Syntax-Guided Optimal Synthesis for Chemical Reaction Networks*

Luca Cardelli[1,2], Milan Češka[3] Martin Fränzle[4], Marta Kwiatkowska[2], Luca Laurenti[2], Nicola Paoletti[5], and Max Whitby[2]

[1] Microsoft Research Cambridge, UK
[2] Department of Computer Science, University of Oxford, UK
[3] Faculty of Information Technology, Brno University of Technology, Czech Republic
[4] Dept. of Computer Science, Carl von Ossietzky Universität Oldenburg, Germany
[5] Department of Computer Science, Stony Brook University, USA

Abstract. We study the problem of optimal syntax-guided synthesis of stochastic Chemical Reaction Networks (CRNs) that plays a fundamental role in design automation of molecular devices and in the construction of predictive biochemical models. We propose a sketching language for CRNs that concisely captures syntactic constraints on the network topology and allows its under-specification. Given a sketch, a correctness specification, and a cost function defined over the CRN syntax, our goal is to find a CRN that simultaneously meets the constraints, satisfies the specification and minimizes the cost function. To ensure computational feasibility of the synthesis process, we employ the Linear Noise Approximation allowing us to encode the synthesis problem as a satisfiability modulo theories problem over a set of parametric Ordinary Differential Equations (ODEs). We design and implement a novel algorithm for the optimal synthesis of CRNs that employs almost complete refutation procedure for SMT over reals and ODEs, and exploits a meta-sketching abstraction controlling the search strategy. Through relevant case studies we demonstrate that our approach significantly improves the capability of existing methods for synthesis of biochemical systems and paves the way towards their automated and provably-correct design.

1 Introduction

Chemical Reaction Networks (CRNs) are a versatile language widely used for modelling and analysis of biochemical systems. The power of CRNs derives from the fact that they provide a compact formalism equivalent to Petri nets [42], Vector Addition Systems (VAS) [36] and distributed population protocols [4].

CRNs also serve as a high-level *programming language* for molecular devices [49, 14] in systems and synthetic biology. Motivated by numerous potential applications ranging from smart therapeutics to biosensors, the construction of

*This work has been partially supported by the Czech Grant Agency grant No. GA16-17538S (M. Češka), Royal Society professorship, and EPSRC Programme on Mobile Autonomy (EP/M019918/1).

CRNs that exhibit prescribed dynamics is a major goal of synthetic biology [21, 17, 52]. Formal verification methods are now commonly embodied in the design process of biological systems [32, 34, 40] in order to reason about their correctness and performance. However, there is still a costly gap between the design and verification process, exacerbated in cases where stochasticity must be considered, which is typically the case for molecular computation. Indeed, automated synthesis of *stochastic CRNs* is generally limited to the estimation or synthesis of rate parameters [20, 53], which neglect the network structure, and suffers from scalability issues [23].

Current research efforts in design automation aim to eliminate this gap and address the problem of *program synthesis* – automatic construction of programs from high-level specifications. The field of syntax-guided program synthesis [1] has made tremendous progress in recent years, based on the idea of supplementing the specification with a syntactic template that describes a high-level structure of the program and constrains the space of allowed programs. Applications range from bit-streaming programming [47] and concurrent data structures [46], to computational biology [37]. Often not only the correctness of synthesized programs is important, but also their optimality with respect to a given cost [8].

In this paper we consider the problem of optimal syntax-guided synthesis of CRNs. We work in the setting of *program sketching* [47], where the template is a partial program with holes (incomplete information) that are automatically resolved using a constraint solver. We define a sketching language for CRNs that allows designers to not only capture the high-level topology of the network and known dependencies among particular species and reactions, but also to compactly describe parts of the CRN where only limited knowledge is available or left unspecified (partially specified) in order to examine alternative topologies. A *CRN sketch* is therefore a parametric CRN, where the parameters can be unknown species, (real-valued) rates or (integer) stoichiometric constants. Our sketching language is well-suited for biological systems, where partial knowledge and uncertainties due to noisy or imprecise measurements are very common. We associate to a sketch a *cost* function that captures the structural complexity of the CRNs and reflects the cost of physically implementing it using DNA [14].

Traditionally, the dynamical behaviour of a CRN is represented as a *deterministic* time evolution of average species concentrations, described by a set of Ordinary Differential Equations (ODEs), or as a discrete-state *stochastic* process solved through the Chemical Master Equation (CME) [51]. Given the importance of faithfully modelling stochastic noise in biochemical systems [27, 5], we focus on the the (continuous) Linear Noise Approximation (LNA) of the CME [51, 28]. It describes the time evolution of expectation and variance of the species in terms of ODEs, thus capturing the stochasticity intrinsic in CRNs, but, in contrast to solving the CME, scales well with respect to the molecular counts.

We can therefore represent the stochastic behaviour of a sketch as a set of parametric ODEs, which can be adequately solved as a satisfiability modulo theories (SMT) problem over the reals with ODEs. For this purpose, we employ the SMT solver iSAT(ODE) [26] that circumvents the well-known undecidability

of this theory by a procedure generating either a certificate of unsatisfiability, or a solution that is precise up to an arbitrary user-defined precision.

To specify the desired *temporal behaviour* of the network, we support constraints about the expected number and variance of molecules, and, crucially, their derivatives over time. This allows us, for instance, to formalise that a given species shows a specific number of oscillations or has higher variability than another species, thus providing greater expressiveness compared to simple reachability specifications or temporal logic.

We therefore formulate and provide a solution to the following problem. For a given CRN sketch, a formal specification of the required temporal behaviour and a cost function, we seek a sketch instantiation (a concrete CRN) that satisfies the specification and minimizes the cost. The optimal solution for a given sketch is computed using the *meta-sketch* abstraction for CRNs inspired by [8]. It combines a representation of the syntactic search space with the cost function and defines an ordered set of sketches. This cost-based ordering allows us to effectively prune the search space during the synthesis process and guide the search towards the minimal cost.

In summary, this paper makes the following contributions:

- We propose the first sketching language for CRNs that supports partial specifications of the topology of the network and structural dependencies among species and reactions.
- We formulate a novel optimal synthesis problem that, thanks to the LNA interpretation of stochastic dynamics, can be solved as an almost complete decision/refutation problem over the reals involving parametric ODEs. In this way, our approach offers superior scalability with respect to the size of the system and the number of parameters and, crucially, supports the synthesis of the CRN structure and not just of rate parameters.
- We design a new synthesis algorithm that builds on the meta-sketch abstraction, ensuring the optimality of the solution, and the SMT solver iSAT.
- We develop a prototype implementation of the algorithm and evaluate the usefulness and performance of our approach on three case studies, demonstrating the feasibility of synthesising networks with complex dynamics in a matter of minutes.

We stress that CRNs provide not just a programming language for bio-systems, but a more general computational framework. In fact, CRNs are formally equivalent to population protocols and Petri nets. As a consequence, our methods enable effective program synthesis also in other non-biological domains [3].

Related work. In the context of syntax-guided program synthesis (SyGuS) and program sketching, SMT-based approaches such as counter-example guided inductive synthesis [48] were shown to support the synthesis of deterministic programs for a variety of challenging problems [46, 8]. Sketching for probabilistic programs is presented in [43], together with a synthesis algorithm that builds on stochastic search and approximate likelihood computation. A similar approach appears in [31, 11], where genetic algorithms and probabilistic model checking

are used to synthesise probabilistic models from model templates (an extension of the PRISM language [38]) and multi-objective specifications. SyGuS has also been used for data-constrained synthesis, as in [37, 45, 24], where (deterministic) biological models are derived from gene expression data.

A variety of methods exist for estimating and synthesising rate parameters of CRNs, based on either the deterministic or stochastic semantics [35, 6, 53, 10, 20, 41, 2]. In contrast, our approach supports the synthesis of network structure and (uniquely) employs LNA.

Synthesis of CRNs from input-output functional specifications is considered in [23], via a method comprising two separate stages: (1) SMT-based generation of qualitative CRN models (candidates), and (2) for each candidate, parameter estimation of a parametric continuous time Markov chain (pCTMC). In contrast to our work, [23] do not consider solution optimality and require solving an optimisation problem for each concrete candidate on a pCTMC whose dimension is exponential in the number of molecules, making synthesis feasible only for very small numbers of molecules. On the other hand, our approach has complexity independent of the initial molecular population.

In [18], authors consider the problem of comparing CRNs of different size. They develop notions of bisimulations for CRNs in order to map a complex CRN into a simpler one, but with similar dynamical behaviour. Our optimal synthesis algorithm automatically guarantees that the synthesized CRN has the minimal size among all the CRNs consistent with the specification and the sketch.

2 Sketching Language for Chemical Reaction Networks

In this section, we introduce CRNs and the sketching language for their design.

