Component-based Modeling and Reachability Analysis of Genetic Networks

Gregor Goessler

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Component-based Modeling and Reachability Analysis of Genetic Networks

Gregor Gössler

N° 6755

Décembre 2008

Thème COM

ISSN 0249-6399 ISRN INRIA/RR--6755--FR+ENG

Component-based Modeling and Reachability Analysis of Genetic Networks

Gregor Gössler

Thème COM — Systèmes communicants
Équipe-Projet Pop Art

Rapport de recherche n° 6755 — Décembre 2008 — 19 pages

Abstract: Genetic regulatory networks have been modeled as discrete transition systems by many approaches, benefiting from a large number of formal verification algorithms available for the analysis of discrete transition systems. However, most of these techniques do not scale up well. In this article, we explore a modular approach for the analysis of genetic regulatory networks. We present a framework for modeling genetic regulatory networks in a modular yet faithful manner based on the mathematically well-founded formalism of piecewise linear differential inclusions. We then propose a compositional algorithm to efficiently analyze reachability properties of the model. A case study on embryonic cell differentiation involving several hundred cells shows the potential of this approach.

Key-words: Genetic regulatory network, discrete abstraction, formal component model, reachability, compositional simulation

Centre de recherche INRIA Grenoble – Rhône-Alpes
655, avenue de l'Europe, 38334 Montbonnot Saint Ismier
Téléphone : +33 4 76 61 52 00 — Télécopie +33 4 76 61 52 52

Modélisation et analyse compositionnelle de réseaux génétiques

Résumé : De nombreuses approches utilisent les systèmes de transitions discrètes pour modéliser des réseaux génétiques, profitant d'un grand nombre d'algorithmes disponibles pour l'analyse de systèmes de transitions discrètes. Cependant, la plupart de ces techniques ne passe pas à l'échelle. Dans cet article, nous explorons une approche modulaire. Nous présentons un cadre pour la modélisation de réseaux génétiques de manière modulaire et fidèle, basé sur le formalisme mathématiquement bien fondé des inclusions différentielles linéaires par morceaux. Nous proposons ensuite un algorithme compositionnel permettant d'analyser efficacement des propriétés d'atteignabilité du modèle. Une étude de cas sur la différentiation de cellules embryonnaires portant sur plusieurs centaines de cellules montre le potentiel de cette approche.

Mots-clés : réseau de régulation génétique, abstraction discrète, modèle formel à composants, atteignabilité, simulation compositionnelle

1 Introduction

Genetic regulatory networks usually encompass a multitude of complex, interacting feedback loops. Being able to model and analyze their behavior is crucial for understanding the interactions between the proteins, and their functions. Genetic regulatory networks have been modeled as discrete transition systems by many approaches, benefiting from a large number of formal verification algorithms available for the analysis of discrete transition systems. However, most of these approaches face the problem of state space explosion, as even models of modest size (from a biological point of view) usually lead to large transition systems, due to a combinatorial blow-up of the number of states. Even if the modeling formalism allows for a compact representation of the state space, such as Petri nets, subsequent analysis algorithms have to cope with the full state space. In practice, monolithic approaches for the analysis of genetic regulatory networks do not scale.

In order to deal with the problem of state space explosion, different techniques have been developed in the formal verification community, such as partial order reduction, abstraction, and compositional approaches. In this article, we explore the use of compositionality for the analysis of genetic regulatory networks. Compositional analysis means that the behavior of a system consisting of interacting components is analyzed by separately examining the behavior of each component within an abstraction of its environment, rather than by monolithically analyzing the behavior of the overall system. It can therefore be more efficient than non-compositional analysis, and scale better.

A precondition for compositional algorithms to be applicable, is that the model be structured. Therefore, this paper makes two contributions: first, we present a modeling framework for genetic regulatory networks in which the different components of the system (in our case, protein concentrations) and the way they constrain each other, are modeled separately and modularly. Second, we propose a compositional algorithm allowing to efficiently analyze reachability properties of the model.

Cellular functions are often distributed over groups of components that interact within large networks. The components are organized in functional modules, forming a hierarchical architecture [1, 2]. Therefore, the approach of compositional analysis agrees with the modular structure of genetic regulatory networks, and may take advantage of it on different levels of granularity, for instance, between individual genes, sub-networks, or individual cells.

However, efficiency is not everything. The model should also faithfully represent the actual behavior of the modeled network. The approach we present is based on the mathematically well-founded formalism of qualitative simulation [3]. Given a system of piecewise linear differential equations, qualitative simulation abstracts the continuous dynamics into a set of qualitative states and discrete transitions between them. Each qualitative state corresponds either to a *regulatory domain*, in all points of which the system obeys the same linear differential equation, or to a *switching domain* on the frontier between two or more regulatory domains, where the values of one or more variables take some threshold.

