I Continuous Stochastic Logic

Let $\mathcal{C} = (S, s_0, R, L)$ be a labelled CTMC such that $L$ is a labelling function which assigns to each state $s \in S$ the set $L(s)$ of atomic propositions that are valid in state $s$. We consider the specification of the inspected property using the bounded time fragment of CSL with rewards and post-processing functions. The syntax of this logic is defined in the following way. A state formula $\Phi$ is given as

$$\Phi ::= \text{true} \mid a \mid \neg \Phi \mid \Phi \land \Phi \mid P_\sim p[\phi] \mid R_\sim r[\mathcal{C} \leq t] \mid R_\sim r[\mathcal{C} = t] \mid E_\sim r[\mathcal{C} = t]$$

where $\phi$ is a path formula given as $\phi ::= X \Phi \mid \Phi U^I \Phi$, $a$ is an atomic proposition, $\sim \in \{<, \leq, \geq, >\}$, $p \in [0, 1]$ is a probability, $r \in \mathbb{R}_{\geq 0}$ is an expected reward and $I = [a, b]$ is a bounded time interval such that $a, b \in \mathbb{R}_{\geq 0}$ and $a \leq b$. Path operators $\mathcal{G}$ (always) and $\mathcal{F}$ (eventually) are derived in the standard way using the operator $U$. In order to specify properties containing rewards ($R_\sim r[\mathcal{C} \leq t]$ is the cumulative reward acquired up to time $t$, $R_\sim r[\mathcal{C} = t]$ is the instantaneous reward in time $t$) the CTMC $\mathcal{C}$ is enhanced with reward (cost) structures. Two types of reward structures are used. A state reward $\rho(s)$ defines the rate with which a reward is acquired in state $s \in S$. A reward of $t \cdot \rho(s)$ is acquired if a CTMC remains in state $s$ for $t$ time units. A transition reward $\iota(s_i, s_j)$ defines the reward acquired each time the transition $(s_i, s_j)$ occurs.

Since the reward function has to be defined before the actual analysis of the CTMC, the rewards for particular states have to be known prior to the specification of the property. This fact limits the class of properties that can be expressed using such structures. For example, noise expressed by a mean quadratic deviation ($mqd$) of the population probability distribution of a species at a given time cannot be specified using CSL with rewards. To compute the $mqd$ we need to know the mean of the distribution to be able to obtain the corresponding coefficients and encode them into state rewards.

To overcome this problem we introduce the abstract state operator $E_{\sim r}[\mathcal{C} = t]$ which evaluates the state distribution $\pi^{\mathcal{C}, s_0, t}$ at the given time instant $t$ by a user provided real-valued post-processing function $Post(\pi^{\mathcal{C}, s_0, t})$ and compares it to $\sim r$.

The formal semantics of the bounded fragment of CSL with rewards and post-processing functions is defined similarly as the semantics of full CSL and thus we refer the readers to original papers. The key part of the semantics is given by the definition of the satisfaction relation $\models$. It specifies when a state $s$ satisfies the state formula $\Phi$ (denoted as $s \models \Phi$) and when a path $\omega$ satisfies the path formula $\phi$ (denoted as $\omega \models \phi$). The informal definition of $\models$ is as follows:

- $s \models E_{\sim r}[\mathcal{C} = t]$ iff $Post(\pi^{\mathcal{C}, s, t})$ satisfies $\sim r$.
- $s \models P_\sim p[\phi]$ iff the probability of all paths $\omega \in Path^\mathcal{C}(s)$ that satisfy the path formula $\phi$ (denoted as $\text{Prob}^\mathcal{C}(s, \phi)$) satisfies $\sim p$, where
  - $\omega$ satisfies $X \Phi$ iff the second state on $\omega$ satisfies $\Phi$
  - $\omega$ satisfies $\Phi U^I \Psi$ iff there exists a time instant $t \in I$ such that the state on $\omega$ occupied at $t$ satisfies $\Psi$ and all states on $\omega$ occupied before $t' \in [0, t]$ satisfy $\Phi$
• $s \vdash R_{r,s}[C \leq t]$ if the sum of expected rewards over $Path^C(s)$ cumulated until $t$ time units (denoted as $Exp^C(s, X_{C \leq t})$) satisfies $\sim r$

• $s \vdash R_{r,s}[I = t]$ if the sum of expected rewards over all paths $\omega \in Path^C(s)$ at time $t$ (denoted as $Exp^C(s, X_{I = t})$) satisfies $\sim r$.

