Parameter Synthesis and Robustness Analysis of Rule-Based Models[⋆]

Matej Troják ✉, David Šafránek, Lukrécia Mertová, and Luboš Brim

Systems Biology Laboratory, Masaryk University, Brno, Czech Republic
✉ xtrojak@fi.muni.cz

Abstract. We introduce the Quantitative Biochemical Space Language, a rule-based language for a compact modelling of probabilistic behaviour of complex parameter-dependent biological systems. Application of rules is governed by an associated parametrised rate function, expressing partially known information about the behaviour of the modelled system. The parameter values influence the behaviour of the model. We propose a formal verification-based method for the synthesis of parameter values (parameter synthesis) which ensure the behaviour of the modelled system satisfies a given PCTL property. In addition, we demonstrate how this method can be used for robustness analysis.

1 Introduction

In systems biology, models of biological processes have to reflect several levels of abstraction adapted accordingly to the known information. At every level, the system has to be described rigorously in a formal language to avoid misunderstood and ambiguous interpretations.

Rule-based languages represent an intuitive and convenient modelling tool for biologists because the dynamics of biochemical systems is typically determined by the underlying causal rules. Existing rule-based languages focus on specific features such as structures binding [15,19], regulatory interactions [41], modularity [39], or spatial aspects [24]. However, a challenge is to combine suitable levels of abstraction (ranging from qualitative to quantitative aspects) with the compactness of the description while not compromising human readability. To that end, we have introduced Biochemical Space Language (BCSL) [42], a high-level rule-based language that combines several features of rule-based frameworks in a single formalism.

BCSL design stems from a long-time practical experience with describing biochemical processes rigorously but still in a way that is understandable by the users (biologists in this case). The central goal is to describe the biochemistry of a given process at the mechanistic level, in our words, to build the so-called biochemical space of the given process. Biochemical space plays a central role in the platform we are developing for modelling, specification, and analysis of biological processes [43]. In this context, the rule-based description in BCSL serves

[⋆] This work has been supported by the Czech Science Foundation grant 18-00178S.

as a bridge between the explicit biological knowledge and mathematical models that typically encrypt the information in non-trivial chains of approximations. It is worth noting that rule-based description of (bio)chemical processes is the essence of chemistry and hence the rule-based view is natural to systems modellers [38]. BCSL has been successfully used in the international consortium for cyanobacteria modelling and analysis [43].

Apparently, by building the biochemical space in a rule-based language, we obtain an executable alternative to the existing mathematical models [20]. In particular, the long-term goal is to use the biochemical space as an integrated model of the given biological problem. To fulfil this goal, the language has to support *quantitative aspects* of the rules, e.g., the rate of performing the rule. Such quantitative aspects have been addressed in Kappa [15], BNGL [19], and BIOCHAM [11], a more general framework has been introduced in Chromar [24], and in the process-algebraic approach of BioPEPA [13]. However, quantitative aspects have not yet been addressed in BCSL. Due to the specific level of abstraction considered in BCSL, it is not possible to directly adapt the solutions employed in above-mentioned languages.

In this paper, we introduce the *quantitative BCSL* (qBCSL) by extending BCSL with quantitative dynamical aspects. This is realised by associating rules with parametrised *rate functions* of the current state of the system dynamics. The intended meaning is to quantify the rate of the particular interaction. A model with rate-assigned rules gives rise to probabilistic semantics which is expressed by means of parametric Markov Chains (pMC) [16,33] representing the family of Discrete Time Markov Chains (DTMC) for all admissible settings of parameters (*parametrisations*) appearing in rate functions. Based on the tool Storm [18] we establish a framework for (exact) *parameter synthesis* of qBCSL models with respect to PCTL [23] properties. Technically, the method computes a rational function that assigns the probability of satisfying a given property to each parametrisation. Note that the stochastic semantics of a rule-based model is traditionally formalised as a Continuous Time Markov Chain (CTMC) [21]. However, the scalalbility of existing exact methods [2,10] is limited to small models, other available methods are just simulation-based, thus providing only approximative results. Following the idea of using approximate models with discrete-time semantics [3], we consider the DTMC that provides efficient methods to analyse exact probability of PCTL properties.

In addition, we provide an approach for *global robustness analysis* of qBCSL models with respect to a given parameter perturbation and a PCTL formula. Global robustness characterises the mean validity of a formula over all parameter values in the given perturbation set [5,12]. The entire framework is implemented in the open source tool eBCSgen[1] and demonstrated on a biological case study.

The primary contribution of this paper is in bringing the exact parameter synthesis into the field of rule-based models with stochastic semantics. The uniqueness of our solution is not only in the level of abstraction qBCSL provides but also in the fact that we directly interpret rule-based models by means

[1] https://github.com/sybila/eBCSgen

of DTMCs to support formal analysis. Such a setting allows to apply efficient parameter synthesis techniques [16,17,22] to rule-based models.

1.1 Related Work

In qBCSL, objects are projected into multisets that represent the model states. The stochastic multiset rewriting (SMR) was used in [3] to encode expressive process calculi such as π-calculus. In [4], SMR was used as a base for parametrised mass-action reaction-based models encoded by means of interval Markov Chains (iMC) where parameters range over closed intervals. Given the fixed structure of mass-action kinetics and intervals of kinetic parameters values, they compute lower and upper bounds for reachability probability. In our case, we support rational parametrised kinetic functions and employ parameter synthesis techniques giving a symbolic function representing the exact parameter sets.

Methods for parameter synthesis of pMCs have been introduced with symbolic computation of reachability properties through state elimination [16,22,27], recently improved by parameter lifting [40] and fraction-free Gaussian elimination [26]. Here we employ these techniques as implemented in Storm tool.

An alternative approach to the analysis of complex stochastic models under parameter uncertainty is based on statistical methods [1,8,9,35]. There are only a few works that bridge the rule-based framework to such techniques. In [34], a statistical parameter sampling method is employed to analyse unknown parameters in BNGL models represented by means of CTMCs where the rate function is limited to mass action kinetics. The work [28] employes statistical model checking for parameter synthesis of CTMCs. The recent work [30] combines statistical model checking with machine learning techniques to calibration (estimation) of parameters in order to maximise the probability of satisfying a given specification. In [6], the authors adapt simulation-based and moment-based methods. In general, statistical techniques do not give an exact symbolic representation of satisfying parameter sets.

2 Preliminaries

Throughout this section, we consider a given set of atomic propositions AP. *Discrete Time Markov Chain* (DTMC) is a tuple $(\mathcal{S}, s_0, \rho, L)$ where $\mathcal{S}$ is the set of *states*, $s_0 \in \mathcal{S}$ is the *initial state*, $\rho : \mathcal{S} \times \mathcal{S} \to [0, 1]$ is the *transition probability matrix*, where for all $s \in \mathcal{S}$ we require that $\sum_{s' \in \mathcal{S}} \rho(s, s') = 1$, and $L : \mathcal{S} \to 2^{\mathsf{AP}}$ is a *labelling function* which gives the atomic propositions that are true in a state.

The matrix entry $\rho(s, s')$ gives the probability of making a transition from s to s'. The probability of following a finite path $s_0 s_1 \ldots s_n$ is $\rho(s_0, s_1) \cdot \rho(s_1, s_2) \cdot \ldots \cdot \rho(s_{n-1}, s_n)$. These probabilities for finite paths give rise to a unique probability measure $\mathbf{Pr}_s$ on the set $Path_s$ of infinite paths starting in state s defined on the sets of paths having a finite common prefix, such that

$$\mathbf{Pr}(\{\omega \mid \omega = s s_1 \ldots s_n . \omega'\}) = \rho(s, s_1) \cdot \rho(s_1, s_2) \cdot \ldots \cdot \rho(s_{n-1}, s_n).$$

The logic PCTL [23] is a probabilistic variant of CTL where the existential and the universal quantification over paths in CTL is replaced with a *probabilistic operator* $\Pi_{\bowtie\varrho}(\cdot)$, where $\bowtie \in \{\leq, <, >, \geq\}$ and $\varrho \in [0,1]$ is the *probability threshold*, that can be applied to a path formula. The formal syntax of PCTL formulae is given by the following grammar:

$$\begin{aligned}\phi &::= \mathsf{True} \mid a \in \mathsf{AP} \mid \phi \wedge \phi \mid \neg\phi \mid \Pi_{\bowtie\mathsf{n}}(\psi)\\ \psi &::= \mathbf{X}\phi \mid \phi\mathbf{U}\phi\end{aligned}$$

The semantics of PCTL is the same as that of CTL [14] for the fragment where they both coincide. The semantics of the probabilistic operator is:

$$s \models \Pi_{\bowtie\mathsf{n}}(\psi) \quad \mathsf{iff} \quad \mathbf{Pr}_s(\{\omega \in Path_s \mid \omega \models \psi\}) \bowtie \mathsf{n}$$

meaning that the probability measure of the set of paths satisfying ψ is calculated and compared to the threshold n, yielding true or false.

The standard qualitative model checking algorithm proceeds in the same way as for CTL, by induction on ϕ. In [16], a symbolic approach was proposed. It is based on derivation of a finite state automaton (FSA) $\mathcal{A} = (\mathcal{S}, \Sigma, \delta, \mathcal{S}_f)$ from given DTMC. $\mathcal{S}$ is the same set of states as in the DTMC, the alphabet Σ consists of the strictly positive entries of the probability matrix, the set of final states $\mathcal{S}_f$ and the transition function δ depend on the path formula under consideration.