This work introduces a new, modular discretization of piecewise linear differential equations. It extends and generalizes the preliminary, short version of [4]. New contributions include a generalization to switching domains of arbitrary order — where the order of a domain is defined as the number of variables taking threshold values —, and new correctness and completeness results. Representing the full state space including higher-order switching domains is important for two reasons: first, reachability may depend on the existence of paths crossing higher-order switching domains, and second, biologically relevant states like equilibria are often located on higher-order switching domains.

Related work

By now there is a large number of approaches to model and analyze genetic networks. An overview is given in the survey of [5]. The modeling approaches adopt different mathematical frameworks, which vary in expressiveness and the availability and efficiency of verification algorithms. Most of the algorithms "flatten" the model and work on the global state space, without computationally

taking advantage of the modularity of the problem. The approach of [6] compositionally models gene networks in a stochastic framework.

There has been a wide variety of modeling approaches based on differential equations, see e.g. [7] as an early example. However, simulation and verification of the continuous model are expensive, and many properties are not even decidable in this framework. Therefore, several ways have been investigated to discretize the continuous model defined by differential equations. The approach of [3] based on the approximation of nonlinear models by piecewise linear differential inclusions uses a discrete abstraction preserving the qualitative dynamics of networks. [8] uses abstractions over polynomials to define safe discrete approximations of hybrid automata. [9] and [10] use predicate abstraction to automatically compute backward reachable sets of piecewise affine hybrid automata, and find a conservative approximation of reachability for linear hybrid systems, respectively. [11] presents an algorithm for reachability analysis based on iterated refinement of discrete abstractions. [12] addresses the *bounded reachability* problem of hybrid automata.

In order to deal with complex networks, some approaches chose to directly model genetic networks in a discrete framework, such as the influential early work of [13] and more recently [14] based on logical equations, Petri nets [15, 16, 17, 18], rule-based formalisms like term rewriting systems [19, 20], or multi-valued decision diagrams [21]. Formal verification can then be carried out enumeratively (for instance, [22, 23, 24]) or symbolically, see for example [25].

Organization of the paper

In Section 2, we introduce the modeling framework. Starting from the qualitative framework of [3], we show how a genetic network can be modeled in a modular way in our framework, and we compare both frameworks. Section 3 presents a reachability algorithm taking advantage of the modularity of the model. Section 4 illustrates and benchmarks our results with a case study, and Section 5 concludes.

2 Component-based Modeling of Genetic Networks

This section briefly introduces the notions of piecewise linear system, and its qualitative simulation as defined in [3]. We then define a modularized approximation of qualitative simulation, and compare both frameworks.

2.1 Piecewise Linear Systems

The production of a protein in a cell is regulated by the current protein concentrations, which can activate or inhibit the production, for instance by binding to the gene and disabling transcription. At the same time, proteins are degraded. This behavior of a genetic network can be modeled by a system of differential equations of the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) - \mathbf{g}(\mathbf{x}, \mathbf{u})\mathbf{x} \tag{1}$$

where $\mathbf{x}$ is a vector of protein concentrations representing the current state, $\mathbf{u}$ is a vector of *input* concentrations, and the vector-valued function $\mathbf{f}$ and matrix-valued function $\mathbf{g}$ model the production rates, and degradation rates, respectively.

The approach of [3] considers an abstraction where the state space of each variable x_i is partitioned into a set of intervals $\mathcal{D}_i^r$ and a set of p_i *threshold values* $\mathcal{D}_i^s$. This induces a partition of the continuous state space into a discrete set of *domains*, in each of which Equation (1) is approximated with a system of linear differential equations.

Definition 1 (Domain) *Consider a Cartesian product* $\theta = \theta_1 \times ... \times \theta_n$ *with* $\theta_i = \{\theta_i^1, ..., \theta_i^{p_i}\}$ *an ordered set of* thresholds *such that* $0 < \theta_i^1 < ... < \theta_i^{p_i} < max_i$. *Let*

$$\mathcal{D}_i^r(\theta) = \{[0, \theta_i^1)\} \cup \{(\theta_i^j, \theta_i^{j+1}) \mid 1 \leq j < p_i\} \cup \{(\theta_i^{p_i}, max_i]\}$$

and $\mathcal{D}_i^s(\theta) = \big\{\{\theta_i^j\} \mid 1 \leq j \leq p_i\big\}$. *We omit the argument* θ *when it is clear from the context. Let* $\mathcal{D}_i = \mathcal{D}_i^r \cup \mathcal{D}_i^s$, *and* $\mathcal{D} = \mathcal{D}_1 \times \mathcal{D}_2 \times ... \times \mathcal{D}_n$ *be the set of* domains. *The domains in* $\mathcal{D}^r = \mathcal{D}_1^r \times \mathcal{D}_2^r \times ... \times \mathcal{D}_n^r$ *are called* regulatory *domains, the domains* $\mathcal{D}^s = \mathcal{D} \setminus \mathcal{D}^r$ *are called* switching domains.

The state space $[0, max_1] \times \cdots \times [0, max_n]$ is thus partitioned into the set of domains $\mathcal{D}$.