2.1 Chemical Reaction Networks

CRN Syntax. A *chemical reaction network (CRN)* $\mathcal{C} = (\Lambda, \mathcal{R})$ is a pair of finite sets, where Λ is a set of *species*, $|\Lambda|$ denotes its size, and $\mathcal{R}$ is a set of reactions. Species in Λ interact according to the reactions in $\mathcal{R}$. A *reaction* $\tau \in \mathcal{R}$ is a triple $\tau = (r_\tau, p_\tau, k_\tau)$, where $r_\tau \in \mathbb{N}^{|\Lambda|}$ is the *reactant complex*, $p_\tau \in \mathbb{N}^{|\Lambda|}$ is the *product complex* and $k_\tau \in \mathbb{R}_{>0}$ is the coefficient associated with the rate of the reaction. r_τ and p_τ represent the stoichiometry of reactants and products. Given a reaction $\tau_1 = ([1,1,0],[0,0,2],k_1)$, we often refer to it as $\tau_1 : \lambda_1 + \lambda_2 \xrightarrow{k_1} 2\lambda_3$. The *state change* associated to τ is defined by $\upsilon_\tau = p_\tau - r_\tau$. For example, for τ_1 as above, we have $\upsilon_{\tau_1} = [-1,-1,2]$. The initial condition of a CRN is given by a vector of initial populations $x_0 \in \mathbb{N}^{|\Lambda|}$. A *chemical reaction system* (CRS) $C = (\Lambda, \mathcal{R}, x_0)$ is a tuple where (Λ, R) is a CRN and $x_0 \in \mathbb{N}^{|\Lambda|}$ represents its initial condition.

CRN semantics. Under the usual assumption of mass action kinetics, the *stochastic* semantics of a CRN is generally given in terms of a discrete-state, continuous-time Markov process (CTMC) $(X(t), t \geq 0)$ [28], where the states, $x \in \mathbb{N}^{|\Lambda|}$, are vectors of molecular counts. Such a representation is accurate, but not scalable

in practice because of the state space explosion problem [39, 34]. An alternative *deterministic* model describes the evolution of the concentrations of the species as the solution $\Phi : \mathbb{R}_{\geq 0} \to \mathbb{R}^{|\Lambda|}$ of the following ODEs (the so called rate equations) [13]:

$$\frac{d\Phi(t)}{dt} = F(\Phi(t)) = \sum_{\tau \in \mathcal{R}} \upsilon_\tau \cdot (k_\tau \prod_{S \in \Lambda} \Phi_S^{r_{S,\tau}}(t)) \tag{1}$$

where Φ_S and $r_{S,\tau}$ are the components of vectors Φ and r_τ relative to species S. However, such a model does not take into account the stochasticity intrinsic in molecular interactions. In this paper, we work with the Linear Noise Approximation (LNA) [51, 16, 28], which describes the stochastic behaviour of a CRN in terms of a Gaussian process Y converging in distribution to X [28, 9]. For a CRS $\mathcal{C} = (\Lambda, \mathcal{R}, x_0)$ contained in a system of volume N, we define $Y = N \cdot \Phi + \sqrt{N} \cdot Z$, where Φ is the solution of the rate equations (Eqn 1) with initial condition $\Phi(0) = \frac{x_0}{N}$. Z is a zero-mean Gaussian process with variance $C[Z(t)]$ described by

$$\frac{\mathrm{d}C[Z(t)]}{\mathrm{d}t} = J_F(\Phi(t))C[Z(t)] + C[Z(t)]J_F^T(\Phi(t)) + W(\Phi(t)) \tag{2}$$

where $J_F(\Phi(t))$ is the Jacobian of $F(\Phi(t))$, $J_F^T(\Phi(t))$ its transpose version, and $W(\Phi(t)) = \sum_{\tau \in \mathcal{R}} \upsilon_\tau {\upsilon_\tau}^T k_\tau \prod_{S \in \Lambda} (\Phi_S)^{r_{S,\tau}}(t)$. Y is a Gaussian process with expectation $E[Y(t)] = N\Phi(t)$ and covariance matrix $C[Y(t)] = NC[Z(t)]$. As a consequence, for any $t \in \mathbb{R}_{\geq 0}$, the distribution of $Y(t)$ is fully determined by its expectation and covariance. These are computed by solving the ODEs in Eqn. 1-2, and thus avoiding the state space exploration. We denote by $[\![\mathcal{C}]\!]_N = (E[Y], C[Y])$ the solution of these equations for CRS $\mathcal{C}$ in a system of size N, henceforth called the *LNA model*. By using the LNA we can consider stochastic properties of CRNs whilst maintaining scalability comparable to that of the deterministic model [16]. In fact, the number of ODEs required for LNA is quadratic in the number of species and independent of the molecular counts.

2.2 CRN sketching language

CRN sketches are defined in a similar fashion to concrete CRNs, with the main difference being that species, stoichiometric constants and reaction rates are specified as unknown *variables*. The use of variables considerably increases the expressiveness of the language, allowing the modeller to specify *additional constraints* over them. Constraints facilitate the representation of key background knowledge of the underlying network, e.g. that a reaction is faster than another, or that it consumes more molecules than it produces.

Another important feature is that reactants and products of a reaction are lifted to *choices* of species (and corresponding stoichiometry). In this way, the modeller can explicitly incorporate in the reaction a set of admissible alternatives, letting the synthesiser resolve the choice.

Further, a sketch distinguishes between *optional* and *mandatory* reactions and species. These are used to express that some elements of the network *might*

be present and that, on the other hand, other elements *must* be present. Our sketching language is well suited for synthesis of biological networks: it allows expressing key domain knowledge about the network, and, at the same time, it allows for network under-specification (holes, choices and variables). This is crucial for biological systems, where, due to inherent stochasticity or noisy measurements, the knowledge of the molecular interactions is often partial.

Definition 1 (Sketching language for CRNs)**.** *A* CRN sketch *is a tuple* $\mathcal{S} = (\Lambda, \mathcal{R}, \mathsf{Var}, \mathsf{Dec}, \mathsf{Ini}, \mathsf{Con})$*, where:*

- $\Lambda = \Lambda_m \cup \Lambda_o$ *is a finite set of* species*, where* Λ_m *and* Λ_o *are sets of mandatory and optional species, respectively.*
- $\mathsf{Var} = \mathsf{Var}_\Lambda \cup \mathsf{Var}_c \cup \mathsf{Var}_r$ *is a finite set of* variable names*, where* Var_Λ*,* Var_c *and* Var_r *are sets of* species, coefficient and rate variables*, respectively.*
- Dec *is a finite set of* variable declarations*. Declarations bind variable names to their respective domains of evaluation and are of the form* $x : D$*, where* $x \in \mathsf{Var}$ *and* D *is the domain of* x*. Three types of declaration are supported:*
  - Species*, where* $x \in \mathsf{Var}_\Lambda$ *and* $D \subseteq \Lambda$ *is a finite non-empty set of species.*
  - Stoichiometric coefficients*, where* $x \in \mathsf{Var}_c$ *and* $D \subseteq \mathbb{N}$ *is a finite non-empty set of non-negative integers.*
  - Rate parameters*, where* $x \in \mathsf{Var}_r$ *and* $D \subseteq \mathbb{R}_{\geq 0}$ *is a bounded set of non-negative reals.*
- Ini *is the* set of initial states*, that is, a predicate on variables* $\{\lambda_0\}_{\lambda \in \Lambda}$ *describing the initial number of molecules for each species.*
- Con *is a finite set of* additional constraints*, specified as quantifier-free formulas over* Var*.*
- $\mathcal{R} = \mathcal{R}_m \cup \mathcal{R}_o$ *is a finite set of* reactions*, where* $\mathcal{R}_m$ *and* $\mathcal{R}_o$ *are sets of mandatory and optional reactions, respectively. As for a concrete CRNs, each* $\tau \in \mathcal{R}$ *is a triple* $\tau = (r_\tau, p_\tau, k_\tau)$*, where in this case* $k_\tau \in \mathsf{Var}_r$ *is a rate variable; the reaction complex* r_τ *and the product complex* p_τ *are* sets of reactants and products*, respectively. A* reactant $R \in r_\tau$ (product $P \in p_\tau$) is a finite choice of species and coefficients*, specified as a (non-empty) set* $R = \{c_i \lambda_i\}_{i=1,\ldots,|R|}$*, where* $c_i \in \mathsf{Var}_c$ *and* $\lambda_i \in \mathsf{Var}_\Lambda$*. We denote with* f_{r_τ} *the* uninterpreted choice function *for the reactants of* τ*, that is, a function* $f_{r_\tau} : r_\tau \to \mathsf{Var}_c \times \mathsf{Var}_\Lambda$ *such that* $f_{r_\tau}(R) \in R$ *for each* $R \in r_\tau$*. The choice function for products,* f_{p_τ}*, is defined equivalently.*

As an example, reaction $\tau = (\{\{c_1\lambda_1, c_2\lambda_2\}, \{c_3\lambda_3\}\}, \{\{c_4\lambda_4, c_5\lambda_5\}\}, k)$ is preferably written as $\{c_1\lambda_1, c_2\lambda_2\} + c_3\lambda_3 \xrightarrow{k} \{c_4\lambda_4, c_5\lambda_5\}$, using the shortcut $c_3\lambda_3$ to indicate the single-option choice $\{c_3\lambda_3\}$. A possible concrete choice function for the reactants of τ is the function $\overline{f_{r_\tau}} = \{\{c_1\lambda_1, c_2\lambda_2\} \mapsto c_1\lambda_1, \{c_3\lambda_3\} \mapsto c_3\lambda_3\}$ that chooses option $c_1\lambda_1$ as first reactant.