A set $Sat_C(\Phi) = \{s \in S \mid s \vdash \Phi\}$ denotes the set of states that satisfy $\Phi$.

Note that the syntax and semantics can be easily extended with “quantitative” formulae in the form $\Phi := P_\leq \gamma[\phi] \mid R_{=\leq}[C \leq t] \mid R_{=\leq}[I = t] \mid E_{\leq\geq}[I = t]$, i.e., the topmost operator of the formula $\Phi$ returns a quantitative result, as used, e.g., in PRISM [1]. In this case the result of a decision procedure is not in the form of a boolean yes/no answer but is the actual numerical value of the probability $Prob^C(s, \phi)$, the expected reward $Exp^C(s, \gamma)$, or the value of $Post^C(s, t)$. The computation of a numerical value is of the same complexity class as the computation of a result to be compared leading to a boolean answer, although in some cases the comparison may be carried out on less precise or preliminary results.

II Transient Analysis

The aim of transient analysis is to compute a transient probability distribution. Given an initial distribution $\pi^{C, s_0, 0}$ (i.e., $\pi^{C, s_0, 0}(s) = 1$ if $s_0 = s$, and 0, otherwise) at time 0 of a CTMC $C = (S, s_0, R)$ what will the transient state distribution $\pi^{C, s_0, t}$ look like in some future yet finite time $t \in \mathbb{R}_{\geq 0}$?

Transient analysis of a CTMC may be efficiently carried out by a standard technique called uniformisation [2]. The transient probability in time $t$ is obtained as a sum of expressions giving the state distributions after $i$ discrete reaction steps of the respective uniformised discrete time Markov chain (DTMC) weighted by the $i$th probability of the Poisson process. It is the probability of $i$ such steps occurring in time $t$, assuming the delays between steps of the CTMC $C$ are exponentially distributed with rate $q$. Formally, for the rate $q$ satisfying $q \geq \max\{E^C(s) \mid s \in S\}$ ($E$ is the exit rate of state $s$) the uniformised DTMC $unif(C)$ is defined as $unif(C) = (\mathbb{S}, s_0, Q^{unif(C)})$ where

$$Q^{unif(C)}(s, s') = \left\{ \begin{array}{ll} \frac{R(s, s')}{q} & \text{if } s \neq s' \\ 1 - \sum_{s'' \neq s} \frac{R(s, s'')}{q} & \text{otherwise,} \end{array} \right.$$  

and the $i$th Poisson probability in time $t$ is given as $\gamma_{i, q, t} = e^{-q.t} \cdot (q.t)^i \cdot i!$. The transient probability can be computed as follows:

$$\pi^{C, s_0, t} = \sum_{i=0}^{\infty} \gamma_{i, q, t} \cdot \pi^{C, s_0, 0} \cdot (Q^{unif(C)})^i \approx \sum_{i=L_c}^{R_c} \gamma_{i, q, t} \cdot \pi^{C, s_0, 0} \cdot (Q^{unif(C)})^i.$$  

Although the sum is in general infinite, for a given precision $\epsilon$ the lower and upper bounds $L_c, R_c$ can be estimated by using techniques such as that of Fox and Glynn [3] which also allow for efficient solutions of the Poisson process. In order to make the computation of uniformisation feasible the matrix-matrix multiplication is reduced to a vector-matrix multiplication, i.e.,

$$\pi^{C, s_0, 0} \cdot (Q^{unif(C)})^i = (\pi^{C, s_0, 0} \cdot (Q^{unif(C)})^{i-1}) \cdot (Q^{unif(C)}).$$

Standard uniformisation can be intractable when the system under study is too complex, i.e., contains more than an order of $10^7$ states and the upper estimate $R_c$, denoting the number of vector-matrix multiplications as iterations, is high (more than an order of $10^9$). Therefore, many approximation techniques have been studied in order to reduce the state space and to lower the number of iterations $R_c$. State space reductions are based on the observation that in many cases (especially in biochemical systems) a
significant amount of the probability mass in a given time is localised in a manageable set of states. Thus
neglecting states with insignificant probability can dramatically reduce the state space while the resulting
approximation of the transient probability is still sufficient. Methods allowing efficient state-space reduc-
tion are based on finite projection techniques [4, 5], dynamic state space truncation [6] and aggregation
methods [7].