Definition 2 (Piecewise linear system) *A* piecewise linear system *is a tuple* $M = (X, \theta, \boldsymbol{\mu}, \boldsymbol{\nu})$ *where*

- $X = \{x_1, ..., x_n\}$ *a set of real-valued* state variables*;*
- $\theta = \theta_1 \times ... \times \theta_n$, *with* $\theta_i = \{\theta_i^1, ..., \theta_i^{p_i}\}$ *such that* $0 < \theta_i^1 < ... < \theta_i^{p_i} < max_i$, *associates with each dimension* i *an ordered set of* p_i thresholds*;*
- $\boldsymbol{\mu} : \mathcal{D}^r(\theta) \to \mathbb{R}^n_{\geq 0}$ *associates with each regulatory domain a vector of* production rates*;*
- $\boldsymbol{\nu} : \mathcal{D}^r(\theta) \to diag(\mathbb{R}^n_{>0})$ *associates with each regulatory domain a diagonal matrix of* degradation rates.

Within a regulatory domain $D \in \mathcal{D}^r$, the protein concentrations $\mathbf{x}$ evolve according to the ratio of production rate and degradation rate:

$$\dot{\mathbf{x}} = \boldsymbol{\mu}(D) - \boldsymbol{\nu}(D)\mathbf{x} \tag{2}$$

and thus converge monotonically towards the *focal point* $\boldsymbol{\phi}$, solution of $\mathbf{0} = \boldsymbol{\mu}(D) - \boldsymbol{\nu}(D)\mathbf{x}$.

Definition 3 ($\boldsymbol{\phi}$) *For any* $D \in \mathcal{D}^r$, *let* $\boldsymbol{\phi}(D)$ *denote the* focal point *of* D *such that*

$$\phi_i(D) = \mu_i(D)/\nu_i(D)$$

for any variable $x_i \in X$.

Hypothesis 1 *Throughout this paper we make the generic assumption that for any regulatory domain* D, $\exists D' \in \mathcal{D}^r \,.\, \boldsymbol{\phi}(D) \in D'$, *that is, all focal points lie within regulatory domains, as in [3].*

If $\boldsymbol{\phi}(D) \in D$ then the systems stays in D, otherwise it eventually leaves D and enters an adjacent switching domain. In switching domains, where $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ are not defined, the behavior of M is defined using differential inclusions as proposed by [26, 27].

Notations: For any $i \in \{1, ..., n\}$, let reg_i and $switch_i$ be predicates on $\mathcal{D}$ characterizing the regulatory intervals and thresholds of $\mathcal{D}_i$, respectively: $reg_i(D) \iff D_i \in \mathcal{D}_i^r$, and $switch_i(D) \iff D_i \in \mathcal{D}_i^s$ for any $D = (D_1, ..., D_n) \in \mathcal{D}$. For a domain D, let $switching(D) = \{x_i \mid switch_i(D)\}$ be the set of switching variables in D. The *order* of a domain D is the number of variables taking a threshold value in D, that is, $order(D) = |switching(D)|$. Let $succ_i : \mathcal{D}_i \to \mathcal{D}_i$ and $prec_i : \mathcal{D}_i \to \mathcal{D}_i$ be the successor and predecessor function on the ordered set of intervals $\mathcal{D}_i$ (in the sense that for any $D_1, D_2 \in \mathcal{D}_i$, $D_1 < D_2$ if $\forall x_1 \in D_1 \; \forall x_2 \in D_2 \,.\, x_1 < x_2$). We put $succ_i\big((\theta_i^{p_i}, max_i]\big) = prec_i\big([0, \theta_i^1)\big) = \perp$ ("undefined").

Definition 4 ($R(D)$) *For any domain* $D = (D_1, ..., D_n) \in \mathcal{D}$, *let* $R(D)$ *be the set of regulatory domains that have* D *in their boundary:*

$$R(D) = \big\{(D'_1, ..., D'_n) \mid \big(reg_i(D_i) \wedge D'_i = D_i\big) \vee \Big(switch_i(D_i) \wedge \big(D'_i = prec(D_i) \vee D'_i = succ(D_i)\big)\Big)\big\}$$

As a special case we have $R(D) = \{D\}$ for $D \in \mathcal{D}^r$.

Example 1 *Consider the example of two proteins a and b inhibiting each other's production [3]. The respective production rates of proteins a and b are defined by*

$$\mu_a = \begin{cases} 20 & \text{if } 0 \leq x_a < \theta_a^2 \wedge 0 \leq x_b < \theta_b^1 \\ 0 & \text{otherwise} \end{cases}$$
$$\mu_b = \begin{cases} 20 & \text{if } 0 \leq x_a < \theta_a^1 \wedge 0 \leq x_b < \theta_b^2 \\ 0 & \text{otherwise} \end{cases}$$

with $\theta_a^1 = \theta_b^1 = 4$ and $\theta_a^2 = \theta_b^2 = 8$. The degradation rate ν of both proteins is always 2. The example is thus modeled by the piecewise linear system $M = \big(\{x_a, x_b\}, \{\theta_a^1, \theta_a^2\} \times \{\theta_b^1, \theta_b^2\}, (\mu_a, \mu_b)^t, diag(\nu, \nu)\big)$.