The regular language $\mathcal{L}(\mathcal{A}, s)$ recognized by $\mathcal{A}$ with an initial state $s \in \mathcal{S}$, corresponds to the (possibly infinite) set Ω of finite paths from s to some final state in $\mathcal{S}_f$, following only transitions allowed by δ.

A regular expression r over an alphabet Σ is computed using the state-elimination algorithm [25]. The evaluation $\mathsf{val}(\mathsf{r})$ of the regular expression can be done by replacing union by addition, concatenation by multiplication, and star by the limit of a geometric series (for the formal definition, see [16]).

The evaluation of a regular expression r computed for a language $\mathcal{L}(\mathcal{A}, s)$ is the probability measure in s of the set of paths with prefixes in Ω:

$$\mathsf{val}(\mathsf{r}) = \mathbf{Pr}_s(\{\omega \in Path_s \mid \exists k \geq 0.\omega(k) \in \mathcal{S}_f \wedge \forall l < k, \exists a \in \Sigma.\omega(l+1) \in \delta(\omega(l), a)\})$$

The model checking problem can be then solved for a state s by evaluating a regular expression r equivalent to the language recognized by the automaton with the initial state s, i.e. $s \models \Pi_{\bowtie\mathsf{n}}(\psi)$ iff $\mathsf{val}(\mathsf{r}) \bowtie \mathsf{n}$.

We can also directly specify properties which evaluate to a numerical value – the result of *quantitative model checking*. This is achieved by replacing the probability bound from Π operator with '=?'. Note that this is only allowed when the Π in question is the outermost operator of the property. The evaluation is then given as $\Pi_{=?}(\psi) = \mathbf{Pr}_s(\{\omega \in Path_s \mid \omega \models \psi\})$ which means it can be computed using the symbolic approach as $\Pi_{=?}(\psi) = \mathsf{val}(\mathsf{r})$.

3 Quantitative Biochemical Space Language

In this section, we define quantitative Biochemical Space Language (qBCSL) with quantitative aspects, as an extension of BCSL [42]. The quantitative aspects enable to reason about the *rate* of interactions to occur. All the definitions are demonstrated in a simple example in Section 3.1.

Let $\mathbb{N}_{\mathsf{T}}$, $\mathbb{N}_{\mathsf{A}}$, $\mathbb{N}_{\mathsf{c}}$, and $\mathbb{N}_{\delta}$ be mutually exclusive finite sets of agent, atom, compartment, and feature names, respectively. The *syntax* of the qBCSL objects is given by the following grammar:

$$\begin{array}{ll|ll}
\text{multiset} & \mathsf{M} ::= \emptyset \mid \mathsf{T} \mid \mathsf{M},\mathsf{M} & \text{atomic name} & \eta ::= x \in \mathbb{N}_{\mathsf{A}} \\
\text{agent} & \mathsf{T} ::= \mu_{\mathsf{c}}^{\lambda}(\gamma) & \text{agent name} & \mu ::= x \in \mathbb{N}_{\mathsf{T}} \\
\text{composition} & \gamma ::= \emptyset \mid \mathsf{A},\gamma & \text{feature} & \delta ::= x \in \mathbb{N}_{\delta} \\
\text{atom} & \mathsf{A} ::= \eta\{\delta\} & \text{compartment} & \mathsf{c} ::= x \in \mathbb{N}_{\mathsf{c}} \\
 & & \text{complex ID} & \lambda ::= x \in \mathbb{N}
\end{array}$$

We restrict ourselves only to finite expressions and require that an atomic name occurs at most once in a composition.

We denote by $\mathbb{M}$ the set of all multisets. We assume the *structural congruence* $\equiv$ to be the least congruence on multisets satisfying axioms $\mathsf{M}_1,\mathsf{M}_2 \equiv \mathsf{M}_2,\mathsf{M}_1$ and $\mathsf{M},\emptyset \equiv \mathsf{M}$, where $\mathsf{M}_1,\mathsf{M}_2$ represents the union of multisets M_1 and M_2. Additionally, we assume a similar relation $\equiv_\gamma$ on compositions defined as the least congruence satisfying axioms $\mathsf{A},\gamma \equiv_\gamma \gamma,\mathsf{A}$ and $\emptyset,\gamma \equiv_\gamma \gamma$.

The structural congruence $\equiv$ (resp. $\equiv_\gamma$) allows us to formally define the algebraic multiset operations $\in, \subseteq, \subset, \cup, \cap$ and $\setminus$ on qBCSL terms. For example, $\mathsf{T} \in \mathsf{M}$ corresponds to $\exists \mathsf{M}' \in \mathbb{M}.\mathsf{M} \equiv \mathsf{T},\mathsf{M}'$ and $\mathsf{M} \subseteq \mathsf{M}'$ corresponds to $\exists \mathsf{M}'' \in \mathbb{M}.\mathsf{M}' \equiv \mathsf{M},\mathsf{M}''$. Moreover, by $\mathsf{M}(\mathsf{T})$ we denote the number of occurrences of agent T in the multiset M and by $\mathsf{M}(\mathsf{M}')$ the number of occurrences of multiset M' in the multiset M, which is at least one in the case $\mathsf{M} \subseteq \mathsf{M}'$ (formally, it is defined as minimal $\mathsf{M}(\mathsf{T})$ for all $\mathsf{T} \in \mathsf{M}'$).

We denote by Λ the set of all complex IDs of a multiset M. Two multisets are *equal*, $\mathsf{M}_1 = \mathsf{M}_2$, if there exists a bijective function $\mathsf{h} : \Lambda_1 \to \Lambda_2$ such that $\mathsf{M}_1^{\mathsf{h}} \equiv \mathsf{M}_2$ where $\mathsf{M}_1^{\mathsf{h}}$ denotes M_1 with every occurrence of a complex ID λ replaced by $\mathsf{h}(\lambda)$.

Agent signature $\sigma_{\mathsf{T}} : \mathbb{N}_{\mathsf{T}} \to 2^{\mathbb{N}_{\mathsf{A}}}$ is a function from an agent name to a set of atomic names. Set of possible agent signatures is denoted as Σ_{T}. *Atomic signature* $\sigma_{\mathsf{A}} : \mathbb{N}_{\mathsf{A}} \to 2^{\mathbb{N}_{\delta}}$ is a function from an atomic name to a non-empty set of feature names. Set of possible agent signatures is denoted as Σ_{A}.

Let $\mathbb{V}_{\mathsf{c}}$ and $\mathbb{V}_{\lambda}$ be mutually exclusive finite sets of the compartment and complex variables, respectively. Additionally, let $\mathbb{V}_{\delta} = \mathbb{N}_{\delta} \cup \{\varepsilon\}$ be a set of feature names extended by a special symbol ε. *Pattern* P is defined according to the same grammar as multisets with the following modifications:

$$\begin{array}{ll}
\text{feature} & \delta ::= s \in \mathbb{V}_{\delta} \\
\text{compartment} & \mathsf{c} ::= v \in \mathbb{V}_{\mathsf{c}} \cup \mathbb{N}_{\mathsf{c}} \\
\text{complex ID} & \lambda ::= l \in \mathbb{V}_{\lambda}
\end{array}$$

We denote by $\mathcal{P}$ the set of all patterns and with $\mathcal{P}_\perp$ we denote the set of all patterns restricted to $\delta \in \mathbb{N}_\delta$ and $\mathsf{c} \in \mathbb{N}_\mathsf{c}$. Note that the congruence relation defined on multisets does *not* hold in case of patterns. A pattern is *well-formed* if the atoms are alphanumerically sorted with respect to their names.

An *instantiation* is a function $\mathcal{I} : \mathbb{V}_\delta \cup \mathbb{V}_\mathsf{c} \cup \mathbb{V}_\lambda \to \mathbb{N}_\delta \cup \mathbb{N}_\mathsf{c} \cup \mathbb{N}$ such that $\mathcal{I}(s) \in \mathbb{N}_\delta$, $\mathcal{I}(v) \in \mathbb{N}_\mathsf{c}$, and $\mathcal{I}(l) \in \mathbb{N}$ for $s \in \mathbb{V}_\delta$, $v \in \mathbb{V}_\mathsf{c}$, and $l \in \mathbb{V}_\lambda$, respectively. We denote by Γ the set of all instantiations.

Given an atomic signature σ_A and a pattern $\mathsf{P} \in \mathcal{P}$, with $\mathcal{I}(\mathsf{P})$ we denote the multiset obtained by replacing each occurrence of a term ν appearing in P with the corresponding instantiation $\mathcal{I}(\nu)$ respecting the signature σ_A. Particularly, the signature σ_A restricts instantiation of each feature ε to one of the feature names defined for the appropriate atomic name. Please note that the same term repeating on separate positions in the pattern can be instantiated to different values.

Given two finite patterns $\mathsf{P} = \mathsf{T}_1, \mathsf{T}_2, \dots, \mathsf{T}_n$ and $\mathsf{P}' = \mathsf{T}'_1, \mathsf{T}'_2, \dots, \mathsf{T}'_m$, instantiations $\mathcal{I}, \mathcal{I}' \in \Gamma$ are *consistent* with respect to the given patterns P, P', written $\mathcal{I}(\mathsf{P}) \varDelta \mathcal{I}'(\mathsf{P}')$, if $\forall i \in [1, \mathsf{min}(m, n)]$ the following conditions hold:

1. $\lambda(\mathsf{T}_i) = \lambda(\mathsf{T}'_i) \Rightarrow \lambda(\mathcal{I}(\mathsf{T}_i)) = \lambda(\mathcal{I}'(\mathsf{T}'_i))$
2. $\mathsf{c}(\mathsf{T}_i) = \mathsf{c}(\mathsf{T}'_i) \Rightarrow \mathsf{c}(\mathcal{I}(\mathsf{T}_i)) = \mathsf{c}(\mathcal{I}'(\mathsf{T}'_i))$
3. $\mathsf{A}_k(\mathsf{T}_i) = \mathsf{A}_k(\mathsf{T}'_i) \Rightarrow \mathsf{A}_k(\mathcal{I}(\mathsf{T}_i)) = \mathsf{A}_k(\mathcal{I}'(\mathsf{T}'_i))$

where $\lambda(\mathsf{T})$ denotes the complex ID λ of the agent T, $\mathsf{c}(\mathsf{T})$ denotes compartment c of the agent T, and $\mathsf{A}_k(\mathsf{T})$ denotes the atom from the composition γ of agent T on a position k.