Holes and syntactic sugar. Unknown information about the network can be also expressed using *holes*, i.e. portions of the model left "unfilled" and resolved by the synthesiser. Holes, denoted with ?, are implicitly encoded through sketch variables. To correctly interpret holes, we assume default domains, $D_r \subseteq \mathbb{R}$

bounded and $D_c \subseteq \mathbb{N}$ finite, for rate and coefficient variables, respectively. We also support the implicit declaration of variables, as shown in Example 1.

The following example illustrates the proposed sketching language and the optimal solution obtained using our synthesis algorithm introduced in Section 4.

Example 1 (Bell shape generator)**.** *For a given species K, our goal is to synthesize a CRN such that the evolution of K, namely the expected number of molecules of K, has a bell-shaped profile during a given time interval, i.e. during an initial interval the population K increases, then reaches the maximum, and finally decreases, eventually dropping to 0. Table 1 (left) defines a sketch for the bell-shape generator inspired by the solution presented in [12].*

$\Lambda_m = \{K\}, \Lambda_o = \{A, B\}, \mathcal{R}_m = \{\tau_1, \tau_2\},$
$\mathcal{R}_o = \{\tau_3\}$, $\mathsf{Dec} = \{c_1, \ldots, c_4 : [0, 2],$
$k_1, k_2, k_3 : [0, 0.1]$, $\lambda_1, \lambda_2 : \{A, B\}\}$,
$\mathsf{Con} = \{\lambda_1 \neq \lambda_2,\ c_1 < c_2,\ c_3 > c_4\}$,
$\mathsf{Ini} = \{K_0 = 1 \wedge A_0 \in [0, 100] \wedge B_0 \in [0, 100]\}$

$\tau_1 = \lambda_1 + c_1 K \xrightarrow{k_1} c_2 K$
$\tau_2 = \{0, 1\}\lambda_2 + c_3 K \xrightarrow{k_2} ?\lambda_2 + c_4 K$
$\tau_3 = \emptyset \xrightarrow{k_3} \{\lambda_2, [1, 2]K\}$

$A + K \rightarrow^{56} 2K; \quad K + B \rightarrow^{43} 2B$

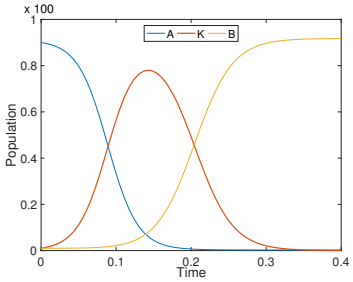


Table 1: **Left:** The sketch for bell-shape generator, with Volume $N = 100$. **Right:** CRN producing the bell-shape profile (species K) synthesized by our algorithm

This sketch reflects our prior knowledge about the control mechanism of the production/degradation of K. It captures that the solution has to have a reaction generating K (τ_1) and a reaction where K is consumed (τ_2). We also know that τ_1 requires a species, represented by variable λ_1, that is consumed by τ_1, and thus τ_1 will be blocked after the initial population of the species is consumed. An additional species, λ_2, different from λ_1, may be required. However, the sketch does not specify its role exactly: reaction τ_2 consumes either none or one molecule of λ_2 and produces an unknown number of λ_2 molecules, as indicated by the hole ?. There is also an optional reaction, τ_3, that does not have any reactants and produces either 1 molecule of λ_2 or between 1 and 2 molecules of K. The sketch further defines the mandatory and optional sets of species, the domains of the variables, and the initial populations of species. We assume the default domain $D_c = [0, 2]$, meaning that the hole ? can take values from 0 to 2. Note that many sketch variables are implicitly declared, e.g. term $[1, 2]K$ corresponds to $c'\lambda'$ with fresh variables $c' : [1, 2]$ and $\lambda' : \{K\}$.

Table 1 (right) shows the optimal CRN computed by our algorithm for the cost function given in Definition 3 and the bell-shape profile produced by the CRN.

We now characterise when a concrete network is a valid instantiation of a sketch.

Definition 2 (Sketch instantiation)**.** *A CRS $\mathcal{C} = (\Lambda_\mathcal{C}, \mathcal{R}_\mathcal{C}, x_0)$ is a valid instantiation of a sketch $\mathcal{S} = (\Lambda, \mathcal{R}, \mathsf{Var}, \mathsf{Dec}, \mathsf{Ini}, \mathsf{Con})$ if: $\mathsf{Ini}(x_0)$ holds; there exists an interpretation I of the variables in Var and choice functions such that:*

1. *all additional constraints are satisfied:* $I \models \bigwedge_{\phi \in \mathsf{Con}} \phi$,
2. *for each* $\tau \in \mathcal{R}_m$ *there is* $\tau' \in \mathcal{R}_\mathcal{C}$ *that* realises τ, *i.e.,* τ' *is obtained from* τ *by replacing variables and choice functions with their interpretation*[6], *and*
3. *for each* $\tau' \in \mathcal{R}_\mathcal{C}$ *there is* $\tau \in \mathcal{R}$ *such that* τ' *realises* τ;

and the following conditions hold:

4. *for each* $\tau' = (r_{\tau'}, p_{\tau'}, k_{\tau'}) \in \mathcal{R}_\mathcal{C}$: $k_{\tau'} > 0$ *and* $r_{\tau'} + p_{\tau'} > 0$
5. $\Lambda_m \subseteq \Lambda_\mathcal{C}$ *and* $\Lambda_\mathcal{C} \subseteq \Lambda_m \cup \Lambda_o$ *and*
6. *for each species* $A \in \Lambda_\mathcal{C}$ *there is* $r \in \mathcal{R}_\mathcal{C}$ *such that* A *appears in* r *as reactant or product.*

Such an interpretation is called consistent *for* $\mathcal{S}$. *For sketch* $\mathcal{S}$ *and consistent interpretation* I, *we denote with* $I(\mathcal{S})$ *the* instantiation of $\mathcal{S}$ through I. *We denote with* $L(\mathcal{S})$ *the set of valid instantiations of* $\mathcal{S}$.

Condition 4. states that there are no void reactions, i.e. having null rate ($k_{\tau'} = 0$), or having no reactants and products ($r_{\tau'} + p_{\tau'} = 0$). Further, condition 6. ensures that the concrete network contains only species occurring in some reactions.

Example 2. *A CRS* $\mathcal{C}_1 = \{\{A, B, K\}, \{\tau'_1, \tau'_2, \tau'_3\}, x_0\}$ *where*

$$\tau'_1 = A + K \xrightarrow{0.01} 2K \quad \tau'_2 = B + K \xrightarrow{0.1} 2B \quad \tau'_3 = \emptyset \xrightarrow{0.001} K,$$

with $x_0 = (A_0 = 100, B_0 = K_0 = 1)$ *is a valid instantiation of the bell shape sketch* $\mathcal{S}$ *from Example 1. Reactions* τ'_1, τ'_2 *and* τ'_3 *realise respectively reaction sketches* τ_1, τ_2 *and* τ_3. *The corresponding consistent interpretation is* $I = \{\lambda_1 \mapsto A, c_1 \mapsto 1, k_1 \mapsto 0.01, c_2 \mapsto 2, c'_1 \mapsto 1, \lambda_2 \mapsto B, c_3 \mapsto 1, k_2 \mapsto 0.1, H \mapsto 2, c_4 \mapsto 0, k_3 \mapsto 0.001, f_{p_{\tau_3}} \mapsto \{\{\lambda_2, [1,2]K\} \mapsto [1,2]K\}, c'_2 \mapsto 1\}$, *where* c'_i *is the* i*-th implicit stoichiometric variable and* H *is the only hole. The interpretation of* $f_{p_{\tau_3}}$ *indicates that the choice* $\{\lambda_2, [1,2]K\}$ *is resolved as* $[1,2]K$.

Since a sketch instantiation corresponds to a CRS, we remark that its behaviour is given by the LNA model. Similarly, as we will show in Section 4, the SMT encoding of a sketch builds on a symbolic encoding of the LNA equations.