2.2 Qualitative Model

In this section we shortly present the qualitative model of a given piecewise linear system, as defined in [3]. The continuous behaviors can be approximated by a discrete transition graph on the set of domains $\mathcal{D}$. This graph simulates the behavior of the underlying genetic network.

Definition 5 (eq) *Given $\theta = \theta_1 \times ... \times \theta_n$, we define predicates $eq_i^{\#}$ on $\mathcal{D}$, $i \in \{1, ..., n\}$, $\# \in \{<, \leq, \geq, >\}$ such that for any domain $D = (D_1, ..., D_n) \in \mathcal{D}$,*

$$eq_i^{\#}(D) \iff \exists D' \in R(D) \ \forall x \in D_i' \ . \ \phi_i(D') \# x \quad \text{for } \# \in \{<, >\}$$
$$eq_i^{=}(D) \iff \exists D' \in R(D) \ \exists x \in D_i' \ . \ \phi_i(D') = x$$

and $eq_i^{\leq} = (eq_i^{<} \vee eq_i^{=})$, $eq_i^{\geq} = (eq_i^{=} \vee eq_i^{>})$.

Intuitively, the predicates $eq_i^{\#}$ reflect the relative position of focal points of the adjacent regulatory domains. The predicates $eq_i^{<}(D)$ and $eq_i^{>}(D)$ specify when some adjacent regulatory domain has its focal point "left" of D_i and "right" of D_i, respectively.

Definition 6 *Given a piecewise linear system $M = (X, \theta, \boldsymbol{\mu}, \boldsymbol{\nu})$ of dimension n, the qualitative model $Q(M)$ is defined as the transition graph $Q(M) = (\mathcal{D}, \rightarrow)$ with transitions $\rightarrow \subseteq \mathcal{D} \times \mathcal{D}$ such that $\forall D, D' \in \mathcal{D}$, there is a transition $D \rightarrow D'$ if $D \neq D'$ and one of the following conditions holds*

1. $switching(D) \subset switching(D') \wedge not\_tr(D) \wedge order\_inc\_enabled(D, D')$
2. $switching(D') \subset switching(D) \wedge not\_tr(D') \wedge order\_dec\_enabled(D, D')$

where

$$not\_tr(D) = \bigwedge_{i=1,...,n} \big(reg_i(D) \quad \vee \min_{D' \in R(D)} \phi_i(D') < D_i < \max_{D' \in R(D)} \phi_i(D')\big)$$

$$order\_inc\_enabled(D, D') = \bigwedge_{i=1,...,n} \Big(D_i' = prec_i(D_i) \wedge eq_i^{<}(D) \quad \vee \quad D_i' = succ_i(D_i) \wedge eq_i^{>}(D) \quad \vee \quad D_i' = D_i\Big)$$

$$order\_dec\_enabled(D, D') = \bigwedge_{i=1,...,n} \Big(D_i' = prec_i(D_i) \wedge eq_i^{\leq}(D') \quad \vee \quad D_i' = succ_i(D_i) \wedge eq_i^{\geq}(D') \quad \vee \quad D_i' = D_i\Big)$$

Intuitively, condition $not\_tr(D)$ is satisfied if domain D is not *transient*, in the sense that it is instantaneously crossed by any trajectory of M reaching D [3]. The predicates $order\_inc\_enabled$ (D, D') (resp. $order\_dec\_enabled(D, D')$) specify that a transition from D to a higher (resp. lower) order domain D' is possible only if its direction is in every dimension consistent with the relative position of the focal points of the source (resp. target) domain. $Q(M)$ corresponds exactly to the qualitative transition graph defined in [3].

Example 2 *Continuing Example 1, Figure 1 shows the qualitative model $Q(M)$ of the piecewise linear system M, according to Definition 6.*

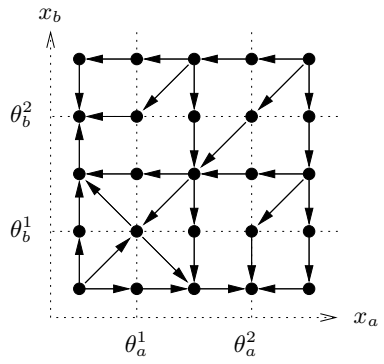

Figure 1: Transition graph of the qualitative model $Q(M)$ (solid graph). θ_a^i and θ_b^i are thresholds on the concentrations x_a and x_b, respectively.

2.3 Labeled Transition Systems and Constraints

In the following, we present a simplified version of the component framework adopted in [28]. For a set of states Q, let $\mathcal{P}(Q)$ be the set of predicates on Q. A predicate on Q thus represents a (sub)set of states.