Pattern expansion is a function $\langle \_ \rangle : \mathcal{P} \times \Sigma_\mathsf{T} \to \mathcal{P}$ which extends a given pattern P to a pattern $\langle \mathsf{P} \rangle$ such that every occurrence of a composition γ of an agent T is extended by atoms whose names are not yet present in γ and are defined in the given signature $\sigma_\mathsf{T} \in \Sigma_\mathsf{T}$. These newly added atoms have assigned feature ε and are inserted to the composition in such way that they preserve the alphanumerical order.

Let $\mathbb{V}$ be a set of parameters. For each parameter v, a domain of admissible positive values is assigned, denoted by $\mathcal{D}(\mathsf{v}) \in 2^{\mathbb{R}^+}$. In the following, we define the grammar for the algebraic rational *rate expression* f:

$$\begin{array}{ll} \text{rate expression} & \mathsf{f} ::= \frac{\mathsf{g}}{\mathsf{g}} \mid \mathsf{g} \\ \text{polynomial expression} & \mathsf{g} ::= \mathsf{c} \mid \mathsf{v} \mid [\ \mathsf{t}\ ] \mid \mathsf{g} + \mathsf{g} \mid \mathsf{g} \times \mathsf{g} \mid \mathsf{g}^\mathsf{n} \end{array}$$

where $\mathsf{c} \in \mathbb{R}$ is a *constant*, $\mathsf{v} \in \mathbb{V}$ is a *parameter*, $\mathsf{n} \in \mathbb{N}$ is an *exponent*, and $\mathsf{t} \in \mathcal{P}$ is a *pattern* such that all agents have the same complex ID λ.

We denote by $\mathbb{F}$ the set of all rate expressions and with $\mathbb{F}_\mathsf{v}$ rate expressions without the patterns (note that $\mathbb{F}_\mathsf{v} \subseteq \mathbb{F}$). For the sake of readability, we allow additional simplifications (e.g. parentheses) which can always be converted to a form given by the provided grammar.

Multiset evaluation $\mathbb{F} \times \mathbb{M} \to \mathbb{F}_\mathsf{v}$ of a rate expression f on a multiset M, written $\mathsf{f}(\mathsf{M})$, is a rate expression $\mathsf{f}' \in \mathbb{F}_\mathsf{v}$ such that each pattern $[\ \mathsf{t}\ ]$ is replaced by an integer $\sum_{\mathcal{I} \in \Gamma} \mathsf{M}(\mathcal{I}\langle \mathsf{t} \rangle)$ expressing the sum of all possible instantiations of the pattern. Note that number of possible instantiations Γ is finite with respect to the set of all complex IDs Λ of multiset M.

Rewrite rule R is a triple $(\mathsf{P}_l, \mathsf{P}_r, \mathsf{f}) \in \mathcal{P} \times \mathcal{P} \times \mathbb{F}$, usually written as $\mathsf{P}_l \xrightarrow{\mathsf{f}} \mathsf{P}_r$. It describes a structural change of a multiset defined by the difference between left-hand and right-hand side patterns, associated with the rate expression f.

A qBCSL *model* $\mathcal{M}$ is a tuple $(\mathcal{R}, \sigma_\mathsf{T}, \sigma_\mathsf{A}, \mathsf{M}_0, \mathsf{V})$ such that $\mathcal{R}$ is a finite set of rewrite rules, $\sigma_\mathsf{T} \in \Sigma_\mathsf{T}$ is an agent signature, $\sigma_\mathsf{A} \in \Sigma_\mathsf{A}$ is an atomic signature, $\mathsf{M}_0 \in \mathbb{M}$ is an initial multiset, and $\mathsf{V} \in \mathbb{V}$ is a set of parameters.

3.1 Example

We provide an example consisting of a fragment of photosynthesis processes of cyanobacteria. Note that this fragment is not accurate and its purpose is to demonstrate all the formal aspects of the language only.

Let $\mathsf{ps}^1_\mathsf{tlm}(\mathsf{p700}\{\mathsf{n}\}, \mathsf{a}\{\mathsf{n}\}, \mathsf{achl}\{*\})$ denote an *agent* – photosystem of cyanobacteria – in thylakoid membrane compartment (tlm) with three active domains represented as atoms: photosystem reaction center $\mathsf{p700}$, primary acceptor of photosystem a (both in neutral state n), and chlorophyll antenna achl in excited state $*$.

Next, let us have an *agent signature* $\sigma_\mathsf{T} = \{\mathsf{ps} \to \{\mathsf{p700}, \mathsf{a}, \mathsf{achl}\}\}$, which defines allowed set of atoms for the photosystem. Note that each atomic name defined in the agent signature for an agent has to be used in its composition. An *atomic signature* $\sigma_\mathsf{A} = \{\mathsf{p700} \to \{\mathsf{n}, +\}, \mathsf{a} \to \{\mathsf{n}, -\}, \mathsf{achl} \to \{\mathsf{n}, *, +\}\}$ defines allowed states for reaction center $\mathsf{p700}$, acceptor a, and antenna achl.

We can use a *pattern* $\mathsf{P} = \mathsf{ps}^\mathsf{x}_\mathsf{tlm}(\mathsf{p700}\{\mathsf{n}\}, \mathsf{a}\{\varepsilon\})$ to describe the photosystem such that its affiliation to a particular complex is not given, only identified by a variable x. The state of $\mathsf{p700}$ is specified as neutral while for the acceptor a it is unknown (denoted with ε). Additionally, note that not every atom from the signature has to be specified (achl is omitted), which is the key aspect for compactness of the rule-based approach.

Such pattern can be *instantiated* by function $\mathcal{I} = \{\mathsf{x} \to 1, \varepsilon \to -\}$ which assigns to each unspecified element of a pattern a particular value. Applying the instantiation on the pattern P, we obtain $\mathcal{I}(\mathsf{P}) = \mathsf{ps}^1_\mathsf{tlm}(\mathsf{p700}\{\mathsf{n}\}, \mathsf{a}\{-\})$.

However, the achl atom is missing in the composition. For this purpose, the *pattern expansion* is defined, which, when applied on a pattern, creates the expanded pattern $\langle \mathsf{P} \rangle = \mathsf{ps}^\mathsf{x}_\mathsf{tlm}(\mathsf{p700}\{\mathsf{n}\}, \mathsf{a}\{\varepsilon_1\}, \mathsf{achl}\{\varepsilon_2\})$. Given the instantiation function $\mathcal{I} = \{\mathsf{x} \to 1, \varepsilon_1 \to -, \varepsilon_2 \to +\}$, the instantiation of expanded pattern is $\mathcal{I}\langle \mathsf{P} \rangle = \mathsf{ps}^1_\mathsf{tlm}(\mathsf{p700}\{\mathsf{n}\}, \mathsf{a}\{-\}, \mathsf{achl}\{+\})$.

$$\mathsf{ps}^\mathsf{x}_\mathsf{tlm}(\mathsf{p700}\{\mathsf{n}\}, \mathsf{achl}\{+\}) \xrightarrow{k_1 \times [\mathsf{ps}^\mathsf{x}_\mathsf{tlm}(\mathsf{p700}\{\mathsf{n}\}, \mathsf{achl}\{+\})]} \mathsf{ps}^\mathsf{x}_\mathsf{tlm}(\mathsf{p700}\{+\}, \mathsf{achl}\{\mathsf{n}\}) \quad (1)$$

The *rule* 1 represents a reduction of oxidized primary electron donor in photosystem. It describes a change of states of $\mathsf{p700}$ and achl regardless of the state of acceptor a. Complex variable x ensures that the complex ID of the agent does not change. The rate expression is dependent on the number of occurrences of the pattern in a given multiset and a parameter $k_1 \in [5, 10]$ representing admissible values for *mass action* law constant.

$$\mathsf{ps}^{\mathsf{x}}_{\mathsf{tlm}}(\emptyset), \mathsf{ps}^{\mathsf{y}}_{\mathsf{tlm}}(\emptyset) \xrightarrow{k_2 \times [\mathsf{ps}^{\mathsf{x}}_{\mathsf{tlm}}(\emptyset)] \times ([\mathsf{ps}^{\mathsf{y}}_{\mathsf{tlm}}(\emptyset)] - 1)} \mathsf{ps}^{\mathsf{x}}_{\mathsf{tlm}}(\emptyset), \mathsf{ps}^{\mathsf{x}}_{\mathsf{tlm}}(\emptyset) \qquad (2)$$

The rule 2 describes a formation of a complex from two ps agents. The formation is independent of the particular conformation of compositions of the agents (represented by $\emptyset$). Similar to the previous rule, the rate is dependent on the number of occurrences and a parameter $k_2 \in [0, 2]$ representing admissible values for mass action law constant.