3 Specification Language

We are interested in checking whether a CRN exhibits a given temporal profile. For this purpose, our specification language supports constraints about the expected number and variance of molecules, and, importantly, about their derivatives over time. This allows us, for instance, to synthesise a network where a given species shows a bell-shape profile (as in Example 1), or has variance greater than its expectation (considered in Section 5). Before explaining the specification language, we introduce the logical framework over which properties, together with CRN sketches, will be interpreted and evaluated.

[6] When τ' realises sketch reaction τ, its reactants $r_{\tau'}$ is a set of the form $\{c_R \lambda_R\}_{R \in r_\tau}$, i.e. containing a concrete reactant for each choice R. Then, this is readily encoded in the reactant vector form $r_{\tau'} \in \mathbb{N}^{|\Lambda|}$ as per CRN definition (see Section 2.1). Similar reasoning applies for products $p_{\tau'}$.

3.1 Satisfiability modulo ODEs

In syntax-guided synthesis, the synthesis problem typically reduces to an SMT problem [1]. Since we employ LNA, which generally involves non-linear ODEs, we resort to the framework of *satisfiability modulo ODEs* [25, 26, 30], which provides solving procedures for this theory that are sound and complete up to a user-specified precision. We stress that this framework allows for continuous encoding of the LNA equations, thus avoiding discrete approximations of its dynamics. Crucially, we can express arbitrary-order derivatives of the LNA variables, as these are smooth functions, and hence admit derivatives of all orders.

We employ the SMT solver iSAT(ODE) [25] that supports arithmetic constraint systems involving non-linear arithmetic and ODEs. The constraints solved are quantifier-free Boolean combinations of Boolean variables, arithmetic constraints over real- and integer-valued variables with bounded domains, and ODE constraints over real variables plus flow invariants. *Arithmetic constraints* are of the form $e_1 \sim e_2$, where $\sim \in \{<, \leq, =, \geq, >\}$ and $e_{1,2}$ are expressions built from real- and integer-valued variables and constants using functions from $\{+, -, \cdot, \sin, \cos, \mathrm{pow}_{\mathbb{N}}, \exp, \min, \max\}$. *ODE constraints* are time-invariant and given by $\frac{\mathrm{d}x}{\mathrm{d}t} = e$, where e is an expression as above[7] containing variables themselves defined by ODE constraints. *Flow invariant constraints* are of the form $x \leq c$ or $x \geq c$, with x being an ODE-defined variable and c being a constant. ODE constraints have to occur under positive polarity and are interpreted as first-order constraints on pre-values x and post-values x' of their variables, i.e., they relate those pairs (x, x') being connected by a trajectory satisfying $\frac{\mathrm{d}x}{\mathrm{d}t} = e$ and, if present, the flow invariant throughout.

Due to undecidability of the fragment of arithmetic addressed, iSAT(ODE) implements a sound, yet quantifiably incomplete, unsatisfiability check based on a combination of interval constraint propagation (ICP) for arithmetic constraints, safe numeric integration of ODEs, and conflict-driven clause learning (CDCL) for manipulating the Boolean structure of the formula. This procedure investigates "boxes", i.e. Cartesian products of intervals, in the solution space until it either finds a proof of unsatisfiability based on a set of boxes covering the original domain or finds some hull-consistent box [7], called a *candidate solution box*, with edges smaller than a user-specified width $\delta > 0$. While the interval-based unsatisfiability proof implies unsatisfiability over the reals, thus rendering the procedure sound, the report of a candidate solution box only guarantees that a slight relaxation of the original problem is satisfiable. Within this relaxation, all original constraints are first rewritten to equi-satisfiable inequational form $t \sim 0$, with $\sim \in \{>, \geq\}$, and then relaxed to the strictly weaker constraint $t \sim -\delta$. In that sense, iSAT and related algorithms [30, 50] provide reliable verdicts on either unsatisfiability of the original problem or satisfiability of its aforementioned δ-relaxation, and do in principle[8] always terminate with one of these two verdicts. Hence the name "δ-decidability" used by Gao et al. in [29].

[7] where we can additionally use non-total functions $/$, $\sqrt{}$ and ln.

[8] i.e., when considering the abstract algorithms using unbounded precision rather than the safe rounding employed in their floating-point based actual implementations.

3.2 Specification for CRNs

The class of properties we support are formulas describing a dynamical profile composed as a finite sequence of phases. Each phase i is characterised by an arithmetic predicate $\mathsf{pre\text{-}post}_i$, describing the system state at its start and end points (including arithmetic relations between these two), as well as by flow invariants (formula inv_i) pertaining to the trajectory observed during the phase. Formally, a specification φ comprising $M \geq 1$ phases is defined by

$$\varphi = \bigwedge_{i=1}^{M} \mathsf{inv}_i \wedge \mathsf{pre\text{-}post}_i \tag{3}$$

Note that entry as well as target conditions of phases can be expressed within $\mathsf{pre\text{-}post}_i$. Initial conditions are not part of the specification but, as explained in Section 4, the sketch definition.

CRS correctness. For a CRS $\mathcal{C}$, Volume N, and property φ, we are interested in checking whether $\mathcal{C}$ is *correct* with respect to φ, written $[\![\mathcal{C}]\!]_N \models \varphi$, i.e., whether $\mathcal{C}$ at Volume N exhibits the dynamic behavior required by φ. Since $[\![\mathcal{C}]\!]_N$ is a set of ODEs, this corresponds to checking whether $\hat{\varphi} \wedge \varphi_{[\![\mathcal{C}]\!]_N}$ is satisfiable, where $\varphi_{[\![\mathcal{C}]\!]_N}$ is an SMT formula encoding the set of ODEs given by $[\![\mathcal{C}]\!]_N$ and their higher-order derivatives[9] by means of the corresponding ODE constraints, and $\hat{\varphi}$ is the usual bounded model checking (BMC) unwinding of the step relation $\bigwedge_{i=1}^{M}(\mathit{phase} = i \Rightarrow \mathsf{inv}_i \wedge \mathsf{pre\text{-}post}_i) \wedge \mathit{phase}' = \mathit{phase} + 1$ encoding the phase sequencing and the pertinent phase constraints, together with the BMC target $\mathit{phase} = M$ enforcing all phases to be traversed. As this satisfiability problem is undecidable in general, we relax it to checking whether $\hat{\varphi} \wedge \varphi_{[\![\mathcal{C}]\!]_N}$ is δ-satisfiable in the sense of admitting a candidate solution box of width δ. In that case, we write $[\![\mathcal{C}]\!]_N \models^{\delta} \varphi$.

Example 3 (Specification for the bell-shape generator)**.** *The required bell-shaped profile for Example 1 can be formalized using a* 2*-phase specification as follows:*

$$\mathsf{inv}_1 \equiv E^{(1)}[K] \geq 0, \;\; \mathsf{pre\text{-}post}_1 \equiv E^{(1)}[K]' = 0 \;\wedge\; E[K]' > 30,$$
$$\mathsf{inv}_2 \equiv E^{(1)}[K] \leq 0, \;\; \mathsf{pre\text{-}post}_2 \equiv E[K]' \leq 1 \;\wedge\; T' = 1$$

where $E[K]$ is the expected value of species K and $E^{(1)}[K]$ its first derivative. T is the global time. Primed notation $(E[K]', E^{(1)}[K]', T')$ indicates the variable value at the end of the respective phase. Constraints inv_1 and inv_2 require, respectively, that $E[K]$ is not decreasing in the first phase, and not increasing in the second (and last) phase. $\mathsf{pre\text{-}post}_1$ states that, at the end of phase 1*, $E[K]$ is a local optimum ($E^{(1)}[K]' = 0$), and has an expected number of molecules greater than* 30*. $\mathsf{pre\text{-}post}_2$ states that, at the final phase, the expected number of molecules of K is at most* 1 *and that the final time is* 1*.*

This example demonstrates that we can reason over complex temporal specifications including, for instance, a relevant fragment of bounded metric temporal logic [44].

[9] Only the derivatives appearing in φ are included. These are encoded using the Faà di Bruno's formula [33].

4 Optimal Synthesis of Chemical Reaction Networks

In this section we formulate the optimal synthesis problem where we seek to find a concrete instantiation of the sketch (i.e. a CRN) that satisfies a given property and has a minimal cost. We further show the encoding of the problem using satisfiability modulo ODEs and present an algorithm scheme for its solution.