Definition 7 (Labeled transition system) *A* labeled transition system *(LTS) is a tuple* $(Q, A, \rightarrow)$ *where Q is a set of* states*, A is a set of* actions*, and* $\rightarrow \subseteq Q \times A \times Q$ *is a* transition relation.

Definitions: We write $q \xrightarrow{a} q'$ for $(q, a, q') \in \rightarrow$. An LTS $(Q, A, \rightarrow)$ is *deterministic* if for any $q, q_1, q_2 \in Q$ and $a \in A$, $q \xrightarrow{a} q_1 \wedge q \xrightarrow{a} q_2 \implies q_1 = q_2$. In that case, the transition relation is a function, and we put $a(q) = q'$. For $B = (Q, A, \rightarrow)$ and $a \in A$, let $enabled(a)$ be the predicate characterizing the set $\{q \mid \exists q' \in Q \,.\, q \xrightarrow{a} q'\}$. We sometimes identify an LTS $(Q, A, \rightarrow)$ with the transition graph $(Q, \{q \rightarrow q' \mid \exists a \in A \,.\, q \xrightarrow{a} q'\})$ by "dropping the action names". We write $\rightarrow^*$ for the transitive and reflexive closure of $\rightarrow$. Given states q and q', q' is *reachable* from q if $q \rightarrow^* q'$.

Definition 8 (Predecessors) *Given an LTS $B = (Q, A, \rightarrow)$ and a predicate $P \in \mathcal{P}(Q)$, let the predicate $pre_a(P)$ characterize the* predecessors *of P by action a: $pre_a(P)(q) \iff \exists q' \,.\, q \xrightarrow{a} q' \wedge P(q')$. Let $pre(P) = \bigvee_{a \in A} pre_a(P)$, $pre^0(P) = P$, and $pre^{i+1}(P) = pre(pre^i(P))$, $i \geq 0$.*

The predicate $pre_a(P)$ (resp. $pre(P)$) characterizes the states from which execution of a (resp. execution of some action) leads to a state satisfying P.

We define two operations on LTS: composition and restriction. The composition of LTSs is an LTS again, and so is the restriction of an LTS.

Definition 9 (Composition) *The* composition *of two LTS $B_1 = (Q_1, A_1, \rightarrow_1)$ and $B_2 = (Q_2, A_2, \rightarrow_2)$ with $A_1 \cap A_2 = \emptyset$ is the LTS $B_1 \| B_2 = (Q_1 \times Q_2, A_1 \cup A_2, \rightarrow)$ where $(q_1, q_2) \xrightarrow{a} (q_1', q_2')$ if either $a \in A_1$, $q_1 \xrightarrow{a}_1 q_1'$, and $q_2' = q_2$, or $a \in A_2$, $q_2 \xrightarrow{a}_2 q_2'$, and $q_1' = q_1$.*

This is the standard asynchronous product. Restrictions allow to disable a subset of transitions of an LTS.

Definition 10 (Constraints) *Given an LTS $B = (Q, A, \rightarrow)$, a* state constraint *is a predicate $P \in \mathcal{P}(Q)$. An* action constraint *is a tuple of predicates $U = (U^a)_{a \in A}$ with $U^a \in \mathcal{P}(Q)$.*

Definition 11 (Restriction) *The* restriction *of an LTS $B = (Q, A, \rightarrow)$ with a state constraint P is the LTS $B/P = (Q, A, \rightarrow')$ where $\rightarrow' = \{(q, a, q') \in \rightarrow \mid P(q')\}$. The* restriction *of B with an action constraint $U = (U^a)_{a \in A}$ is the LTS $B/U = (Q, A, \rightarrow')$ where $\rightarrow' = \{(q, a, q') \in \rightarrow \mid U^a(q)\}$.*

Example 3 *Consider two LTSs $B_i = (\{low_i, high_i\}, \{inc_i, dec_i\}, \rightarrow_i)$ where $\rightarrow_i = \{(low_i, inc_i, high_i), (high_i, dec_i, low_i)\}$. Figure 2 (left) shows B_1 and B_2. The composition is $B_1 \| B_2 = (\{(low_1, low_2), (low_1, high_2), (high_1, low_2), (high_1, high_2)\}, \{inc_1, inc_2, dec_1, dec_2\}, \rightarrow)$ as shown in Figure 2 (middle).*

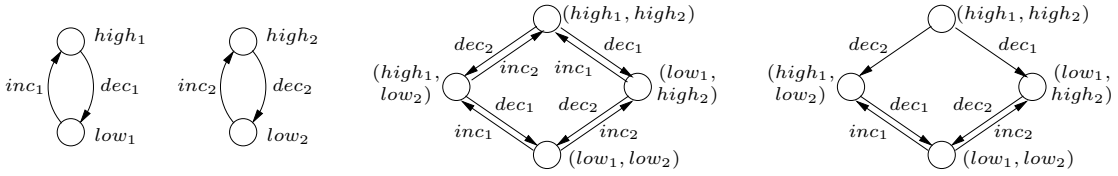

Figure 2: The labeled transition systems of B_1, B_2, $B_1 \| B_2$, and $(B_1 \| B_2)/U$.