3.2 Semantics

The semantics for the qBCSL is given in two steps – (1) we construct a parametric Quantitative Labelled Transition System (pQLTS) by transitive *rewriting* of multisets with rules such that nodes represent multisets, transitions applied rules, and quantitative labels evaluated rate expressions; and (2) we create parametric DTMC (pMC) from pQLTS such that labels of outgoing edges for each state are normalised to probability functions of parameters.

Let $\mathcal{M} = (\mathcal{R}, \sigma_{\mathsf{T}}, \sigma_{\mathsf{A}}, \mathsf{M}_0, \mathsf{V})$ be a qBCSL model. The *rewriting* of the multisets is given by labelled transition relation $\mathsf{M}_1 \xrightarrow{\mathsf{f}'} \mathsf{M}_2$ with $\mathsf{f}' \in \mathbb{F}_{\mathsf{V}}$ and $\mathsf{M}_1, \mathsf{M}_2 \in \mathbb{M}$ satisfying the following inference rule:

$$\frac{\begin{array}{c} \mathsf{R} : \mathsf{P}_l \xrightarrow{\mathsf{f}} \mathsf{P}_r \qquad \mathsf{M}_s = \mathsf{M}_t \\ \exists\, \mathcal{I}, \mathcal{I}' \in \mathit{\Gamma}.\ \mathcal{I}\langle \mathsf{P}_l \rangle = \mathsf{M}_l \wedge \mathcal{I}'\langle \mathsf{P}_r \rangle = \mathsf{M}_r \\ \mathcal{I}\langle \mathsf{P}_l \rangle \mathit{\Delta} \mathcal{I}'\langle \mathsf{P}_r \rangle \\ \mathsf{Unique}(\mathsf{M}_s; \mathsf{M}_l) \wedge \mathsf{Unique}(\mathsf{M}_t; \mathsf{M}_r) \end{array}}{\mathsf{M}_s, \mathsf{M}_l \xrightarrow{\mathsf{f}(\mathsf{M}_s, \mathsf{M}_l)} \mathsf{M}_t, \mathsf{M}_r}$$

It is possible to consider multiset rewriting which is context-free in terms of complex manipulations. It enables so-called *side effects* – modifications beyond the scope of the rule (e.g. synthesis of a new agent with an already existing complex ID). In order to avoid these side effects, we define predicate $\mathsf{Unique}(\mathsf{M}_1; \mathsf{M}_2)$ which holds if $\forall (\mathsf{T}_1, \mathsf{T}_2) \in \mathsf{M}_1 \times \mathsf{M}_2 . \lambda(\mathsf{T}_1) \neq \lambda(\mathsf{T}_2)$ for some $\mathsf{M}_1, \mathsf{M}_2 \in \mathbb{M}$. This predicate is used in conditions of inference rule of labelled transition relation, which ensures that if the rule is modifying a complex, it is modifying it as a whole and if the rule is creating a new complex, it has a unique identifier across the newly created multiset. An indirect consequence of disabled side effects is that the number of encoded particular agents of a model is finite.

We define *parametric Quantitative Labelled Transition System* pQLTS as a triple $(\mathcal{S}, \mathcal{L}, \mapsto)$ where each transition corresponds to the application of a rewrite rule. For a model, it is obtained by transitive closure of inference rule starting from M_0. The label $\ell \in \mathcal{L}$ of a transition is an evaluated rate expression of the applied rule. We denote by $\ell(s, s')$ the label of transition $t(s, s') \in \mapsto$.

Parametric Markov chain pMC is a tuple $(\mathcal{S}, s_0, \rho', \mathsf{V}, L)$ where $\mathcal{S}$ is a finite set of *states*, $s_0 \in \mathcal{S}$ is the *initial state*, $\rho' : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{F}_{\mathsf{V}}$ is the *parametric* *transition probability matrix*, V is a finite set of parameters, and $L : \mathcal{S} \to 2^{\mathsf{AP}}$ is a labelling function which gives the atomic propositions that are true in a state.

We consider a given set of atomic propositions AP which are expressions over the set of patterns $\mathcal{P}_\perp$ of type $[\mathsf{a}] \bowtie \mathsf{n}$ where $\mathsf{a} \in \mathcal{P}_\perp$, $\bowtie \in \{\leq, <, >, \geq\}$, and $\mathsf{n} \in \mathbb{N}$. Moreover, Boolean combinations of such expressions are also allowed.

We define the *probabilistic semantics* of a qBCSL model using a translation from its pQLTS into a pMC. We have to calculate, for each states s and s' of pQLTS, the probability of moving from s to s', by exploiting rate functions. We define a function $\vartheta : \mathcal{S} \to \mathbb{F}_{\mathsf{v}}$ where

$$\vartheta(s) = \textstyle\sum_{s' \in \mathcal{S}} \ell(s, s')$$

such that by default if $t(s, s') \notin \mapsto$ then $\ell(s, s') = 0$.

We derive a pMC $(\mathcal{S}, s_0, \rho', \mathsf{V})$ from a pQLTS $(\mathcal{S}, \mathcal{L}, \mapsto)$ by computing *parametric transition probability matrix* $\rho' : \mathcal{S} \times \mathcal{S} \to \mathbb{F}_{\mathsf{v}}$ such that $\forall s, s' \in \mathcal{S}. s \neq s'$ holds that if $\vartheta(s) = 0$ then $\rho'(s, s') = 0$ and $\rho'(s, s) = 1$; $\rho'(s, s') = \ell(s, s')/\vartheta(s)$ otherwise. Moreover, V is set of all parameters used in the rate expression in $\mathcal{L}$.

Given the set of parameters V and a domain $\mathcal{D}(\mathsf{v})$ for each parameter $\mathsf{v} \in \mathsf{V}$, the *parameter space* $\boldsymbol{P}$ induced by the set of parameters V is defined as the Cartesian product of individual parameter domains $\boldsymbol{P} = \bigtimes_{\mathsf{v} \in \mathsf{V}} \mathcal{D}(\mathsf{v})$. A *parametrisation* $\mathsf{p} \in \boldsymbol{P}$ is a $|\mathsf{V}|$-tuple holding a single value for each parameter, i.e. $\mathsf{p} = (\mathsf{v}_{1_\mathsf{p}}, \ldots, \mathsf{v}_{|\mathsf{V}|_\mathsf{p}})$, assuming an arbitrary ordering on parameters.

For a pMC C, the set of DMTCs induced by the parameter space $\boldsymbol{P}$ is defined as $\mathcal{C} = \{\mathsf{C}_\mathsf{p} \mid \mathsf{p} \in \boldsymbol{P}\}$. For each C_p, all parameters in the probability matrix are instantiated to respective components of p. A DTMC C_p is *well-defined* iff $\rho(s, s') \in [0, 1]$ for all $s, s' \in \mathcal{S}$ and $\sum_{s' \in \mathcal{S}} \rho(s, s') = 1$ for all $s \in \mathcal{S}$. For every pMC C we assume the set $\mathcal{C}$ contains only well-defined DTMCs.

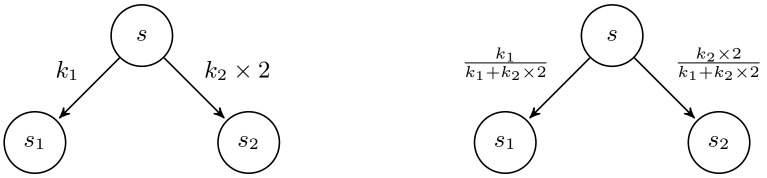
s
k_1
$k_2 \times 2$
s_1
s_2
s
$\frac{k_1}{k_1 + k_2 \times 2}$
$\frac{k_2 \times 2}{k_1 + k_2 \times 2}$
s_1
s_2

Fig. 1. *(left)* A state s of pQLTS with all its outgoing edges, labelled with appropriate multiset evaluation of rate function – state s_1 was created by applying the rule 1 and s_2 by applying the rule 2. *(right)* A pMC constructed from the pQLTS on the right such that the labels of both its outgoing edges are computed as the appropriate label from pQLTS divided by the sum of all outgoing labels, which is $k_1 + k_2 \times 2$. This, in general, ensures the sum of all labels of outgoing edges for a state is always 1.

3.3 Example (continued)

Let $\mathsf{M} = \mathsf{ps}^1_{\mathsf{tlm}}(\mathsf{p700}\{\mathsf{n}\}, \mathsf{a}\{-\}, \mathsf{achl}\{+\}), \mathsf{ps}^2_{\mathsf{tlm}}(\mathsf{p700}\{\mathsf{n}\}, \mathsf{a}\{\mathsf{n}\}, \mathsf{achl}\{*\})$ be a multiset consisting of two ps agents differing in the state of their atoms a and achl, and their complex ID. We show how application of two rules from the previous example modify the multiset M.

Applying the rule 1 changes states of the first ps agent and creates a multiset $\mathsf{M}_1 = \mathsf{ps}^1_{\mathsf{tlm}}(\mathsf{p700}\{+\}, \mathsf{a}\{-\}, \mathsf{achl}\{\mathsf{n}\}), \mathsf{ps}^2_{\mathsf{tlm}}(\mathsf{p700}\{\mathsf{n}\}, \mathsf{a}\{\mathsf{n}\}, \mathsf{achl}\{*\})$. The label of the transition is multiset evaluation of rate function, $\mathsf{f}(\mathsf{M})$, which is $k_1 \times 1$. $([\mathsf{ps}^{\mathsf{x}}_{\mathsf{tlm}}(\mathsf{p700}\{\mathsf{n}\}, \mathsf{achl}\{+\})] = 1)$.