4.1 Problem Formulation

Before explaining our optimal synthesis problem, we first need to introduce the class of cost functions considered. A cost function G for a sketch $\mathcal{S}$ has signature $G : L(\mathcal{S}) \to \mathbb{N}$ and maps valid instantiations of $\mathcal{S}$ to a natural cost. A variety of interesting cost functions fit this description, and, depending on the particular application, the modeller can choose the most appropriate one. A special case is, for instance, the overall number of species and reactions, a measure of CRN complexity used in e.g. CRN comparison and reduction [19, 18]. Importantly, cost functions are defined over the structure of the concrete instantiation, rather than its dynamics. As we shall see, this considerably simplifies the optimisation task, since it leads to a finite set of admissible costs. In the rest of the paper, we consider the following cost function, which captures the structural complexity of the CRN and the cost of physically implementing it using DNA [49, 14].

Definition 3 (Cost function)**.** *For a sketch $\mathcal{S} = (\Lambda, \mathcal{R}, \mathsf{Var}, \mathsf{Dec}, \mathsf{Ini}, \mathsf{Con})$, we consider the cost function $G_{\mathcal{S}} : L(\mathcal{S}) \to \mathbb{N}$ that, for any CRS instantiation $\mathcal{C} = (\Lambda, \mathcal{R}) \in L(\mathcal{S})$, is defined as:*

$$G_{\mathcal{S}}(\mathcal{C}) = 3 \cdot (|\Lambda \cap \Lambda_o|) + \sum_{\tau \in \mathcal{R}_{\mathcal{C}}} \sum_{S \in \Lambda} 6 \cdot r_{S,\tau} + 5 \cdot p_{S,\tau}$$

where $r_{S,\tau}$ ($p_{S,\tau}$) is the stoichiometry of species S as reactant (product) of τ.

This cost function penalizes the presence of optional species (Λ_o) and the number of reactants and products in each reaction. It does not explicitly include a penalty for optional reactions, but this is accounted for through an increased total number of reactants and products. We stress that different cost functions can be used, possibly conditioned also on the values of reaction rates.

Problem 1 (Optimal synthesis of CRNs). Given a sketch S, cost function $G_{\mathcal{S}}$, property φ, Volume N and precision δ, the optimal synthesis problem is to find CRS $\mathcal{C}^* \in L(\mathcal{S})$, if it exists, such that $[\![\mathcal{C}^*]\!]_N \vDash^{\delta} \varphi$ and, for each CRS $\mathcal{C} \in L(\mathcal{S})$ such that $G_{\mathcal{S}}(\mathcal{C}) < G_{\mathcal{S}}(\mathcal{C}^*)$, it holds that $[\![\mathcal{C}]\!]_N \nvDash^{\delta} \varphi$.

An important characteristic of the sketching language and the cost function is that for each sketch $\mathcal{S}$ the set $\{G_{\mathcal{S}}(\mathcal{C}) \mid \mathcal{C} \in L(\mathcal{S})\}$ is finite. This follows from the fact that $\mathcal{S}$ restricts the maximal number of species and reactions as well as the maximal number of reactants and products for each reaction. Therefore, we can define for each sketch $\mathcal{S}$ the minimal cost $\mu_{\mathcal{S}}$ and the maximal cost $\nu_{\mathcal{S}}$.

Example 4. *It is easy to verify that the cost of the CRS $\mathcal{C}$ of Example 2, a valid instantiation of the bell-shape generator sketch $\mathcal{S}$, is $G_{\mathcal{S}}(\mathcal{C}) = 3 \cdot 2 + 6 \cdot 4 + 5 \cdot 5 = 55$, and that minimal and maximal costs of sketch $\mathcal{S}$ are, respectively, $\mu_{\mathcal{S}} = 3 \cdot 1 + 6 \cdot 2 + 5 \cdot 2 = 25$ and $\nu_{\mathcal{S}} = 3 \cdot 2 + 6 \cdot 5 + 5 \cdot 7 = 71$.*

We now define a meta-sketch abstraction for our sketching language that allows us to formulate an efficient optimal synthesis algorithm.

Definition 4 (Meta-sketch for CRNs)**.** *Given a sketch $\mathcal{S}$ and a cost function $G_{\mathcal{S}}$, we define the meta-sketch $\mathcal{M}_{\mathcal{S}} = \{\mathcal{S}(i) \mid \mu_{\mathcal{S}} \leq i \leq \nu_{\mathcal{S}}\}$, where $\mathcal{S}(i)$ is a sketch whose instantiations have cost smaller than i, i.e. $L(\mathcal{S}(i)) = \{C \in L(\mathcal{S}) \mid G_{\mathcal{S}}(C) < i\}$.*

A meta-sketch $\mathcal{M}_{\mathcal{S}}$ establishes a hierarchy over the sketch $\mathcal{S}$ in the form of an ordered set of sketches $\mathcal{S}(i)$. The ordering reflects the size of the search space for each $\mathcal{S}(i)$ as well as the cost of implementing the CRNs described by $\mathcal{S}(i)$. In contrast to the abstraction defined in [8], the ordering is given by the cost function and thus it can be directly used to guide the search towards the optimum.

4.2 Symbolic Encoding

Given a sketch of CRN $\mathcal{S} = (\Lambda, \mathcal{R}, \mathsf{Var}, \mathsf{Dec}, \mathsf{Ini}, \mathsf{Con})$, we show that the dynamics of $L(\mathcal{S})$, set of possible instantiations of $\mathcal{S}$, can be described symbolically by a set of parametric ODEs, plus additional constraints. These equations depend on the sketch variables and on the choice functions of each reaction, and describe the time evolution of mean and variance of the species.

For $S \in \Lambda, \lambda \in \mathsf{Var}$, we define the indicator function $\mathcal{I}_S(\lambda) = 1$ if $\lambda = S$, and 0 otherwise. For $S \in \Lambda$ and $\tau \in \mathcal{R}$, we define the following constants:

$$r_{S,\tau} = \sum_{\substack{R \in r_\tau \\ (c,\lambda) = f_{r_\tau}(R)}} c \cdot \mathcal{I}_S(\lambda), \qquad p_{S,\tau} = \sum_{\substack{P \in p_\tau \\ (c,\lambda) = f_{p_\tau}(P)}} c \cdot \mathcal{I}_S(\lambda), \qquad v_{S,\tau} = p_{S,\tau} - r_{S,\tau}$$

Note that these are equivalent to the corresponding coefficients for concrete CRNs, but now are parametric as they depend on the sketch variables. As for the LNA model of Section 2.1, symbolic expectation and variance together characterise the symbolic behaviour of sketch $\mathcal{S}$, given as the set of parametric ODEs $[\![\mathcal{S}]\!]_N = (N \cdot \Phi, N \cdot C[Z])$, for some Volume N.

The functions $\Phi(t)$ and $C[Z(t)]$ describe symbolically the time evolution of expected values and covariance of all instantiations of $\mathcal{S}$, not just of valid instantiations. We restrict to valid instantiations by imposing the following formula:

$$\mathsf{consist} \equiv \mathsf{Ini}(x_0) \wedge \bigwedge_{\phi \in \mathsf{Con}} \phi \ \wedge \bigwedge_{\tau \in \mathcal{R}_m} \neg \mathsf{void}(\tau) \ \wedge \bigwedge_{S \in \Lambda_m} \mathsf{used}(S)$$

which, based on Definition 2, states that initial state and additional constraints have to be met, all mandatory reactions must not be void, and all mandatory species must be "used", i.e. must appear in some (non-void) reactions. Note that we allow optional reactions to be void, in which case they are not included in the concrete network. Formally, $\mathsf{void}(\tau) \equiv \ (k_\tau = 0) \vee \sum_{S \in \Lambda}(r_{S,\tau} + p_{S,\tau}) = 0$ and $\mathsf{used}(S) \equiv \ \bigvee_{\tau \in \mathcal{R}} \neg \mathsf{void}(\tau) \wedge (r_{S,\tau} + p_{S,\tau}) > 0$.

Sketch correctness. Given an interpretation I consistent for $\mathcal{S}$, call Φ_I and $C[Z]_I$, the concrete functions obtained from Φ and $C[Z]$ by substituting variables and functions with their assignments in I. The symbolic encoding ensures that the

LNA model $[\![I(\mathcal{S})]\!]_N$ of CRS $I(\mathcal{S})$ (i.e. the instantiation of $\mathcal{S}$ through I, see Definition 2) is equivalent to $(\Phi_I, C[Z]_I)$.

With reference to our synthesis problem, this implies that the synthesis of a CRS $\mathcal{C}^*$ that satisfies a correctness specification φ from a sketch $\mathcal{S}$ corresponds to finding a consistent interpretation for $\mathcal{S}$ that satisfies φ. Similarly to the case for concrete CRSs, this corresponds to checking if $\hat{\varphi} \wedge \mathsf{consist} \wedge \varphi_{[\![\mathcal{S}]\!]_N}$ is δ-satisfiable for some precision δ, where $\hat{\varphi}$ is the BMC encoding of ϕ (see Section 3.2) and $\varphi_{[\![\mathcal{S}]\!]_N}$ is the SMT encoding of the symbolic ODEs given by $[\![\mathcal{S}]\!]_N$ and the corresponding derivatives.