Since the number of iterations \(R_e\) inherently depends on the uniformisation rate \(q\) that has to be
greater than the maximal exit rate of all the states of the system, a variant of standard uniformisation,
the so-called adaptive uniformisation [8], has been proposed. It uses a uniformisation rate that adapts
depending on the set of states the system can occupy at a given time, i.e., after a particular number of
reactions. In many cases, a significantly smaller rate \(q\) can be used and thus the number of iterations \(R_e\)
can be significantly reduced during some parts of the computation. Moreover, adaptive uniformisation
can be successfully combined with reduction techniques mentioned above [6]. The downside of adaptive
uniformisation is that the Poisson process has to be replaced with a general birth process which is more
expensive to solve. More details can be found, e.g., in [8].

III Global CSL Model-Checking

Let us first define an auxiliary function \(Eval(C, \Phi)\), that returns the numerical value representing the
quantitative model-checking result for a given CTMC \(C = \{S, s_0, R\}\) and an inspected formula \(\Phi\), in the
following way:

\[
Eval(C, \Phi) = \begin{cases} 
Prob^C(s_0, \phi) & \text{if } \Phi \equiv P_*[\phi] \\
Exp^C(s_0, X_{C \leq t}) & \text{if } \Phi \equiv R_*[C \leq t] \\
Exp^C(s_0, X_{t = t}) & \text{if } \Phi \equiv R_*[t = t] \\
Post(\pi^C,s_0,t) & \text{if } \Phi \equiv E_*[t = t]
\end{cases}
\]

(1)

where \(* \in \{=?, \sim r\}\).

The goal of the local model checking technique is to compute the value \(Eval(C, \Phi)\). On the other hand,
for the global model checking technique it is to compute the values of \(Eval(C, \Phi)\) for all \(C = \{S, s, R\}\)
where \(s \in S\). The crucial advantage of the global approach is the fact that it has the same computational
complexity as the local approach. Therefore, the global model checking technique is much more suitable
for robustness analysis over perturbations of initial conditions that are encoded as the initial state of the
corresponding CTMC.

Global model checking returns the vector of size \(|S|\) such that the \(i\)th position contains the model
checking result provided that \(s_i\) is the initial state. Let \(C = (S, R, L)\) be a labelled CTMC where the
initial state is not specified. The crucial part of this method is to compute the vector of probabilities
\(Prob^C,\phi\) for any path formula \(\phi\) and the vector of expected rewards \(Exp^C_X\) for \(X \in \{X_{t = t}, X_C \leq t\}\) such
that for all \(s \in S\) the following holds:

\[
Prob^C,\phi(s) = Prob^C(s, \phi) \land Exp^C_X(s) = Exp^C(s, X)
\]

In local model checking the computation of \(Prob^C(s, \phi)\) and \(Exp^C(s, X)\) is reduced to the computation
of the transient probability distribution \(\pi^C,s,t\), as described, e.g., in [2, 9]. Thus, for different initial states
\(s\) we have to compute the corresponding transient probability distributions separately. The key idea of
the global model checking method is to use backward transient analysis. The result of backward transient
analysis is the vector \(\tau^{C,A,t}\) such that for an arbitrary set of states \(A\), the value \(\tau^{C,A,t}(s)\) is the probability
that \(A\) is reached from \(s\) at the time \(t\). Without going into details the vector \(\tau^{C,A,t}\) can be computed in a
very similar way using the uniformised DTMC \(unif(C)\) as in the case of vector \(\pi^{C,s,t}\). Only vector-matrix
multiplications are replaced by matrix-transposed-vector multiplication and \(\tau^{C,A,0}(s) = 1\) if \(s \in A\), and
0, otherwise.
The global model checking technique cannot be used if $\Phi$ includes the operator $E_{I}[t^t]$. In such a case we have to compute the value $Post(\pi^{C,s,t})$. Hence the local model checking technique has to be employed, i.e., we first compute the vector $\pi^{C,s,t}$ and then apply the user specified function $Post$.

Now we briefly show how the vector $Prob^{C,\Phi}$ is computed using backward transient analysis. Since the definition of next operator $X \Phi$ does not rely on any real time aspects of CTMCs, its evaluation stems from the probability of the next reaction that can be easily obtained from the transition matrix $R$. The evaluation of the until operator $\Phi_1 U^{t} \Phi_2$ depends on the form of the interval $I$ and is separately solved for the cases of $I = [0, t_1]$ and $I = [t_1, t_2]$ where $t_1, t_2 \in \mathbb{R}_{\geq 0}$. It is based on a modification of the uniformised infinitesimal generator matrix $Q_{uni}$ where certain states are made absorbing. This means that all outgoing transitions are ignored reflecting the validity of $\Phi_1$ and $\Phi_2$ in these states.