Applying the rule 2 forms a complex from both agents and creates a multiset $\mathsf{M}_2 = \mathsf{ps}^1_{\mathsf{tlm}}(\mathsf{p700}\{\mathsf{n}\}, \mathsf{a}\{-\}, \mathsf{achl}\{+\}), \mathsf{ps}^1_{\mathsf{tlm}}(\mathsf{p700}\{\mathsf{n}\}, \mathsf{a}\{\mathsf{n}\}, \mathsf{achl}\{*\})$. The label of the transition is multiset evaluation of rate function, $\mathsf{f}(\mathsf{M})$, which is $k_2 \times 2 \times 1$. $([\mathsf{ps}^{\mathsf{x}}_{\mathsf{tlm}}(\emptyset)] = 2$, $[\mathsf{ps}^{\mathsf{y}}_{\mathsf{tlm}}(\emptyset)] = 2)$. Please note the instantiation of variables in rate functions is independent on the instantiation of left-hand side of the rule.

Both applications give rise to a simple pQLTS, from which a pMC can be constructed (Fig. 1).

4 Model Analysis

We now provide algorithms for parameter synthesis and robustness problems for qBCSL models. Both algorithms are done semi-symbolically.

4.1 Parameter synthesis

Given a qBCSL model $\mathcal{M} = (\mathcal{R}, \sigma_{\mathsf{T}}, \sigma_{\mathsf{A}}, \mathsf{M}_0, \mathsf{V})$ and a PCTL formula ϕ, the *problem of parameter synthesis* is to compute a partitioning of parameter space into three disjoint subsets: TRUE – the model satisfies the property, FALSE – the model does *not* satisfy the property, and $\mathsf{UNKNOWN}$ – the result is not known.

We solve this problem in three steps – (1) we construct pQLTS for the given qBCSL model by transitive closure of inference rule starting from initial state; (2) we derive a pMC from the pQLTS by computing parametric transition probability matrix as a normalisation of the label for all outgoing edges for every state; (3) we apply a method introduced in [16] and elaborated in [22], which is very similar to the model checking of DTMC outlined in preliminaries.

The Finite State Automaton for a pMC and a path formula is derived as in the non-parametric case. The regular expression is also evaluated recursively. Operators of union, concatenation, and star on regular expressions, are replaced by addition, multiplication, and inversion for *rational functions* respectively. Thus, by evaluating the corresponding regular expression, we obtain an algebraic expression of the probability measure of the sets of paths satisfying a path formula, as a rational function of parametrisations. We can use the result to check whether the system satisfies a formula for different values of the parameters, without having to model check the system for any given parametrisation.

This method is applicable to formulas without nested probabilistic operators only, but this does not represent a strong restriction in practice because such formulas are usually not needed to specify the properties of interest.

The computed rational function is used in parameter space exploration. An SMT solver (e.g., Z3) can be used to determine whether there exists a parametrisation inside the candidate region of the parameter space whose corresponding instantiated DTMC exceeds a given threshold on the probability.

The general approach is to maintain a set $\mathsf{UNKNOWN}$ of regions for which the result is still unknown. Initially, it is represented as the whole parameter space $\boldsymbol{P}$. Then, it takes a region out of this set and tries to decide its value. The answer can be definite, i.e. either the region satisfies the formula ϕ and is added to set TRUE or it does not satisfy the formula ϕ is added to set FALSE; or the answer is uncertain and the region is split into smaller subregions. This can be recursively executed until the required precision is met (e.g., coverage of the decided area, a boundary in recursion depth).

For a PCTL formula ϕ, we additionally consider a set of atomic propositions AP' such that the expressions of type $[\mathsf{a}] \bowtie \mathsf{n}$ are extended to $\mathsf{a} \in \mathcal{P}$. These formulae allow to reason about patterns which is very natural in the rule-based setting. The semantics of the expression is:

$$s \models [\mathsf{a}] \bowtie \mathsf{n} \quad \mathsf{iff} \quad \sum_{\mathcal{I} \in \Gamma} s(\mathcal{I}\langle \mathsf{a} \rangle) \bowtie \mathsf{n}$$

In order to use the instantiation, the signatures are required. These are available in the qBCSL model.

We have implemented our approach in the prototype tool eBCSgen, which can generate explicit pMC straightforwardly represented as a PRISM model [32]. The only issue is the presence of patterns allowed in atomic propositions of the PCTL property. Since a pattern basically compactly represents all possible instantiated agents (resp. multisets), it can be expressed as a sum of these agents. To that end, we introduce *formulas* which encode the sum in the PRISM model. Once defined, properties operating with their identifier (in our case the pattern itself) are valid.

Then, we employ Storm, which for a PRISM model, PCTL formula, and given parameter space returns the partitioning of the space to required areas (using $\mathtt{storm} - \mathtt{pars}$). In addition, the tool uses parameter lifting optimisation [40], which improves the state-elimination approach. We apply a simple visualisation to show the result of partitioning graphically.

4.2 Robustness analysis

The *problem of global robustness* [31] of a system s is defined as

$$R^s_{a,P} = \int_P \psi(p) D^s_a(p) dp$$

where a is the property of the system under scrutiny, P is the set of all perturbations, $\psi(p)$ is the probability of the perturbation p, *local robustness* $D^s_a(p)$ is a measure stating how much the property a is preserved in perturbation p. The local robustness returns for each *parameterisation* $p \in P$ the quantitative model checking result for the respective DTMC (built for the parameterisation p) and the given property a.

We solve this problem for given qBCSL model $\mathcal{M}$ and a PCTL property ϕ (with the outermost operator $\Pi_{=?}$). We construct pMC from the model followed by algorithm from [16] to compute the rational function f. Function f can be directly used for evaluation of the local robustness.

We consider the parameter space $\boldsymbol{P}$ as the set of all perturbations. Since each parameterisation $p \in \boldsymbol{P}$ has uniform probability, computing $\psi(p)$ is straightforward – it is inversely proportional to the volume of the entire parameter space. Considering all the assumptions, the robustness for the qBCSL model $\mathcal{M}$ and a property ϕ is computed as $R^{\mathcal{M}}_{\phi,\boldsymbol{P}} = \int_{\boldsymbol{P}} \frac{1}{|\boldsymbol{P}|} f(p)dp$.

We have used Storm to obtain the rational function f and package `scipy` [29] to compute the definite integral of the function in the assumed parameter space. Moreover, since it is possible some discontinuities are present in the function f, we first analyse them using package `sympy` [36] and then integrate without these particular points.