Further suppose that we want to prevent B_1 from entering state $(high_1, high_2)$. This can be done in two ways: by restricting $B_1 \| B_2$ with state constraint $P = low_1 \vee low_2$, or with action constraint $U = (U^{inc_1}, U^{inc_2}, U^{dec_1}, U^{dec_2})$ where $U^{inc_1} = low_2$, $U^{inc_2} = low_1$, and $U^{dec_1} = U^{dec_2} =$ true. The restricted LTSs are $(B_1 \| B_2)/P = (B_1 \| B_2)/U = (\{(low_1, low_2), (low_1, high_2), (high_1, low_2), (high_1, high_2)\}, \{inc_1, inc_2, dec_1, dec_2\}, \rightarrow')$ as shown in Figure 2 (right).

Definition 12 ($incr$, $decr$) *Given a predicate P on $\mathcal{D}$ and $i \in \{1, ..., n\}$, we define the predicates $incr_i(P)$ and $decr_i(P)$ such that for any domain $D = (D_1, ..., D_i, ..., D_n) \in \mathcal{D}$, $incr_i(P)(D) = P\big(D_1, ..., succ_i(D_i), ..., D_n\big)$ if $succ_i(D_i) \neq \bot$, and $incr_i(P)(D) =$ false otherwise. Similarly, let $decr_i(P)(D) = P\big(D_1, ..., prec_i(D_i), ..., D_n\big)$ if $prec_i(D_i) \neq \bot$, and $decr_i(P)(D) =$ false otherwise.*

Intuitively, $incr_i(P)$ and $decr_i(P)$ denote the predicate P "shifted" by one domain along the i-th dimension, towards lower and higher values, respectively. For instance, consider predicate $P = (x_a = \theta_a^2)$ on the state space of Example 1. Then, $incr_a(P) = (\theta_a^1 < x_a < \theta_a^2)$ and $decr_b(P) = (x_a = \theta_a^2 \wedge \theta_b^1 \leq x_b \leq max_b)$.

2.4 Component Model of Genetic Networks

The qualitative model $Q(M)$ (Definition 6) defines one global transition graph whose complexity quickly becomes prohibitive with growing M. Therefore, we now define a component-based model $C(M)$ from a piecewise linear system M. In $C(M)$, the discretized behavior of each protein concentration is modeled separately. This leads to a structured model, whose modularity can be exploited by compositional verification algorithms.

Definition 13 ($C(M)$) *Given a piecewise linear system $M = (X, \theta, \boldsymbol{\mu}, \boldsymbol{\nu})$ with $|X| = n$, we define the LTS $C(M) = (\mathcal{D}, A, \rightsquigarrow) := (B_1 \| B_2 \| ... \| B_n)/U_M$, where the LTS B_i and the restriction U_M are defined as follows.*

- *$\forall i = 1, ..., n$. $B_i =$ $\mathsf{counter}(\mathcal{D}_i)$, where $\mathsf{counter}(\mathcal{D}_i)$ is a counter defined on $\mathcal{D}_i(\theta)$ by the LTS $\mathsf{counter}(\mathcal{D}_i) = \big(\mathcal{D}_i, \{inc_i, dec_i\}, \{(d, succ_i, d') \mid d, d' \in \mathcal{D}_i \wedge succ_i(d) = d'\} \cup \{(d, prec_i, d') \mid d, d' \in \mathcal{D}_i \wedge prec_i(d) = d'\}\big)$. The set of states is the set of intervals $\mathcal{D}_i$, and inc_i and dec_i are the actions of $\mathsf{counter}(\mathcal{D}_i)$.*

- *U_M is an action constraint such that $U_M(inc_i) = V_i^>$ and $U_M(dec_i) = V_i^<$ with*

$$V_i^< = reg_i \wedge not\_tr \wedge eq_i^< \vee \quad (3)$$
$$switch_i \wedge decr_i(eq_i^<) \quad (4)$$
$$V_i^> = reg_i \wedge not\_tr \wedge eq_i^> \vee \quad (5)$$
$$switch_i \wedge incr_i(eq_i^>) \quad (6)$$

Actions inc_i (dec_i) correspond to an increase (decrease) by one of the discretized concentration $level_i$ of protein i. The predicates $V_i^<$ and $V_i^>$ specify when a transition decrementing $level_i$ and incrementing $level_i$, respectively, is enabled. More precisely, lines (3) and (5) specify that there is a transition from a lower-order to a higher-order switching domain in the direction of the focal point of the source domain. Condition not_tr makes sure that there is a transition from a switching domain $D \in \mathcal{D}^s$ to a higher-order switching domain only if D is not transient for any dimension. Lines (4) and (6) give the conditions for transitions decreasing the order: they must be compatible with the relative position of the focal point of the destination domain. All LTSs $\mathsf{counter}(\mathcal{D}_i)$ are deterministic, therefore $C(M)$ is deterministic.