For any CSL formula $\Phi$, let $C[\Phi] = (S, s_0, R[\Phi], L)$, where $R[\Phi](s, s') = R(s, s')$, if $s \models \Phi$, and 0, otherwise. The formula $\phi = \Phi_1 U^{[0, t]} \Phi_2$ can be evaluated using the vector $\tau^{C,\Phi_1,\Phi_2}$ in the following way:

$$
Prob^{C,\phi} = \tau^{C[\neg \Phi_1 \land \Phi_2],A,t} \quad \text{where } s \in A \text{ iff } s \models \Phi_2.
$$

For the formula $\phi = \Phi_1 U^{[t_1, t_2]} \Phi_2$ the evaluation is split into two parts: staying in states satisfying $\Phi_1$ until time $t_1$ and reaching a state satisfying $\Phi_2$, while remaining in states satisfying $\Phi_1$, within time $t_2 - t_1$. The formula $\phi$ can be evaluated using the vector $\tau^{C,\pi,t}$ that takes a vector $\pi$ instead of a set $A$ (i.e., $\tau^{C,\pi,0} = \pi$) in the following way:

$$
Prob^{C,\phi} = \tau^{C[\neg \Phi_1 \land \Phi_2],A,t_1} \quad \text{where } \pi = \tau^{C[\neg \Phi_1 \land \Phi_2],A,t_2-t_1} \quad \text{and } s \in A \text{ iff } s \models \Phi_2.
$$

The backward transient analysis can be also used in the case of reward computation. Since operator $R_{\sim_p}[I^{[t]}]$ expresses the expected reward at time $t$, the vector $Exp^{C,X_{\leq t}}$ can be computed as follows:

$$
Exp^{C,X_{\leq t}} = \tau^{C,\pi,t} \quad \text{where } \pi = \rho \text{ such that } \rho \text{ is the given state reward structure.}
$$

For evaluation of the operator $R_{\sim_p}[C^{\leq t}]$ we have to use mixed Poisson probabilities (see, e.g., [2, 10]) in the backward transient analysis. It means that during the uniformisation the Poisson probabilities $\gamma_{i,q,t}$ are replaced by the mixed Poisson probabilities $\tilde{\gamma}_{i,q,t}$ that can be computed as:

$$
\tilde{\gamma}_{i,q,t} = \frac{1}{q} \cdot \left(1 - \sum_{j=1}^{i} \gamma_{j,q,t}\right).
$$

Using the given state reward structure $\rho$ we can compute the vector $Exp^{C,X_{\leq t}}$ in the following way:

$$
Exp^{C,X_{\leq t}} = \tau^{C,\pi,t} \quad \text{where } \pi = \rho \text{ and the mixed Poisson probabilities } \tilde{\gamma}_{i,q,t} \text{ are used.}
$$

To recap the overall method of stochastic model checking of CTMCs over CSL formulae we present the methods from an abstract perspective. The evaluation of a structured formula $\Phi$ proceeds by bottom-up evaluation of a set of atomic propositions, probabilistic or expected reward inequalities and their boolean combinations. This evaluation gives us a discrete set of states that are further used in the following computation. The process continues up the formula until the root is reached. The final verdict is reported either in the form of a boolean yes/no answer or as the actual numerical value of the probability or the expected reward. This process can be easily extended for the operator $E_{I}[t^t]$, though the local model checking method has to be used.

IV Min-Max Approximation

The key idea of min-max approximation proposed in [11] is to approximate the largest set of states satisfying $\Phi$, and the smallest set of states satisfying $\Phi$ with respect to the space of perturbations $P$.
Let \( C \) be a set of parameterised CTMCs induced by the space of perturbations \( P \). We compute the approximation \( \text{Sat}_C^\top(\Phi) \) and \( \text{Sat}_C^\bot(\Phi) \) such that

\[
\text{Sat}_C^\top(\Phi) \supseteq \bigcup_{c_p \in C} \text{Sat}_{c_p}(\Phi) \land \text{Sat}_C^\bot(\Phi) \subseteq \bigcap_{c_p \in C} \text{Sat}_{c_p}(\Phi),
\]