$$
\begin{array}{rcl}
\mathsf{KaiC}^{x}_{\mathsf{cyt}}(\mathsf{S}\{\mathsf{u}\}), \mathsf{KaiC}^{x}_{\mathsf{cyt}}(\mathsf{S}\{\mathsf{u}\}) & \xrightarrow{\frac{\mathsf{kcat}_1 \times [\mathsf{KaiA2}^{y}_{\mathsf{cyt}}(\emptyset)])}{(\mathsf{Km}+[\mathsf{KaiC}^{x}_{\mathsf{cyt}}(\mathsf{S}\{\mathsf{u}\}),\mathsf{KaiC}^{x}_{\mathsf{cyt}}(\mathsf{S}\{\mathsf{u}\})])}} & \mathsf{KaiC}^{x}_{\mathsf{cyt}}(\mathsf{S}\{\mathsf{p}\}), \mathsf{KaiC}^{x}_{\mathsf{cyt}}(\mathsf{S}\{\mathsf{p}\}) \\
\mathsf{KaiC}^{x}_{\mathsf{cyt}}(\mathsf{T}\{\mathsf{u}\}), \mathsf{KaiC}^{x}_{\mathsf{cyt}}(\mathsf{T}\{\mathsf{u}\}) & \xrightarrow{\frac{\mathsf{kcat}_3 \times [\mathsf{KaiB4}^{y}_{\mathsf{cyt}}(\mathsf{act}\{\mathsf{a}\}),\mathsf{KaiA2}^{y}_{\mathsf{cyt}}(\emptyset)])}{(\mathsf{Km}+[\mathsf{KaiC}^{x}_{\mathsf{cyt}}(\mathsf{T}\{\mathsf{u}\}),\mathsf{KaiC}^{x}_{\mathsf{cyt}}(\mathsf{T}\{\mathsf{u}\})])}} & \mathsf{KaiC}^{x}_{\mathsf{cyt}}(\mathsf{T}\{\mathsf{p}\}), \mathsf{KaiC}^{x}_{\mathsf{cyt}}(\mathsf{T}\{\mathsf{p}\}) \\
\mathsf{KaiC}^{x}_{\mathsf{cyt}}(\mathsf{S}\{\mathsf{p}\}), \mathsf{KaiC}^{x}_{\mathsf{cyt}}(\mathsf{S}\{\mathsf{p}\}) & \xrightarrow{\frac{\mathsf{kcat}_2 \times [\mathsf{KaiA2}^{y}_{\mathsf{cyt}}(\emptyset)])}{(\mathsf{Km}+[\mathsf{KaiC}^{x}_{\mathsf{cyt}}(\mathsf{S}\{\mathsf{p}\}),\mathsf{KaiC}^{x}_{\mathsf{cyt}}(\mathsf{S}\{\mathsf{p}\})])}} & \mathsf{KaiC}^{x}_{\mathsf{cyt}}(\mathsf{S}\{\mathsf{u}\}), \mathsf{KaiC}^{x}_{\mathsf{cyt}}(\mathsf{S}\{\mathsf{u}\}) \\
\mathsf{KaiC}^{x}_{\mathsf{cyt}}(\mathsf{T}\{\mathsf{p}\}), \mathsf{KaiC}^{x}_{\mathsf{cyt}}(\mathsf{T}\{\mathsf{p}\}) & \xrightarrow{\frac{\mathsf{kcat}_4 \times [\mathsf{KaiB4}^{y}_{\mathsf{cyt}}(\mathsf{act}\{\mathsf{a}\}),\mathsf{KaiA2}^{y}_{\mathsf{cyt}}(\emptyset)])}{(\mathsf{Km}+[\mathsf{KaiC}^{x}_{\mathsf{cyt}}(\mathsf{T}\{\mathsf{p}\}),\mathsf{KaiC}^{x}_{\mathsf{cyt}}(\mathsf{T}\{\mathsf{p}\})])}} & \mathsf{KaiC}^{x}_{\mathsf{cyt}}(\mathsf{T}\{\mathsf{u}\}), \mathsf{KaiC}^{x}_{\mathsf{cyt}}(\mathsf{T}\{\mathsf{u}\}) \\
\mathsf{KaiB4}^{x}_{\mathsf{cyt}}(\mathsf{act}\{\mathsf{i}\}) & \xrightarrow{\frac{\mathsf{kcat}_{\mathsf{b2}} \times [\mathsf{KaiB4}^{x}_{\mathsf{cyt}}(\mathsf{act}\{\mathsf{i}\})]}{\mathsf{Km}_{\mathsf{b2}}+[\mathsf{KaiB4}^{x}_{\mathsf{cyt}}(\mathsf{act}\{\mathsf{i}\})]}} & \mathsf{KaiB4}^{x}_{\mathsf{cyt}}(\mathsf{act}\{\mathsf{a}\}) \\
\mathsf{KaiB4}^{x}_{\mathsf{cyt}}(\mathsf{act}\{\mathsf{a}\}) & \xrightarrow{\frac{\mathsf{kcat}_{\mathsf{b1}} \times [\mathsf{KaiB4}^{x}_{\mathsf{cyt}}(\mathsf{act}\{\mathsf{a}\})]}{\mathsf{Km}_{\mathsf{b1}}+[\mathsf{KaiB4}^{x}_{\mathsf{cyt}}(\mathsf{act}\{\mathsf{a}\})]}} & \mathsf{KaiB4}^{x}_{\mathsf{cyt}}(\mathsf{act}\{\mathsf{i}\}) \\
\mathsf{KaiC}^{x}_{\mathsf{cyt}}(\emptyset), \mathsf{KaiC}^{y}_{\mathsf{cyt}}(\emptyset) & \xrightarrow{\mathsf{k}_{\mathsf{dimer}} \times [\mathsf{KaiC}^{x}_{\mathsf{cyt}}(\emptyset)] \times ([\mathsf{KaiC}^{y}_{\mathsf{cyt}}(\emptyset)]-1)} & \mathsf{KaiC}^{x}_{\mathsf{cyt}}(\emptyset), \mathsf{KaiC}^{x}_{\mathsf{cyt}}(\emptyset) \\
\mathsf{KaiC}^{x}_{\mathsf{cyt}}(\emptyset), \mathsf{KaiC}^{x}_{\mathsf{cyt}}(\emptyset) & \xrightarrow{\mathsf{k}_{\mathsf{dimer}} \times [\mathsf{KaiC}^{x}_{\mathsf{cyt}}(\emptyset),\mathsf{KaiC}^{x}_{\mathsf{cyt}}(\emptyset)]} & \mathsf{KaiC}^{x}_{\mathsf{cyt}}(\emptyset), \mathsf{KaiC}^{y}_{\mathsf{cyt}}(\emptyset)
\end{array}
$$

Fig. 2. Rules of simplified Miyoshi et al. model. The first four rules are responsible for the change of phosphorylation level of KaiC dimers. The rate functions of these rules represent enzymatic laws and are dependent on current numbers of KaiA dimers and KaiB tetramers. The next two rules change the activity level of KaiB4 complex and the last two rules form and disassembly the KaiC dimer. The particular values of known constants are $\mathsf{Km} = 0.602$, $\mathsf{Km}_{\mathsf{b1}} = 2.423$, $\mathsf{kcat}_{\mathsf{b1}} = 0.602$, $\mathsf{k}_{\mathsf{dimer}} = 1.77$, $\mathsf{Km}_{\mathsf{b2}} = 66.75$, and $\mathsf{kcat}_{\mathsf{b2}} = 0.346$. The exact meaning of individual constants and parameters is described in [37]

5 Case study

In this section, we demonstrate our contribution on a case study[2] from the biological domain. Miyoshi et al. [37] ODE model describes circadian rhythms in cyanobacteria. We have adopted this model to our rule-based formalism with several simplifications in order to avoid combinatorial explosion.

The core of the circadian rhythms model is formed by three main proteins – KaiA, KaiB, and KaiC. The protein KaiC has two phosphorylation sites (S – serine and T – threonine), both of them can be either phosphorylated or unphosphorylated. Two KaiC proteins can form a homo-dimer.

[2] An additional case study targeting a tumour growth is available in Appendix A.

Protein KaiA can also form a homo-dimer and act as a kinase for phosphorylation of KaiC dimers. Since the KaiA dimer cannot undergo any modification, we model it as a single agent. Protein KaiB can form a homo-tetramer, which can be either active or inactive as a whole. For this reason and, again, for the simplicity, we model it as a single agent.

The KaiA dimer has a positive enzymatic effect on the phosphorylation of KaiC dimers. On the other hand, active KaiB tetramer then serves as an inhibitor of KaiC dimer phosphorylation, i.e. it enhances its dephosphorylation. This is done such that it forms a complex with KaiA dimer and inhibits its phosphorylation efforts.

1. **Phosphorylation experiment**
   (a) *initial state*:
   $$\mathsf{KaiC}^1_{\mathsf{cyt}}(\mathsf{S}\{\mathsf{u}\},\mathsf{T}\{\mathsf{u}\}),\mathsf{KaiC}^2_{\mathsf{cyt}}(\mathsf{S}\{\mathsf{u}\},\mathsf{T}\{\mathsf{u}\}),\mathsf{KaiB4}^3_{\mathsf{cyt}}(\mathsf{act}\{\mathsf{a}\}),\mathsf{KaiA2}^4_{\mathsf{cyt}}(\emptyset)$$
   (b) *property of interest*:
   $$\Pi_{\geq 0.99}(\mathsf{True}\ \mathbf{U}\ [\mathsf{KaiC}^x_{\mathsf{cyt}}(\mathsf{S}\{\mathsf{p}\},\mathsf{T}\{\mathsf{p}\}),\mathsf{KaiC}^x_{\mathsf{cyt}}(\mathsf{S}\{\mathsf{p}\},\mathsf{T}\{\mathsf{p}\})] > 0)$$
   (c) *parameters*:
   $$\mathsf{kcat}_1 \in [0,1] \qquad \mathsf{kcat}_2 = 0.539 \qquad \mathsf{k}_{\mathsf{enz}} = 8.756 \times 10^{-4}$$
   $$\mathsf{kcat}_3 \in [0,2] \qquad \mathsf{kcat}_4 = 0.89$$
   (d) *additional rule* for construction of KaiA dimer and KaiB4 tetramer complex:
   $$\mathsf{KaiB4}^x_{\mathsf{cyt}}(\emptyset),\mathsf{KaiA2}^y_{\mathsf{cyt}}(\emptyset) \xrightarrow{\mathsf{k}_{\mathsf{enz}} \times [\mathsf{KaiB4}^x_{\mathsf{cyt}}(\emptyset)] \times [\mathsf{KaiA2}^y_{\mathsf{cyt}}(\emptyset)]} \mathsf{KaiB4}^x_{\mathsf{cyt}}(\emptyset),\mathsf{KaiA2}^x_{\mathsf{cyt}}(\emptyset)$$
2. **Dephosphorylation experiment**
   (a) *initial state*:
   $$\mathsf{KaiC}^1_{\mathsf{cyt}}(\mathsf{S}\{\mathsf{p}\},\mathsf{T}\{\mathsf{p}\}),\mathsf{KaiC}^2_{\mathsf{cyt}}(\mathsf{S}\{\mathsf{p}\},\mathsf{T}\{\mathsf{p}\}),\mathsf{KaiB4}^3_{\mathsf{cyt}}(\mathsf{act}\{\mathsf{a}\}),\mathsf{KaiA2}^3_{\mathsf{cyt}}(\emptyset)$$
   (b) *property of interest*:
   $$\Pi_{\geq 0.99}(\mathsf{True}\ \mathbf{U}\ [\mathsf{KaiC}^x_{\mathsf{cyt}}(\mathsf{S}\{\mathsf{u}\},\mathsf{T}\{\mathsf{u}\}),\mathsf{KaiC}^x_{\mathsf{cyt}}(\mathsf{S}\{\mathsf{u}\},\mathsf{T}\{\mathsf{u}\})] > 0)$$
   (c) *parameters*:
   $$\mathsf{kcat}_1 = 0.539 \qquad \mathsf{kcat}_2 \in [0,1] \qquad \mathsf{k}_{\mathsf{enz}} = 8.756 \times 10^{-4}$$
   $$\mathsf{kcat}_3 = 1.079 \qquad \mathsf{kcat}_4 \in [0,2]$$
   (d) *additional rule* for disassembly of KaiA dimer and KaiB4 tetramer complex:
   $$\mathsf{KaiB4}^x_{\mathsf{cyt}}(\emptyset),\mathsf{KaiA2}^x_{\mathsf{cyt}}(\emptyset) \xrightarrow{\mathsf{k}_{\mathsf{enz}} \times [\mathsf{KaiB4}^x_{\mathsf{cyt}}(\emptyset),\mathsf{KaiA2}^x_{\mathsf{cyt}}(\emptyset)]} \mathsf{KaiB4}^x_{\mathsf{cyt}}(\emptyset),\mathsf{KaiA2}^y_{\mathsf{cyt}}(\emptyset)$$

Fig. 3. Two setups of the Miyoshi model in qBCSL. The goal of the experiment (1) is to find parametrisation such that the model reaches the fully phosphorylated level of KaiC dimer. The model is extended by a rule for the construction of complex possibly disabling the phosphorylation. The experiment (2) is focused on dephosphorylation of KaiC dimer enabled by an additional rule for the enzymatic complex disassembly.