Cost constraints. For a sketch $\mathcal{S}$ and cost $i \in \mathbb{N}$, the following predicate encodes the cost function of Definition 3 in order to restrict $\mathcal{S}$ into $\mathcal{S}(i)$, i.e. the sketch whose instantiations have cost smaller than i:

$$\mathsf{Con}_G(i) \equiv \left(3 \cdot \sum_{S \in \Lambda_o} \mathcal{I}(\mathsf{used}(S)) + \sum_{\tau \in \mathcal{R}} \mathcal{I}(\neg \mathsf{void}(\tau)) \cdot \sum_{S \in \Lambda} (6 \cdot r_{S,\tau} + 5 \cdot p_{S,\tau}) \right) < i$$

where $\mathcal{I}$ is the indicator function, and used and void are predicates defined above.

4.3 Algorithm Scheme for Optimal Synthesis

In Algorithm 1, we present an algorithm scheme for solving the optimal synthesis problem for CRNs. It builds on the meta-sketch abstraction described in Definition 4, which enables effective pruning of the search space through cost constraints, and the SMT-based encoding of Section 4.2, which allows for the automated derivation of meta-sketch instantiations (i.e. CRNs) that satisfy the specification and the cost constraints.

This scheme repeatedly invokes the SMT solver (δ-Solver) on the sketch encoding, and at each call the cost constraints are updated towards the optimal cost. We consider three approaches: 1) *top-down*: starting from the maximal cost $\nu_{\mathcal{S}}$, it solves meta-sketches with decreasing cost until no solution exists (UNSAT); 2) *bottom-up:* from the minimal cost $\mu_{\mathcal{S}}$, it increases the cost until a solution is found (SAT); 3) *binary search:* it bounds the upper estimate on the optimal solution using a SAT witness and the lower estimate with an UNSAT witness.

We further improve the algorithm by exploiting the fact that UNSAT witnesses can also be obtained at a lower precision δ_{init} ($\delta_{init} \gg \delta$), which consistently improves performance. Indeed, UNSAT outcomes are precise and thus valid for any precision. Note that the top-down strategy does not benefit from this speed-up since it only generates SAT witnesses.

At every iteration, variable i maintains the current cost. The solver is firstly called using the rough precision δ_{init} (line 3). If the solver returns SAT (potential false positive), we refine our query using the required precision δ (line 5). If this query is in turn satisfiable, then the solver also returns a candidate solution box M, where all discrete variables are instantiated to a single value and an interval smaller than δ is assigned to each real-valued variable. Function $\mathsf{getSoln}$ computes the actual sketch instantiation C^* as the centre point of M that δ-satisfies φ.

Algorithm 1 Generalised synthesis scheme

Require: Meta-sketch $\mathcal{M}_{\mathcal{S}}$, property φ, precision δ and initial precision δ_{init}
Ensure: C^* is a solution of Problem 1 if $\exists\, C \in L(\mathcal{M}_{\mathcal{S}}^{\omega}) : C \vDash^{\delta} \varphi$, otherwise $C^* = \mathsf{null}$

1: $i^{\top} \leftarrow \nu_{\mathcal{S}}$; $i^{\perp} \leftarrow \mu_{\mathcal{S}}$; $i \leftarrow g(i^{\perp}, i^{\top})$; $C^* \leftarrow \mathsf{null}$
2: **repeat**
3: $\quad SAT_1 \leftarrow \delta\text{-}\mathsf{Solver}(\mathcal{S}(i), \varphi, \delta_{init})$; $SAT_2 \leftarrow \mathsf{false}$
4: $\quad$ **if** SAT_1 **then**
5: $\quad\quad (M, SAT_2) \leftarrow \delta\text{-}\mathsf{Solver}(\mathcal{S}(i), \varphi, \delta)$
6: $\quad\quad$ **if** SAT_2 **then** $C^* = \mathsf{getSoln}(\mathcal{S}(i), M)$
7: $\quad\quad$ **else** $\delta_{init} = (\delta_{init} - \delta)/2$
8: $\quad (i^{\perp}, i^{\top}) \leftarrow f(i, i^{\perp}, i^{\top}, SAT_2, \mathsf{G}_{\mathcal{S}}(C^*))$; $i \leftarrow g(i^{\perp}, i^{\top})$
9: **until** $i^{\perp} \leq i^{\top}$
10: **return** C^*

The cost of C^* provides the upper bound on the optimal solution. If either query returns UNSAT, the current cost i provides the lower bound on the optimal solution. The second query being UNSAT implies that the rough precision δ_{init} produced a false positive, and thus it is refined for the next iteration (line 7).

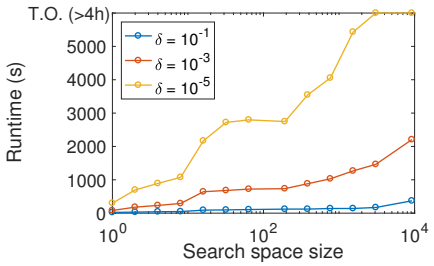
The actual search strategy used in Algorithm 1 is given by the functions f controlling how the upper ($i^{\top}$) and lower ($i^{\perp}$) bounds on the cost are updated and by g determining the next cost to explore. Note that such bounds ensure the termination of the algorithm (line 9). In the bottom-up approach, f "terminates" the search (i.e. causes $i^{\perp} > i^{\top}$) if SAT_2 is true (i.e. when the first SAT witness is obtained), otherwise f sets $(i^{\perp}, i^{\top}) \leftarrow (i+1, i^{\top})$ and g sets $i \leftarrow i^{\perp}$. In the top-down case, f terminates the search if SAT_2 is false (i.e. at the first UNSAT witness), otherwise it sets $(i^{\perp}, i^{\top}) \leftarrow (i^{\perp}, \mathsf{G}_{\mathcal{S}}(C^*)-1)$ and $i \leftarrow i^{\top}$, where $\mathsf{G}_{\mathcal{S}}(C^*)$ is the cost of CRN C^*. Binary search is obtained with f that updates $(i^{\perp}, i^{\top})$ to $(i^{\perp}, \mathsf{G}_{\mathcal{S}}(C^*) - 1)$ if $SAT_2 = \mathsf{true}$, to $(i+1, i^{\top})$ otherwise, and with g that updates i to $i^{\perp} + \lfloor (i^{\top} - i^{\perp})/2 \rfloor$.

5 Experimental Evaluation

We evaluate the usefulness and performance of our optimal synthesis method on three case studies, representative of important problems studied in biology: (1) the **bell-shape generator**, a component occurring in signaling cascades; (2) **Super Poisson**, where we synthesize CRN implementations of stochastic processes with prescribed levels of process noise; and (3) **Phosphorelay network**, where we synthesize CRNs exhibiting switch-like sigmoidal profiles, which is the biochemical mechanism underlying cellular decision-making, driving in turn development and differentiation.

We employ the solver iSAT(ODE) [25, 26][10], even if our algorithm supports any δ-solver. We ran preliminary experiments using the tool dReal [30], finding that iSAT performs significantly better on our instances. All experiments were

[10] Version r2806. Parameters: `--maxdepth=k` (`k` is the BMC unrolling depth) and `--ode-opts=--continue-after-not-reaching-horizon`.



Search Strategy	# iSAT calls	Total time (s)
bottom-up	7/1	2671/1732
top-down	1/6	4863/5612
binary-search	2/4	3440/3121

Table 2: Performance of bell-shape generator model. **Left**: runtimes for different precisions δ and discrete search space size. **Right**: optimal synthesis with different variants of Algorithm 1, fixed discrete search space size (1536) and $\delta = 10^{-3}$.

run on a server with a Intel Xeon CPU E5645 @2.40GHz processor (using a single core) and 24GB @1333MHz RAM.

Bell-Shape Generator. We use the example described in Examples 1 and 3, resulting in 8 parametric ODEs, as the main benchmark. The synthesised CRN is shown in Figure 1. In the first experiment, we evaluate the scalability of the solver with respect to precision δ and the size of the discrete search space, altered by changing the domains of species and coefficient variables of the sketch. We exclude cost constraints as they reduce the size of the search space. Runtimes, reported in Table 2 (left), correspond to a single call to iSAT with different δ values, leading to SAT outcomes in all cases. Note that the size of the continuous state space, given by the domains of rate variables, does not impose such a performance degradation, as shown in Table 3 (right) for a different model.