where \( s \in \text{Sat}_{c_p}(\Phi) \) iff \( s \models \Phi \) in CTMC \( C_p \). To obtain such approximations we extended the satisfaction relation \( \models \) and showed that it is sufficient for an arbitrary path formula \( \phi \), and \( X \in \{X_{C\leq 1}, X_{1\leq t}\} \) to compute the vectors \( \text{Prob}_{\Phi}^{C,\phi}, \text{Prob}_{\Phi}^{C,\phi}, \text{Exp}_{\Phi}^{C,X}, \text{Exp}_{\Phi}^{C,X} \) and \( \text{Exp}_{\Phi}^{C,X} \) such that for each \( s \in S \) the following holds:

\[
\begin{align*}
\text{Prob}_{\Phi}^{C,\phi}(s) & \geq \max\{\text{Prob}_{\Phi}^{C_p,\phi}(s) \mid c_p \in C\} \\
\text{Prob}_{\Phi}^{C,\phi}(s) & \leq \min\{\text{Prob}_{\Phi}^{C_p,\phi}(s) \mid c_p \in C\} \\
\text{Exp}_{\Phi}^{C,X}(s) & \geq \max\{\text{Exp}_{\Phi}^{C_p,X}(s) \mid c_p \in C\} \text{ for } X \in \{X_{1\leq t}, X_{C\leq 1}\} \\
\text{Exp}_{\Phi}^{C,X}(s) & \leq \min\{\text{Exp}_{\Phi}^{C_p,X}(s) \mid c_p \in C\} \text{ for } X \in \{X_{1\leq t}, X_{C\leq 1}\}.
\end{align*}
\] (2)

The min-max approximation can be easily extended to the operator \( E_{\sigma} \). For the given state \( s \in S \) and the time \( t \) it is sufficient to compute the values \( \text{Post}_C^\top(s, t) \) and \( \text{Post}_C^\bot(s, t) \) such that the following holds:

\[
\begin{align*}
\text{Post}_C^\top(s, t) & \geq \max\{\text{Post}(\pi_{C_p,s,t}) \mid c_p \in C\} \\
\text{Post}_C^\bot(s, t) & \leq \min\{\text{Post}(\pi_{C_p,s,t}) \mid c_p \in C\}.
\end{align*}
\] (3)

The approximated sets \( \text{Sat}_C^\top(\Phi) \) and \( \text{Sat}_C^\bot(\Phi) \) are further used in the computation of \( D_{\Phi,P}^\top \) and \( D_{\Phi,P}^\bot \). If the topmost operator of the formula \( \Phi \) is \( P=\top[\phi] \) then

\[
D_{\Phi,P}^\bot = \text{Prob}^{C,\phi}(s) \land \text{Prob}^{C,\phi}(s).
\]

If the topmost operator of the formula \( \Phi \) is \( R=\top[C\leq t] \) and \( R=\top[l=t] \) then

\[
D_{\Phi,P}^\bot = \text{Exp}^{C,X}(s) \land \text{Exp}^{C,X}(s) \text{ for } X = X_{C\leq 1} \text{ and } X = X_{1\leq t}, \text{ respectively}.
\]

Similarly, if the topmost operator of the formula \( \Phi \) is \( E=\top[l=t] \) then

\[
D_{\Phi,P}^\bot = \text{Post}_C^\bot(s, t) \land D_{\Phi,P}^\bot = \text{Post}_C^\bot(s, t).
\]

\section{Parameterised Uniformisation}

The parameterised uniformisation introduced in [11] is a modification of the standard uniformisation technique that allows us to compute strict approximations of the minimal and maximal transient probability with respect to the set \( C \), moreover, the modification preserves the asymptotic time complexity of standard uniformisation. For the given state \( s \in S \) and time \( t \in \mathbb{R}_{\geq 0} \) the parameterised uniformisation returns vectors \( \pi_{\Phi,s,t}^C \) and \( \pi_{\Phi,s,t}^C \) such that for each state \( s' \in S \) the following holds:

\[
\pi_{\Phi,s,t}^C(s') \geq \max\{\pi_{C_p,s,t}(s') \mid c_p \in C\} \land \pi_{\Phi,s,t}^C(s') \leq \min\{\pi_{C_p,s,t}(s') \mid c_p \in C\}
\]

The modification is based on the computation of the local maximum (minimum) of \( \pi_{C_p,s,t}(s') \) over all \( C_p \in C \) for each state \( s' \) and in each iteration \( i \) of the standard uniformisation. It means that in the \( i \)th iteration of the computation for a state \( s' \) we consider only the maximal (minimal) values in the relevant states in the iteration \( i-I \), i.e., the states that affect \( \pi_{C_p,s,t}(s') \).