The above modeling framework enforces *separation of concerns* by making a clear distinction between the behaviors of the individual components, and constraints between the components.

Remark 1 *Since $\|$ is associative, Definition 13 leaves open how the system is actually partitioned into components (in the sense of sets of LTSs). The two extreme cases are that each B_i is considered as one component, or that $B_1\|B_2\|...\|B_n$ is considered as one single component. This choice will usually depend on the degree of interaction between the modeled proteins. Putting all proteins in one component amounts to a non-modular model leading to non-compositional analysis. Representing each protein with a separate component may lead to a large number of components, and loss of efficiency. Usually, a good choice is to have the components reflect the functional and spatial structure of the network by gathering closely interacting proteins in the same component, while modeling more loosely interacting proteins with separate components.*

Example 4 *Figure 3 shows the transition graphs of $\mathsf{counter}(\mathcal{D}_a)$, $\mathsf{counter}(\mathcal{D}_b)$, and $C(M)$ for the piecewise linear system M of Example 1.*

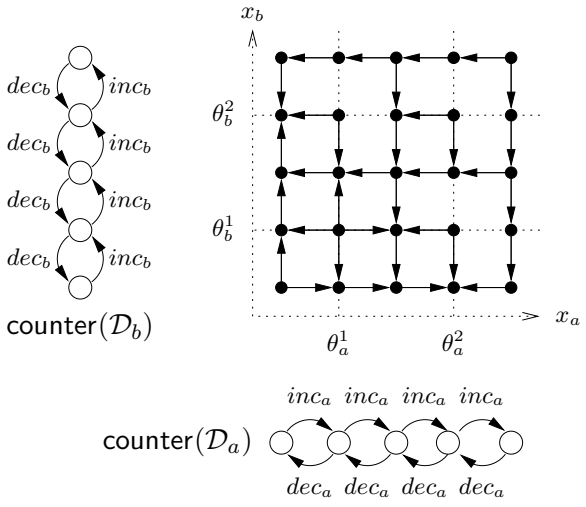

Figure 3: The transition graphs of $\mathsf{counter}(\mathcal{D}_a)$, $\mathsf{counter}(\mathcal{D}_b)$, and $C(M) = \big(\mathsf{counter}(\mathcal{D}_a)\|\mathsf{counter}(\mathcal{D}_b)\big)/U_M$.

2.5 Model Comparison

We show that the component model $C(M) = (\mathcal{D}, A, \rightsquigarrow)$ faithfully models $Q(M) = (\mathcal{D}, \rightarrow)$, up to the representation of "diagonal" transitions of $Q(M)$ synchronously changing more than one state variable. Order-decreasing transitions synchronously changing more than one state variable are serialized, whereas order-increasing transitions synchronously changing more than one state variable are not represented in $C(M)$. From a biological point of view, this approximation is justified by the neglectably small probability of two protein concentrations reaching threshold values exactly at the same instant.

Lemma 1 (Correctness of order-increasing transitions) *For any D, D' with $switching(D) \subset switching(D')$, $D \rightsquigarrow D' \implies D \rightarrow D'$.*

Proof 1 *Suppose that $D \rightsquigarrow D'$. Further suppose w.l.o.g. that $D'_k < D_k$ for some k, and $\forall i \neq k \;.\; D'_i = D_i$. Thus, $V_k^<(D)$ by Definition 13. Since by hypothesis $reg(D_k)$, it follows that $(not\_tr \wedge eq_k^<)(D)$, and $D \rightarrow D'$ follows by Definition 6.*

Lemma 2 (Correctness of order-decreasing transitions) *For any D, D' with $switching(D) \supset switching(D')$ and $not\_tr(D')$, if $\exists D^1, ..., D^{k-1}$ such that $D = D^0 \rightsquigarrow D^1 \rightsquigarrow ... \rightsquigarrow D^{k-1} \rightsquigarrow D^k = D'$ and $order(D^i) = order(D) - i$, then $D \rightarrow D'$.*

In particular, if D' is not transient then $D \rightsquigarrow D' \implies D \rightarrow D'$.

Proof 2 *Suppose w.l.o.g. that $\forall i \;.\; D'_i \leq D_i$, and let $I = \{i \mid D_i \neq D'_i\}$. It follows by Definition 13 that $\bigwedge_{i \in I} decr_i(eq_i^\leq)(D)$, and thus, $\bigwedge_{i \in I} eq_i^\leq(D')$. By definition 6 it follows that $D \rightarrow D'$.*

Theorem 1 (Correctness) *Consider a piecewise linear system $M = (X, \theta, \boldsymbol{\mu}, \boldsymbol{\nu})$. $C(M)$ under-approximates $Q(M)$ in the sense that for any $D, D' \in \mathcal{D}$ with $not\_tr(D')$, $D \rightsquigarrow D' \implies D \rightarrow D'$, and $D \rightsquigarrow^* D' \implies D \rightarrow^* D'$.*

Proof 3 *The first claim follows directly from Lemmas 1 and 2. The second claim follows by decomposing the path in $C(M)$ into monotonically order-increasing and order-decreasing sequences, using the fact that order-increasing transitions are enabled in $C(M)$ only from domains D satisfying $not\_tr(D)$.*

Proposition 1 (Completeness of order-increasing transitions) *If $D \rightarrow D'$ for some $D, D' \in \mathcal{D}$ with $order(D) + 1 = order(D')$, then $D \rightsquigarrow D'$.*

Intuitively, if $Q(M)$ can make some transition changing only the value of one variable, then $C(M)$ can make the same transition.