In the second experiment, we analyse how cost constraints and different variants of Algorithm 1 affect the performance of optimal synthesis. Table 2 (right) shows the number of iSAT calls with UNSAT/SAT outcomes (2nd column) and total runtimes without/with the improvement that attempts to obtain UNSAT witnesses at lower precision ($\delta_{init} = 10^{-1}$). Importantly, the average runtime for a single call to iSAT is significantly improved when we use cost constraints, since these reduce the discrete search space (between 216s and 802s with cost constraints, 1267s without). Moreover, results clearly indicate that UNSAT cases are considerably faster to solve, because inconsistent cost constraints typically lead to trivial UNSAT instances. This favours the bottom-up approach over the top-down. In this example, the bottom-up approach also outperforms binary-search, but we expect the opposite situation for synthesis problems with wider spectra of costs. As expected, we observe a speed-up when using a lower precision for UNSAT witnesses, except for the top-down approach.

Super Poisson. We demonstrate that our approach is able to synthesise a CRN that behaves as a stochastic process, namely, a super Poisson process having variance greater than its expectation. We formalise the behaviour on the interval $[0, 1]$ using a 1-phase specification as shown in Table 3 (left). For $N = 100$ we consider the sketch listed in Table 3 (center) where both reactions are mandatory, reflecting the knowledge that A is both produced and degraded.

$\mathsf{inv}_1 \equiv C[A] > E[A]$	$\Lambda = \{A, B\}, \Lambda_o = \{B\}, \lambda_1, \lambda_2 : \Lambda,$	Rate interval | Time (s)
$\mathsf{pre\text{-}post}_1 \equiv T' = 1$	$\mathcal{R}_m = \{\tau_1, \tau_2\}, A_0 = B_0 = 0,$	$[0,1]$ | 4
	$k_1, k_2 : [0, 100], c_1, c_2, c_3 : [0, 2]$	$[0,10]$ | 18
	$\tau_1 :\rightarrow^{k_1} c_1 A + c_2\lambda_1; \quad \tau_2 : A \rightarrow^{k_2} c_3\lambda_2;$	$[0,100]$ | 31

Table 3: **Left**: The 1-phase specification of the super Poisson process. **Centre**: The sketch. **Right**: Runtimes for different precisions.

Using precision $\delta = 10^{-3}$, we obtained the optimal solution $\{\xrightarrow{23} 2A, A \xrightarrow{94}\}$ (cost 16) in 4s. Notably, the synthesis without the cost constraints took 19s. Moreover, the ability to reason over the variance allows the solver to discard solution $\{\rightarrow A, A \rightarrow\}$ (implementation of a Poisson process [15]), which would have led to a variance equal to expectation. Table 3 (right) demonstrates the scalability of our approach with respect to the size of the continuous parameter space. Despite its non-trivial size (10 ODEs and discrete search space of size 288), we obtain remarkable performance, with runtimes in the order of seconds.

Phosphorelay Network. In the last case study we present a rate synthesis problem (i.e. all discrete parameters are instantiated) for a three-layer phosphorelay network [22]. In this network, each layer Li $(i = 1, 2, 3)$ can be found in phosphorylated form, Lip, and there is the ligand B, acting as an input for the network. The authors of [22] were interested in finding rates such that the time dynamics of $L3p$ shows ultra-sensitivity – a sigmoid shape of the time evolution of $L3p$ – which they obtained by manually varying the parameters until the right profile was discovered. We show that our approach can automatically find these parameters, thus overcoming such a tedious and time-consuming task.

We formalise the required behaviour using the 2-phase specification as shown in Table 4 (left). In particular, we consider a time interval $[0, 1]$ during which $L3p$ never decreases ($E^{(1)}[L3p] \geq 0$), and we require that an inflection point in the second derivative occurs in the transition between the two phases. At the final time we require that the population of $L3p$ is above 100, to rule out trivial solutions. For $N = 1000$ we consider the sketch listed in Table 4 (center), inspired by [22]. Figure 1 lists the rates synthesised for $\delta = 10^{-3}$ and illustrates the obtained sigmoid profile.

$k_1 = 15, k_2 = 53, k_3 = 90, k_4 = 3$

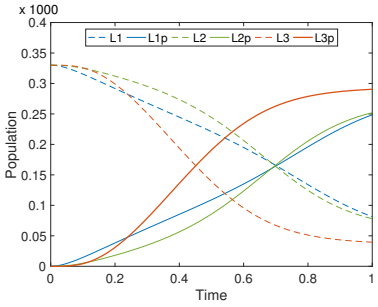


Fig. 1: The synthesised rates and the corresponding profile (without variance constraints).

We further consider a more complex variant of the problem, where we extend the specification to require that the variance of $L3p$ on its inflection point (the point where the variance is known to reach its maximum [22]) is limited by a threshold. This extension led to an encoding with 37 symbolic ODEs, compared to the 9 ODEs (7 species plus two ODEs for the derivatives of $L3p$) needed for the previous specification. Table 4 (right) shows the runtimes of the synthesis process for both variants of the model and different precisions δ. The results demonstrate that neither increasing the

$\mathsf{inv}_1 \equiv E^{(1)}[L3p] \geq 0 \wedge E^{(2)}[L3p] \geq 0$	$L1 + B \xrightarrow{k_1} B + L1p$	ODEs	δ	Time (s)
	$L2 + L1p \xrightarrow{k_2} L1 + L2p$	9	10^{-1}	53
$\mathsf{pre\text{-}post}_1 \equiv E^{(2)}[L3p]' = 0$	$L2p + L3 \xrightarrow{k_3} L2 + L3p$	9	10^{-3}	370
$\mathsf{inv}_2 \equiv E^{(1)}[L3p] \geq 0 \wedge E^{(2)}[L3p] \leq 0$	$L3p \xrightarrow{k_4} L3; \quad \emptyset \xrightarrow{1} B$	9	10^{-5}	719
$\mathsf{pre\text{-}post}_2 \equiv E[L3p]' > 100 \wedge T' = 1$	$k_1, \ldots, k_4 : (0, 100],$	37	10^{-1}	1052
	$Li_0 = 330, Lip_0 = B_0 = 0$	37	10^{-3}	11276
		37	10^{-5}	39047

Table 4: **Left**: The 2-phase specification of the sigmoid profile (no variance constraints). **Centre**: The sketch. **Right**: Runtimes for different precisions and the two variants (without and with covariances).

number of ODEs nor improving the precision leads to exponential slowdown of the synthesis process, indicating good scalability of our approach.

6 Conclusion

Automated synthesis of biochemical systems that exhibit prescribed behaviour is a landmark of synthetic and system biology. We presented a solution to this problem, introducing a novel method for SMT-based optimal synthesis of stochastic CRNs from rich temporal specifications and sketches (syntactic templates). By means of the LNA, we define the semantics of a sketch in terms of a set of parametric ODEs quadratic in the number of species, which allows us to reason about stochastic aspects not possible with the deterministic ODE-based semantics. Able to synthesize challenging systems with up to 37 ODEs and $\sim$ 10K admissible network topologies, our method shows unprecedented scalability and paves the way for design automation for provably-correct molecular devices.

In future work we will explore alternative notions of optimality and encodings, and develop a software tool based on parallel search strategies.

References

[1] R. Alur et al. "Syntax-guided synthesis". In: *Dependable Software Systems Engineering* 40 (2015), pp. 1–25.

[2] A. Andreychenko, L. Mikeev, D. Spieler, and V. Wolf. "Parameter identification for Markov models of biochemical reactions". In: *CAV'11*. Springer. 2011, pp. 83–98.

[3] D. Angluin, J. Aspnes, and D. Eisenstat. "Fast computation by population protocols with a leader". In: *Distributed Computing* 21.3 (2008), pp. 183–199.

[4] D. Angluin, J. Aspnes, D. Eisenstat, and E. Ruppert. "The computational power of population protocols". In: *Distributed Computing* 20.4 (2007), pp. 279–304.

[5] A. Arkin, J. Ross, and H. H. McAdams. "Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected Escherichia coli cells". In: *Genetics* 149.4 (1998), pp. 1633–1648.

[6] J. Barnat et al. “On parameter synthesis by parallel model checking”. In: *IEEE/ACM Trans. Comput. Biol. Bioinf.* 9.3 (2012), pp. 693–705.

[7] F. Benhamou, F. Goualard, L. Granvilliers, and J. F. Puget. “Revising Hull and Box Consistency”. In: *ICLP’99*. MIT Press, 1999, pp. 230–244.

[8] J. Bornholt, E. Torlak, D. Grossman, and L. Ceze. “Optimizing Synthesis with Metasketches”. In: *POPL’16*. ACM, 2016, pp. 775–788.

[9] L. Bortolussi, L. Cardelli, M. Kwiatkowska, and L. Laurenti. “Approximation of probabilistic reachability for chemical reaction networks using the linear noise approximation”. In: *QEST’16*. Springer. 2016, pp. 72–88.

[10] L. Bortolussi, D. Milios, and G. Sanguinetti. “Smoothed model checking for uncertain Continuous-Time Markov Chains”. In: *Information and Computation* 247 (2016), pp. 235 –253.