In [11] we have defined the function \( \sigma(s) \) (formally \( \sigma(p,s,\pi) \)) which for each state \( s \in S \), perturbation point \( p \in P \), and probability distribution \( \pi \) (or pseudo-distribution with the sum smaller or larger than
1) returns the difference of probability mass inflow and outflow to/from state \( s \). If all reactions are described by mass action kinetics the resulting \( \sigma \) functions are monotonic with respect to any single perturbed stochastic rate constant \( k_r \). This allows us to efficiently compute for each state \( s' \) the local maximum (minimum) of \( \pi_{\alpha_i^{s,s'}}(s') \) over all \( C_p \in C \) corresponding to \( P \).

However, in the case of more complex rate functions than those resulting from mass action kinetics, the corresponding \( \sigma(s) \) function does not have to be in general monotonic over \( k_r \in [k_r^\perp, k_r^\uparrow] \) for all states \( s \). This makes the computation of local extremes with respect to \( k_r \) more complex, though still tractable. In the following let us assume the space of perturbations \( P = [k_r^\perp, k_r^\uparrow] \times \mathbb{P}' \) is be decomposed along the \( k_r \) axis.

The key idea is for each state \( s \) to be able to efficiently decompose \( P \) into subspaces \( P = P_1 \cup \ldots \cup P_n \), such that for each \( P_i \) the function \( \sigma(s) \) over \( P_i \) is monotonic and then use the original method. The problem is that a computation of such a strict decomposition into monotonic subspaces is computationally demanding. Therefore we use a simplification, by off-line functional analysis we identify properties of \( \sigma \) functions for a given class of reaction kinetics and then obtain a partial decomposition of \( P \) based on function derivations into subspaces where monotonicity is guaranteed. For the remaining subspaces \( P_j \), where monotonicity of \( \sigma \) is not guaranteed, we employ a less accurate approximation.

We decompose the function \( \sigma(s) \) over \( P_j \) into functions \( \alpha_k^{s,P_j} \) and \( \beta_l^{s,P_j} \) such that:

\[
\sigma(s) = \sum_{k=1}^{K} \alpha_k^{s,P_j} - \sum_{l=1}^{L} \beta_l^{s,P_j}
\]

and each \( \alpha_k^{s,P_j} \) and \( \beta_l^{s,P_j} \) is monotonic. This allows us to use the original method to compute the maximum and minimum of the functions \( \alpha_k^{s,P_j} \) and \( \beta_l^{s,P_j} \) over the interval \( P_j \), denoted as \( \max(\alpha_k^{s,P_j}) \), \( \min(\alpha_k^{s,P_j}) \), \( \max(\beta_l^{s,P_j}) \) and \( \min(\beta_l^{s,P_j}) \), respectively. Note that this decomposition can be easily obtained from the definition of the rate function \( f_r \). Now the maximum and minimum of \( \sigma(p,s) \) over \( P_j \) can be approximated in the following way:

\[
\max\{\sigma(p,s) \mid p \in P_j\} \leq \sum_{k=1}^{K} \max(\alpha_k^{s,P_j}) - \sum_{l=1}^{L} \min(\beta_l^{s,P_j})
\]

\[
\min\{\sigma(p,s) \mid p \in P_i\} \geq \sum_{k=1}^{K} \min(\alpha_k^{s,P_j}) - \sum_{l=1}^{L} \max(\beta_l^{s,P_j}).
\]

This approximation increases the inaccuracy of parameterised uniformisation, however, the subspaces \( P_j \) where the monotonicity of \( \sigma(s) \) is not guaranteed are usually small and, together with perturbation space decomposition introduced in the following section, they keep on getting smaller. Hence, the additional inaccuracy of the presented extension is manageable. Despite the fact that the time demands of this approximation are orders of magnitudes lower than other numerical methods computing maximum/minimum of \( \sigma(s) \) over \( P_i \), they still significantly slow down the computation of parameterised uniformisation.

