ and, in particular, $\Lambda_{\max}^{\mathcal{R}_j^k} - \Lambda_{\min}^{\mathcal{R}_j^k} = \mathbf{w}([f(\mathcal{R}_j^k)]) + e^{\top} + e^{\bot}$, where $e^{\top}$ and $e^{\bot}$ are the global approximation errors for the upper and lower bound of $\hat{\Lambda}_\phi$.

We now derive a closed-form expression for $e^{\top}$, based on Proposition 2. Let $e^{\top} = \overline{g}_{k_\epsilon}$, where k_ϵ is the number of uniformisation steps and, by Eqn 42, there exist positive constants M_1 and M_2 such that

$$\overline{g}_{k_\epsilon} = \overline{g}_{k_\epsilon - 1} \cdot \left(1 + \frac{M_2}{q}\right) + \frac{M_1}{q}(\mathbf{w}(\mathcal{R}_j^k)). \tag{69}$$

Let $c_2 = 1 + \frac{M_2}{q}$ and $c_1 = \frac{M_1}{q}$. Solving the recurrence at Eqn 69, we get the following expression:

$$\overline{g}_{k_\epsilon} = c_j^{k,\top} \cdot \mathbf{w}(\mathcal{R}_j^k)) \tag{70}$$

where $c_j^{k,\top} = \dfrac{c_1 \cdot \left(c_2^{k_\epsilon} - 1\right)}{c_2 - 1}$. Following the same derivation for $e^{\bot}$, we conclude that there exist positive constants $c_j^{k,\top}$ and $c_j^{k,\bot}$ such that $e^{\top} = c_j^{k,\top} \cdot \mathbf{w}(\mathcal{R}_j^k)$ and $e^{\bot} = c_j^{k,\bot} \cdot \mathbf{w}(\mathcal{R}_j^k)$. Then, for all $\mathcal{R}_j^k \in D^k$:

$$\Lambda_{\max}^{\mathcal{R}_j^k} - \Lambda_{\min}^{\mathcal{R}_j^k} \leq (m + c^{\top} + c^{\bot}) \cdot \mathbf{w}(\mathcal{R}_j^k) \tag{71}$$

where $m = \max\{m_j^k \mid \mathcal{R}_j^k \in D_k\}$, $c^{\top} = \max\{c_j^{k,\top} \mid \mathcal{R}_j^k \in D_k\}$ and $c^{\bot} = \max\{c_j^{k,\bot} \mid \mathcal{R}_j^k \in D_k\}$. Since regions are decomposed by sectioning at the mid-point of each parameter interval, the width of a region is halved at every step, so the equation is expressed as:

$$\Lambda_{\max}^{\mathcal{R}_j^k} - \Lambda_{\min}^{\mathcal{R}_j^k} \leq (m + c^{\top} + c^{\bot}) \cdot \frac{\mathbf{w}(\mathcal{P})}{2^k}. \tag{72}$$

Then, for an arbitrary precision ϵ', there exists a finite number of decomposition steps k such that $\Lambda_{\max}^{\mathcal{R}_j^k} - \Lambda_{\min}^{\mathcal{R}_j^k} \leq \epsilon'$ for all $\mathcal{R}_j^k \in D^k$.

(2) Now we show how to determine ϵ' s.t. the termination condition $\Lambda_\phi^{\top} - \Lambda_\phi^{\bot} \leq \epsilon$ is met. Let $\mathcal{R}_{\top}^k$ be one of the regions with the highest upper probability bound, thus $\Lambda_\phi^{\top} = \Lambda_{\max}^{\mathcal{R}_{\top}^k}$. Note that the highest lower bound M is at least $\Lambda_{\min}^{\mathcal{R}_{\top}^k}$, and thus every region $\mathcal{R}_j^k$ in $\mathcal{T}$ is such that

$$\Lambda_{\max}^{\mathcal{R}_j^k} \geq M \geq \Lambda_{\min}^{\mathcal{R}_{\top}^k} \geq \Lambda_{\max}^{\mathcal{R}_{\top}^k} - \epsilon' = \Lambda_\phi^{\top} - \epsilon'.$$

Then, the smallest lower bound in the true region, $\Lambda_\phi^{\bot}$, is at least $M - \epsilon'$ and so, $\Lambda_\phi^{\bot} \geq \Lambda_\phi^{\top} - 2 \cdot \epsilon'$. This implies that termination is achieved with $\epsilon' = \frac{\epsilon}{2}$ and, according to Eqn 72, in a number of steps equal to:

$$k = \left\lceil \log_2 \left(\frac{2 \cdot (m + c^{\top} + c^{\bot}) \cdot \mathbf{w}(\mathcal{P})}{\epsilon} \right) \right\rceil.$$

□