The rules of the model are available in Fig. 2. The mechanism of phosphorylation and activation causes the model to have an oscillatory behaviour. For our simplified case, we investigate whether the probability of reaching the phosphorylated KaiC dimer followed by reaching the unphosphorylated dimer is close to one.

We assume two different experiments both having different initial conditions, one additional rule for manipulation of KaiA and KaiB interaction, different unknown parameters, and finally a different property of interest. Both experiments are specified in Fig. 3. The first experiment expresses conditions with unphosphorylated KaiC dimer and property of reaching the phosphorylated KaiC dimer. For the second experiment, it is the other way around. The probability for both properties should be close to one since the oscillation should always be present.

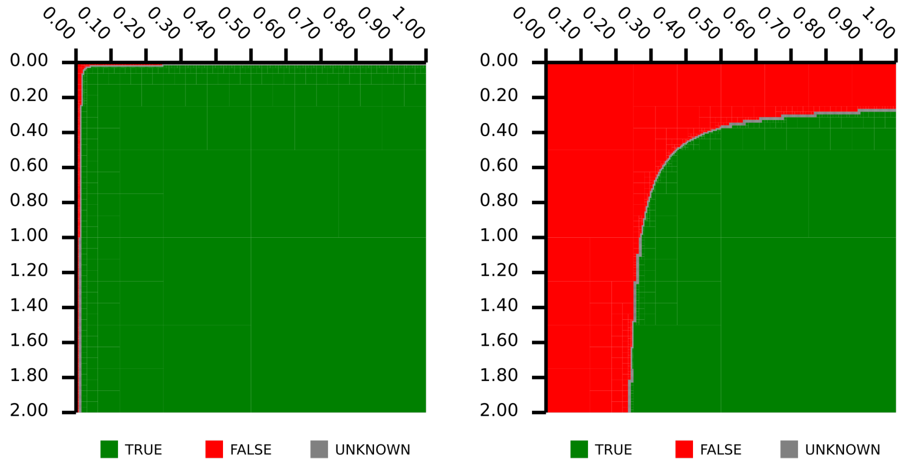

Fig. 4. Visualisation of results of parameter synthesis for the Miyoshi model. The *left* picture depicts the results for the phosphorylation experiment (Fig. 3, 1). The horizontal axis represents values of the parameter $\mathsf{kcat}_1 \in [0, 1]$ and the vertical axis represents values of the parameter $\mathsf{kcat}_3 \in [0, 2]$. The *right* picture depicts the results for the dephosphorylation experiment (Fig. 3, 2). The horizontal axis represents values of the parameter $\mathsf{kcat}_2 \in [0, 1]$ and the vertical axis represents values of the parameter $\mathsf{kcat}_4 \in [0, 2]$.

In Fig. 4, there is a visualisation of parameter synthesis for both cases. The results of the first experiment show that the property is almost always satisfied except for some marginal cases when the parameter values are close to zero. This fact is in agreement with the global robustness degree, which is approximately 0.995. In the second experiment, the property was satisfied in a smaller fraction of parameter space, caused by different initial conditions and the additional rule. However, this difference is very insignificant, which confirms the robustness degree with a value of approximately 0.98. These results confirm that the behaviour of the model is very robust to perturbation of parameters directly responsible for phosphorylation activity, thus showing the oscillatory behaviour is very persistent.

6 Conclusions

First, we have defined a quantitative version of the Biochemical Space Language (qBCSL). The language allows us to specify parametrised quantitative aspects (rates) of the dynamics of individual rules, resulting in probabilistic behaviour of models considered in discrete time. Second, we have encoded the semantics of qBCSL models by means of parametric Markov Chains. That enables applications of existing symbolic parameter synthesis methods. Finally, we have shown how to (exactly) compute robustness of a given property with respect to a given parameter perturbation. Bridging the efficient parameter synthesis methods with rule-based modelling is an important step towards application of formal methos in biological domain [7,10]. To that end, we have demonstrated our approach on a case study from the biological domain.

The main challenge to be faced in future is the scalability. Rule-based models can expand in large state spaces making thus the construction of the pQLTS (and pMC) infeasible. In particular, we want to find ways allowing to avoid enumeration of the pMC, e.g., by employing on-the-fly and static analysis approaches.

References

1. Backenköhler, M., Bortolussi, L., Wolf, V.: Moment-based parameter estimation for stochastic reaction networks in equilibrium. TCBB **15**(4), 1180–1192 (2018)
2. Baier, C., Haverkort, B., Hermanns, H., Katoen, J.P.: Model-checking algorithms for continuous-time markov chains. IEEE Transactions on software engineering **29**(6), 524–541 (2003)
3. Barbuti, R., Caravagna, G., Maggiolo-Schettini, A., Milazzo, P.: An intermediate language for the stochastic simulation of biological systems. TCS **410**(33-34), 3085–3109 (2009)
4. Barbuti, R., Levi, F., Milazzo, P., Scatena, G.: Probabilistic model checking of biological systems with uncertain kinetic rates. TCS **419**, 2–16 (2012)
5. Bartocci, E., Bortolussi, L., Nenzi, L., Sanguinetti, G.: On the robustness of temporal properties for stochastic models. arXiv preprint arXiv:1309.0866 (2013)
6. Bock, C., Bortolussi, L., Krüger, T., Mikeev, L., Wolf, V.: Model-based whole-genome analysis of DNA methylation fidelity. In: HSB. pp. 141–155. Springer (2015)
7. Bonzanni, N., Feenstra, K.A., Fokkink, W., Krepska, E.: What can formal methods bring to systems biology? In: International Symposium on Formal Methods. pp. 16–22. Springer (2009)
8. Bortolussi, L., Milios, D., Sanguinetti, G.: Smoothed model checking for uncertain continuous-time markov chains. Inform. and Comput. **247**, 235 – 253 (2016)
9. Bortolussi, L., Silvetti, S.: Bayesian statistical parameter synthesis for linear temporal properties of stochastic models. In: TACAS. pp. 396–413. Springer (2018)
10. Brim, L., Češka, M., Šafránek, D.: Model checking of biological systems. In: International School on Formal Methods for the Design of Computer, Communication and Software Systems. pp. 63–112. Springer (2013)
11. Calzone, L., Fages, F., Soliman, S.: BIOCHAM: an environment for modeling biological systems and formalizing experimental knowledge. Bioinformatics **22**(14), 1805–1807 (2006)
12. Česka, M., Šafránek, D., Dražan, S., Brim, L.: Robustness analysis of stochastic biochemical systems. PloS One **9**(4), e94553 (2014)
13. Ciocchetta, F., Hillston, J.: Bio-PEPA: an extension of the process algebra PEPA for biochemical networks. ENTCS **194**(3), 103–117 (2008)
14. Clarke, E.M., Emerson, E.A., Sistla, A.P.: Automatic verification of finite-state concurrent systems using temporal logic specifications. ACM Transactions on Programming Languages and Systems (TOPLAS) **8**(2), 244–263 (1986)