The aforementioned parameterised uniformisation can be straightforwardly employed also for backward transient analysis. It means that we can efficiently compute the vectors \( \tau^C_{\perp,A,t} \) and \( \tau^C_{\top,A,t} \) such that for the given set of states \( A \) and each state \( s \in S \) the following holds:

\[
\tau^C_{\top,A,t}(s) \geq \max\{\tau^C_{\top,A,t}(s) \mid C_p \in C\} \land \tau^C_{\perp,A,t}(s) \leq \min\{\tau^C_{\top,A,t}(s) \mid C_p \in C\}
\]

Once we know how to compute the vectors \( \tau^C_{\top,A,t} \) and \( \tau^C_{\perp,A,t} \) the global model checking technique for non-parameterised CTMCs can be directly employed. To obtain the vectors \( \overline{\text{Prov}}^C_{\top}, \overline{\text{Prov}}^C_{\perp}, \overline{\text{Exp}}^C_{\top} \) and \( \overline{\text{Exp}}^C_{\perp} \) satisfying Equation 2, it is sufficient to replace the backward transient distribution \( \tau^C_{\top,A,t} \) by
the vectors \( \tau^C_{\top} \) and \( \tau^C_{\bot} \). However for a general class of user-defined post-processing functions \( \text{Post} \), the vectors \( \pi^C_{\top} \) and \( \pi^C_{\bot} \) cannot be directly used to compute values of \( \text{Post}^C_\top(s, t) = \text{Post}(\pi^C_{\top}) \) nor \( \text{Post}^C_\bot(s, t) = \text{Post}(\pi^C_{\bot}) \) that would satisfy Equation 3, since there is no guarantee about the projective properties of the function \( \text{Post} \).

Now we show the main idea of computing \( \text{Post}^C_\top(s, t) \) and \( \text{Post}^C_\bot(s, t) \) for the post-processing function \( \text{Post} \) defined as the mean quadratic deviation of a probability distribution. This function allows us to quantify and characterise noise in different variants of signalling pathways that are studied in the second case study. The post-processing function is defined as \( \text{Post}(\pi) = \sum_{s \in S} |s(A) - \text{mean}(\pi, A)|^2 \cdot \pi(s), \) where \( s(A) \) gives the population of \( A \) in state \( s \) and \( \text{mean}(\pi, A) \) is the mean of the distribution \( \pi \) defined as \( \text{mean}(\pi, A) = \sum_{s \in S} s(A) \cdot \pi(s) \).

Let us suppose we have an upper and lower bound on the probability distribution \( \pi_\top, \pi_\bot \) obtained by the parameterised uniformisation. It means that \( \forall C_p \in C \) and \( \forall s \in S \), \( \pi_\bot(s) \leq \pi^C(s) \leq \pi_\top(s) \). To find the maximal value \( \max \{ \text{Post}(\pi_p^\top) \mid C_p \in C \} \) means to find the distribution \( \pi^\max \) such that \( \sum_{s \in S} \pi^\max(s) = 1, \forall s \in S \), \( \pi_\bot(s) \leq \pi^\max(s) \leq \pi_\top(s) \) and the probability mass in \( \pi^\max \) is distributed with the farthest distance from the mean. Clearly, such a distribution has a maximal mean quadratic deviation. Note that the number of distributions satisfying the first two conditions is uncountable. Thus we cannot employ a direct search strategy.

Our searching strategy builds on the observation that only the distributions that localise most of the mass as close as possible to the mean are considered. This leads again to a piece-wise quadratic function. It is also important to note that the perturbation space decomposition presented in the next section allows us to obtain the values \( \text{Post}^C_\top(s, t) \) and \( \text{Post}^C_\bot(s, t) \) with the desired precision.

**VI Perturbation Space Decomposition Strategy**

For the sake of simplicity, we present the parametric decomposition strategy only on the computation of \( \pi^C_{\top, s, t} \) and \( \tau^C_{\top, s, t} \) for \( * \in \{ \top, \bot \} \) and \( \mathbf{P} \), since it can be easily extended to the computation of \( D^C_{\Phi, \mathbf{P}, \star} \) for any formula \( \Phi \). The key part of the parametric decomposition is to decide when the inspected subspace should be further decomposed. The condition for the decomposition is different for \( \pi^C_{\top, s, t} \) and \( \tau^C_{\top, s, t} \). Since the vector \( \pi^C_{\top, s, t} \) gives us the transient probability distribution from the state \( s \) that is further used to compute \( D^C_{\Phi, \mathbf{P}, \star} \), we consider the following condition. The space \( \mathbf{P} \) is decomposed if during the computation of parameterised uniformisation in an iteration \( i \) it holds that:

\[
\sum_{k=1}^{||S||} \pi^C_{\top, s,i}(s_k) - \sum_{k=1}^{||S||} \pi^C_{\bot, s,i}(s_k) > \text{ERR}
\]

where \( \pi^C_{\star, s,i} \) denotes the corresponding approximation of \( \pi^C_{\star, s,0} \cdot (Q^\text{uni}(C))^i \).