[11] R. C. Calinescu, M. Češka, S. Gerasimou, M. Kwiatkowska, and N. Paoletti. “Designing Robust Software Systems through Parametric Markov Chain Synthesis”. In: *IEEE International Conference on Software Architecture (ICSA 2017)*. IEEE. 2017.

[12] L. Cardelli. “Artificial biochemistry”. In: *Algorithmic Bioprocesses*. Springer, 2009, pp. 429–462.

[13] L. Cardelli. “Morphisms of reaction networks that couple structure to function”. In: *BMC systems biology* 8.1 (2014), p. 84.

[14] L. Cardelli. “Two-domain DNA strand displacement”. In: *Mathematical Structures in Computer Science* 23.02 (2013), pp. 247–271.

[15] L. Cardelli, M. Kwiatkowska, and L. Laurenti. “Programming discrete distributions with chemical reaction networks”. In: *DNA’16*. Springer. 2016, pp. 35–51.

[16] L. Cardelli, M. Kwiatkowska, and L. Laurenti. “Stochastic analysis of chemical reaction networks using linear noise approximation”. In: *Biosystems* 149 (2016), pp. 26–33.

[17] L. Cardelli, M. Kwiatkowska, and M. Whitby. “Chemical Reaction Network Designs for Asynchronous Logic Circuits”. In: *DNA’16*. Springer, 2016, pp. 67–81.

[18] L. Cardelli, M. Tribastone, M. Tschaikowski, and A. Vandin. “Comparing chemical reaction networks: A categorical and algorithmic perspective”. In: *LICS’16*. ACM. 2016, pp. 485–494.

[19] L. Cardelli, M. Tribastone, M. Tschaikowski, and A. Vandin. “Symbolic computation of differential equivalences”. In: *ACM SIGPLAN Notices*. Vol. 51. 1. ACM. 2016, pp. 137–150.

[20] M. Češka, F. Dannenberg, N. Paoletti, M. Kwiatkowska, and L. Brim. “Precise Parameter Synthesis for Stochastic Biochemical Systems”. In: *Acta Informatica* (2016), pp. 1–35.

[21] H.-L. Chen, D. Doty, and D. Soloveichik. “Rate-independent computation in continuous chemical reaction networks”. In: *ITCS’14*. ACM. 2014, pp. 313–326.

[22] A. Csikász-Nagy, L. Cardelli, and O. S. Soyer. "Response dynamics of phosphorelays suggest their potential utility in cell signalling". In: *J. R. Soc. Interface* 8.57 (2011), pp. 480–488.

[23] N. Dalchau, N. Murphy, R. Petersen, and B. Yordanov. "Synthesizing and tuning chemical reaction networks with specified behaviours". In: *DNA'15*. Springer, 2015, pp. 16–33.

[24] S.-J. Dunn, G. Martello, B. Yordanov, S. Emmott, and A. Smith. "Defining an essential transcription factor program for naive pluripotency". In: *Science* 344.6188 (2014), pp. 1156–1160.

[25] A. Eggers, M. Fränzle, and C. Herde. "SAT Modulo ODE: A Direct SAT Approach to Hybrid Systems". In: *ATVA'08*. Springer, 2008, pp. 171–185.

[26] A. Eggers, N. Ramdani, N. S. Nedialkov, and M. Fränzle. "Improving the SAT Modulo ODE Approach to Hybrid Systems Analysis by Combining Different Enclosure Methods". In: *Software and Systems Modeling* 14.1 (2015), pp. 121–148.

[27] A. Eldar and M. B. Elowitz. "Functional roles for noise in genetic circuits". In: *Nature* 467.7312 (2010), pp. 167–173.

[28] S. N. Ethier and T. G. Kurtz. *Markov processes: characterization and convergence*. Vol. 282. John Wiley & Sons, 2009.

[29] S. Gao, J. Avigad, and E. M. Clarke. "δ-complete decision procedures for satisfiability over the reals". In: *IJCAR'12*. Springer. 2012, pp. 286–300.

[30] S. Gao, S. Kong, and E. M. Clarke. "dReal: An SMT solver for nonlinear theories over the reals". In: *CADE'13*. Springer. 2013, pp. 208–214.

[31] S. Gerasimou, G. Tamburrelli, and R. Calinescu. "Search-Based Synthesis of Probabilistic Models for Quality-of-Service Software Engineering". In: *ASE'15*. 2015, pp. 319–330.

[32] M. Giacobbe, C. C. Guet, A. Gupta, T. A. Henzinger, T. Paixao, and T. Petrov. "Model Checking Gene Regulatory Networks". In: *TACAS'15*. Springer. 2015, pp. 469–483.

[33] M. Hardy. "Combinatorics of partial derivatives". In: *Electron. J. Combin* 13.1 (2006), p. 13.

[34] J. Heath, M. Kwiatkowska, G. Norman, D. Parker, and O. Tymchyshyn. "Probabilistic model checking of complex biological pathways". In: *Theoretical Computer Science* 391.3 (2008), pp. 239–257.

[35] S. Hoops et al. "COPASI – a complex pathway simulator". In: *Bioinformatics* 22.24 (2006), pp. 3067–3074.

[36] R. M. Karp and R. E. Miller. "Parallel program schemata". In: *Journal of Computer and system Sciences* 3.2 (1969), pp. 147–195.

[37] A. S. Koksal, Y. Pu, S. Srivastava, R. Bodik, J. Fisher, and N. Piterman. "Synthesis of Biological Models from Mutation Experiments". In: *POPL'13*. ACM, 2013, pp. 469–482.

[38] M. Kwiatkowska, G. Norman, and D. Parker. "PRISM 4.0: Verification of probabilistic real-time systems". In: *CAV'11*. Springer. 2011, pp. 585–591.

[39] M. Kwiatkowska and C. Thachuk. "Probabilistic model checking for biology". In: *Software Systems Safety* 36 (2014), p. 165.

[40] M. R. Lakin, D. Parker, L. Cardelli, M. Kwiatkowska, and A. Phillips. "Design and analysis of DNA strand displacement devices using probabilistic model checking". In: *J. R. Soc. Interface* 9.72 (2012), pp. 1470–1485.

[41] C. Madsen, F. Shmarov, and P. Zuliani. "BioPSy: An SMT-based Tool for Guaranteed Parameter Set Synthesis of Biological Models". In: *CMSB'15*. Springer, 2015, pp. 182–194.

[42] T. Murata. "Petri nets: Properties, analysis and applications". In: *Proceedings of the IEEE* 77.4 (1989), pp. 541–580.

[43] A. V. Nori, S. Ozair, S. K. Rajamani, and D. Vijaykeerthy. "Efficient Synthesis of Probabilistic Programs". In: *PLDI'14*. ACM, 2015, pp. 208–217.

[44] J. Ouaknine and J. Worrell. "Some recent results in metric temporal logic". In: *FORMATS'08*. Springer. 2008, pp. 1–13.

[45] N. Paoletti, B. Yordanov, Y. Hamadi, C. M. Wintersteiger, and H. Kugler. "Analyzing and synthesizing genomic logic functions". In: *CAV'14*. Springer. 2014, pp. 343–357.

[46] A. Solar-Lezama, C. G. Jones, and R. Bodik. "Sketching Concurrent Data Structures". In: *PLDI'08*. ACM, 2008, pp. 136–148.

[47] A. Solar-Lezama, R. Rabbah, R. Bodík, and K. Ebcioğlu. "Programming by Sketching for Bit-streaming Programs". In: *PLDI'05*. ACM, 2005, pp. 281–294.

[48] A. Solar-Lezama, L. Tancau, R. Bodik, S. Seshia, and V. Saraswat. "Combinatorial Sketching for Finite Programs". In: *ASPLOS'06*. ACM, 2006, pp. 404–415.

[49] D. Soloveichik, G. Seelig, and E. Winfree. "DNA as a universal substrate for chemical kinetics". In: *Proceedings of the National Academy of Sciences of the United States of America* 107.12 (2010), pp. 5393–5398.

[50] V. X. Tung, T. Van Khanh, and M. Ogawa. "raSAT: An SMT Solver for Polynomial Constraints". In: *IJCAR'16*. Springer, 2016, pp. 228–237.

[51] N. G. Van Kampen. *Stochastic processes in physics and chemistry*. Vol. 1. Elsevier, 1992.

[52] B. Yordanov, J. Kim, R. L. Petersen, A. Shudy, V. V. Kulkarni, and A. Phillips. "Computational design of nucleic acid feedback control circuits". In: *ACS synthetic biology* 3.8 (2014), pp. 600–616.

[53] C Zimmer and S Sahle. "Parameter Estimation for Stochastic Models of Biochemical Reactions". In: *J Comput Sci Syst Biol* 6 (2012), pp. 011–021.