15. Danos, V., Laneve, C.: Formal molecular biology. Theor. Comp. Sci. **325**(1), 69 – 110 (2004)
16. Daws, C.: Symbolic and parametric model checking of discrete-time markov chains. In: ICTAC. pp. 280–294. Springer (2004)
17. Dehnert, C., Junges, S., Jansen, N., Corzilius, F., Volk, M., Bruintjes, H., Katoen, J.P., Abraham, E.: Prophesy: A probabilistic parameter synthesis tool. In: CAV. pp. 214–231. Springer (2015)
18. Dehnert, C., Junges, S., Katoen, J.P., Volk, M.: A storm is coming: A modern probabilistic model checker. In: CAV. pp. 592–600. Springer (2017)
19. Faeder, J.R., Blinov, M.L., Hlavacek, W.S., et al.: Rule-based modeling of biochemical systems with BioNetGen. Methods Mol. Biol. **500**, 113–167 (2009)
20. Fisher, J., Henzinger, T.A.: Executable cell biology. Nature biotechnology **25**(11), 1239 (2007)
21. Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. The journal of physical chemistry **81**(25), 2340–2361 (1977)
22. Hahn, E.M., Hermanns, H., Zhang, L.: Probabilistic reachability for parametric markov models. STTT **13**(1), 3–19 (2011)
23. Hasson, H., Jonsson, B.: A logic for reasoning about time and probability. FAOC **6**, 512–535 (1994)
24. Honorato-Zimmer, R., Millar, A.J., Plotkin, G.D., Zardilis, A.: Chromar, a rule-based language of parameterised objects. TCS (2017)
25. Hopcroft, J.E.: Introduction to automata theory, languages, and computation. Pearson Education India (2008)
26. Hutschenreiter, L., Baier, C., Klein, J.: Parametric markov chains: PCTL complexity and fraction-free gaussian elimination. arXiv preprint arXiv:1709.02093 (2017)
27. Jansen, N., Corzilius, F., Volk, M., Wimmer, R., Ábrahám, E., Katoen, J.P., Becker, B.: Accelerating parametric probabilistic verification. In: QEST. pp. 404–420. Springer (2014)
28. Jha, S.K., Clarke, E.M., Langmead, C.J., Legay, A., Platzer, A., Zuliani, P.: A bayesian approach to model checking biological systems. In: International conference on computational methods in systems biology. pp. 218–234. Springer (2009)
29. Jones, E., Oliphant, T., Peterson, P., et al.: SciPy: Open source scientific tools for Python (2001–), `http://www.scipy.org/`
30. Khalid, A., Jha, S.K.: Calibration of rule-based stochastic biochemical models using statistical model checking. In: 2018 IEEE BIBM. pp. 179–184 (2018)
31. Kitano, H.: Towards a theory of biological robustness. Molecular systems biology **3**(1), 137 (2007)
32. Kwiatkowska, M., Norman, G., Parker, D.: Quantitative analysis with the probabilistic model checker PRISM. ENTCS **153**(2), 5–31 (2006)
33. Lanotte, R., Maggiolo-Schettini, A., Troina, A.: Parametric probabilistic transition systems for system design and analysis. FAOC **19**(1), 93–109 (Mar 2007)
34. Liu, B., Faeder, J.R.: Parameter estimation of rule-based models using statistical model checking. In: 2016 IEEE BIBM. pp. 1453–1459. IEEE (2016)
35. Lück, A., Wolf, V.: Generalized method of moments for estimating parameters of stochastic reaction networks. BMC Systems Biology **10**(1), 98 (Oct 2016)
36. Meurer, A., Smith, C.P., Paprocki, M., Čertík, O., Kirpichev, S.B., Rocklin, M., et al.: SymPy: symbolic computing in Python. PeerJ Comp. Sci. **3**, e103 (2017)
37. Miyoshi, F., Nakayama, Y., Kaizu, K., Iwasaki, H., Tomita, M.: A mathematical model for the Kai-protein-based chemical oscillator and clock gene expression rhythms in cyanobacteria. Journal of Biological Rhythms **22**(1), 69–80 (2007)

38. Nedbal, L., Červený, J., Schmidt, H.: Scaling and integration of kinetic models of photosynthesis: Towards comprehensive e-photosynthesis. In: Photosynthesis in silico, pp. 17–29. Springer (2009)
39. Pedersen, M., Phillips, A., Plotkin, G.D.: A high-level language for rule-based modelling. PloS One **10**, 1–26 (2015)
40. Quatmann, T., Dehnert, C., Jansen, N., Junges, S., Katoen, J.P.: Parameter synthesis for markov models: Faster than ever. In: ICTAC. pp. 50–67. Springer (2016)
41. Romers, J.C., Krantz, M.: rxncon 2.0: A language for executable molecular systems biology. bioRxiv (2017)
42. Troják, M., Šafránek, D., Brim, L.: Executable biochemical space for specification and analysis of biochemical systems. In: SASB (2018), to appear.
43. Troják, M. et al.: E-cyanobacterium.org: A web-based platform for systems biology of cyanobacteria. In: CMSB (2016)
44. Villasana, M., Radunskaya, A.: A delay differential equation model for tumor growth. Journal of Mathematical Biology **47**(3), 270–294 (2003)

A Tumour growth

Tumour growth is based on *mitosis* (i.e. cell division). The cell cycle is the process between two mitoses and it consists of four phases: the resting phase G_1, the DNA replication phase S, the resting phase G_2, and the mitosis phase M in which the cells segregate the duplicated sets of chromosomes between daughter cells. The three phases G_1, S, and G_2 constitute the pre-mitotic phase, also called *interphase*.

We have adopted the model of tumour growth [44] to our language. It considers two populations of tumour cells: those in interphase and those in mitosis. We represent the tumour cell as an agent T. The current phase is expressed with an atom phase in its composition, which can have two different states – i for interphase and m for mitosis. For simplicity, we omit the compartment from the rules since it does not change and plays no important role in this model.

$$
\begin{aligned}
&\mathsf{T}^x(\mathsf{phase}\{\mathsf{i}\}) \xrightarrow{\mathsf{a}_1\times[\mathsf{T}^x(\mathsf{phase}\{\mathsf{i}\})]} \mathsf{T}^x(\mathsf{phase}\{\mathsf{m}\})\\
&\mathsf{T}^x(\mathsf{phase}\{\mathsf{m}\}) \xrightarrow{\mathsf{a}_2\times[\mathsf{T}^x(\mathsf{phase}\{\mathsf{m}\})]} \mathsf{T}^y(\mathsf{phase}\{\mathsf{i}\}), \mathsf{T}^z(\mathsf{phase}\{\mathsf{i}\})\\
&\mathsf{T}^x(\mathsf{phase}\{\mathsf{i}\}) \xrightarrow{\mathsf{d}_1\times[\mathsf{T}^x(\mathsf{phase}\{\mathsf{i}\})]} \emptyset\\
&\mathsf{T}^x(\mathsf{phase}\{\mathsf{m}\}) \xrightarrow{\mathsf{d}_2\times[\mathsf{T}^x(\mathsf{phase}\{\mathsf{m}\})]} \emptyset
\end{aligned}
$$

Fig. 5. Rules of the tumour growth model. The first rule describes the change of the phase of a cell from interphase to mitosis. The second rule describes the duplication of the cell to two daughter cells. Note that both start in interphase. The last two rules describe the death of cells in both possible states.

The rules of the model are available in Fig. 5. Note that this model is a demonstration where all rules are *reaction-based*, i.e. they do not describe an abstract rule, only modification of concrete agents.

Given rate functions of rules are parametrised. Parameters a_1 and a_2 are present in rules responsible for change of phase and cell division, while parameters d_1 and d_2 are in the rules where the cell *disappears* or *dies*. The values $\mathsf{a}_2 = 0.5$ and $\mathsf{d}_1 = 0.3$ are constant the other two parameters are given by admissible ranges: $\mathsf{a}_1 \in [0; 3]$ and for $\mathsf{d}_2 \in [0.001; 0.5]$.

For the *initial state*, we assume a single agent $\mathsf{T}^1(\mathsf{phase}\{\mathsf{i}\})$. Please note that the model gives rise to infinite pMC since the second rule can *generate* additional agents. To obtain a finite abstract probabilistic model, we have heuristically limited the number of states of the model. Particularly, we generate all the states having the number of individuals of both species less or equal to 5 and we introduce a special abstract state which represents all the other states, which limits the size of possible state space to 6^2. This approximation is incorrect only in cases when one wants to reach a state which is represented by the special state.

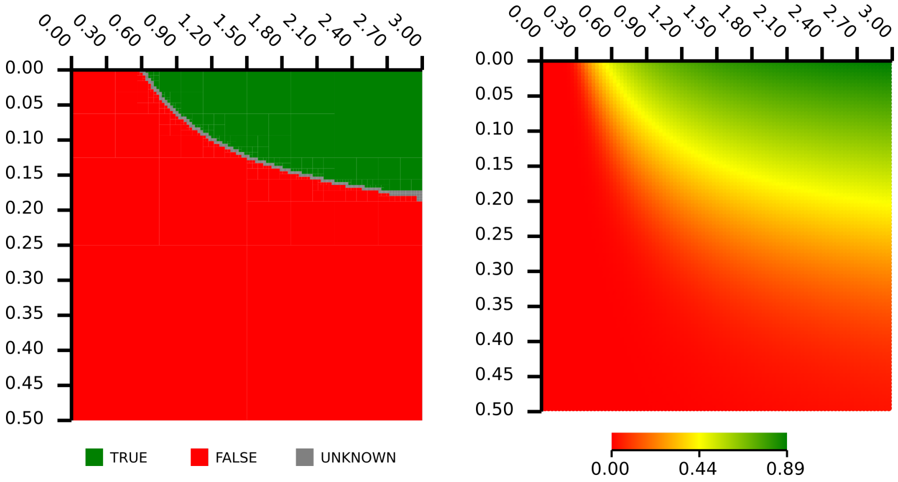

Fig. 6. Visualisation of results of parameter synthesis (*left*) and quantitative model checking using sampling (*right*) for property ϕ for tumour growth model. The horizontal axis represents values of the parameter $a_1 \in [0, 3]$ and the vertical axis represents values of the parameter $d_2 \in [0.001, 0.5]$. The probability threshold 0.5 from the property ϕ is visible in both sampling (approximately the yellow line) and parameter synthesis (the grey line). It shows that the parameter synthesis method gives us a very precise result and is in agreement with quantitative model checking.

We are interested in property whether the population of tumour cells will reach almost its maximum with the probability higher than 0.5, meaning that the growth is not random but has rather tendency to grow without limitations. This property can be expressed as $\phi = \Pi_{\geq 0.5}(\mathsf{True}\ \mathbf{U}\ \mathsf{T}^j(\emptyset) > 8)$. In Fig. 6, there is a visualisation of parameter synthesis. The results show that the higher values of the parameter a_1 (cell division) and the lower values of the parameter d_2 increase the probability of property satisfaction. This result is quite expected, because both parameters directly influence cell division (a_1) and degradation (d_2) of cells. We have also computed the global robustness degree of the property, which is approximately 0.24. It can be interpreted as 24% of parameter space satisfies the property $\mathsf{True}\ \mathbf{U}\ \mathsf{T}^j(\emptyset) > 8$.