In contrast to \( \pi^C_{\top, s, t} \), the value \( \tau^C_{\top, s, t}(s) \) for each state \( s \in S \) is further used to \( D^C_{\Phi, \mathbf{P}, \star} \) and thus we consider a different condition. The space \( \mathbf{P} \) is decomposed if during the computation of parameterised uniformisation in an iteration \( i \) for any state \( s \) it holds that:

\[
\tau^C_{\top, s, t}(s) - \tau^C_{\bot, s, t}(s) > \text{ERR}.
\]
If the decomposition takes place we cancel the current computation and decompose the perturbation space $P$ to $n$ subspaces such that $P = P_1 \cup \ldots \cup P_n$. Each subspace $P_j$ defines a new set of CTMCs $C_j = \{ C_j | j \in P_j \}$ that is independently processed in a new computation branch. Note that we could reuse the previous computation and continue from the iteration $i - 1$. However, the most significant part of the error is usually cumulated during the previous iterations and thus the decomposition would have only a negligible impact on error reduction.

A minimal decomposition with respect to the perturbation space $P$ defines a minimal number of subspaces $m$ such that $P = P_1 \cup \ldots \cup P_m$ and for each subspace $P_j$, where $1 \leq j \leq m$, it holds that $Err_{\Phi, P_j}^{C,s} \leq Err$, where $Err_{\Phi, P_j}^{C,s} = D_{\Phi, P_j}^C, \tau - D_{\Phi, P_j}^C, \bot$. Note that the existence of such decomposition is guaranteed only if the evaluation function $D_{\Phi}^C$ over $P$ is continuous. If the evaluation function is continuous there can exist more than one minimal decomposition. However, it cannot be straightforwardly found. To overcome this problem we have considered and implemented several heuristics allowing to iteratively compute a decomposition satisfying the following: (1) it ensures the required error bound whenever $D_{\Phi}^C$ over $P$ is continuous, (2) it guarantees the refinement termination in the situation where $D_{\Phi}^C$ over $P$ is not continuous and the discontinuity causes that $Err$ cannot be achieved. To ensure the termination an additional parameter has to be introduced as a lower bound on the subspace size. Hence this parameter provides a supplementary termination criterion.

References

[1] Kwiatkowska M, Norman G, Parker D (2011) PRISM 4.0: Verification of probabilistic real-time systems. In: Computer Aided Verification. Springer, volume 6806 of LNCS, pp. 585–591.

[2] Kwiatkowska M, Norman G, Parker D (2007) Stochastic model checking. In: Formal Methods for Performance Evaluation, Springer, volume 4486 of LNCS. pp. 220–270. doi:10.1007/978-3-540-72522-0_6.

[3] Fox BL, Glynn PW (1988) Computing Poisson probabilities. Commun ACM 31: 440–445.

[4] Munsky B, Khammash M (2006) The finite state projection algorithm for the solution of the chemical master equation. The Journal of chemical physics 124: 044104.

[5] Henzinger TA, Mateescu M, Wolf V (2009) Sliding Window Abstraction for Infinite Markov Chains. In: Computer Aided Verification, Springer, volume 5643 of LNCS. pp. 337-352. doi:10.1007/978-3-642-02658-4_27.

[6] Didier F, Henzinger TA, Mateescu M, Wolf V (2009) Fast Adaptive Uniformization of the Chemical Master Equation. In: High Performance Computational Systems Biology. IEEE Computer Society, pp. 118–127.

[7] Zhang J, Watson LT, Cao Y (2009) Adaptive aggregation method for the chemical master equation. International Journal of Computational Biology and Drug Design 2: 134-148.

[8] van Moorsel APA, Sanders WH (1994) Adaptive uniformization. ORSA Communications in Statistics: Stochastic Models, vol 10, no 3 : 619–648.

[9] Baier C, Haverkort B, Hermanns H, Katoen JP (2000) Model Checking Continuous-Time Markov Chains by Transient Analysis. In: Computer Aided Verification, Springer, volume 1855 of LNCS. pp. 358-372. doi:10.1007/10722167_28.

[10] Kwiatkowska M, Norman G, Pacheco A (2006) Model Checking Expected Time and Expected Reward Formulae with Random Time Bounds. Computers & Mathematics with Applications 51: 305 - 316.
[11] Brim L, Češka M, Dražan S, Safránek D (2013) Exploring parameter space of stochastic biochemical systems using quantitative model checking. In: Computer Aided Verification. Springer Berlin Heidelberg, volume 8044 of LNCS, pp. 107-123. doi:10.1007/978-3-642-39799-8_7.