Proof 4 *Suppose that $D \rightarrow D'$. By hypothesis there is some dimension k such that $D_k \neq D'_k \wedge \forall j \neq k \;.\; D'_j = D_j$. Further suppose w.l.o.g. that $D'_k < D_k$. By Definition 6, $(not\_tr \wedge eq_k^<)(D)$ holds, and thus $V_k^<(D)$. The implication follows.*

When some diagonal order-decreasing transition is possible in $Q(M)$, then any interleaving is possible in $C(M)$. Formally:

Proposition 2 (Completeness of order-decreasing transitions) *If $D \rightarrow D'$ is an order-decreasing transition in $Q(M)$ with $switching(D) \supset switching(D')$, then for any sequence $D = D^0, D^1, ..., D^{k-1}, D^k = D'$ of (possibly transient) domains with $\forall i = 0, ..., k-1 \;.\; adj(D^i, D^{i+1}) \wedge switching(D^{i+1}) \subset switching(D^i)$ we have $D \rightsquigarrow D^1 \rightsquigarrow ... \rightsquigarrow D^{k-1} \rightsquigarrow D'$, where $adj(D, D') \iff \exists k \;.\; D'_k \in \{prec_k(D_k), succ_k(D_k)\} \wedge \bigwedge_{i \neq k} D'_i = D_i$.*

Proof 5 *Suppose w.l.o.g. that* $\forall i \,.\, D'_i \leq D_i$, *and let* $I = \{i \mid D_i \neq D'_i\}$. *It follows by Definition 6 that* $\bigwedge_{i \in I} eq_i^{\leq}(D')$. *Let* $(i_1, i_2, \ldots, i_k)$ *be some permutation of the indices in* I. *Consider the sequence of domains* $D^{i_0}, D^{i_1}, D^{i_2}, \ldots, D^{i_k} = D'$ *such that* $D^{i_0} = D$, *and for any* $j \in \{1, ..., k\}$, D^{i_j} *differs from* $D^{i_{j-1}}$ *only in the fact that variable* x_{i_j} *is switching in* $D^{i_{j-1}}$ *and regulatory in* D^{i_j}. *By definition of* $V^<$ *it follows that* $D \rightsquigarrow D_{i_1} \rightsquigarrow \cdots \rightsquigarrow D_{i_{k-1}} \rightsquigarrow D'$.

The example of Figures 1 and 3 illustrates the relation between $Q(M)$ and $C(M)$ discussed above.

2.6 Under-approximation

The component model $C(M)$ allows for compositional analysis of its behavior, by taking advantage of its structure. However, with growing size of the piecewise linear system M, even $C(M)$ may become too complex to analyze. Restricting the component model $C(M)$ to regulatory and first-order switching domains drastically reduces its complexity. The restricted model is often still precise enough to prove reachability properties in practice.

Definition 14 ($\mathcal{D}^{01}$) *Let* $\mathcal{D}^{01}$ *characterize the set of regular domains and first-order switching domains:*

$$\mathcal{D}^{01} = \bigvee_{i \in \{1,\ldots,n\}} \bigwedge_{j \in \{1,\ldots,n\} \setminus \{i\}} reg_j$$

Consider the model $C(M)/\mathcal{D}^{01}$, the restriction of $C(M) = (\mathcal{D}, A, \rightsquigarrow)$ to the regular and first-order switching domains, and let $\rightsquigarrow_{\mathcal{D}^{01}}$ denote its transition relation. This is clearly an under-approximation of both $Q(M) = (\mathcal{D}, \rightarrow)$ and $C(M)$:

Proposition 3 *For any two qualitative states* $D, D' \in \mathcal{D}$, $D \rightsquigarrow_{\mathcal{D}^{01}} D' \implies \big(D \rightarrow D' \wedge D \rightsquigarrow D'\big)$.

Example 5 *Consider the example of two proteins* a *and* b *modeled by the piecewise linear system* $M = \big(\{x_a, x_b\}, \{\theta_a\} \times \{\theta_b\}, (\mu_a, \mu_b)^t, diag(\nu, \nu)\big)$ *where*

$$\mu_a = \begin{cases} 2 & \text{if } 0 \leq x_a < \theta_a \wedge 0 \leq x_b < \theta_b \ \vee \ \theta_a < x_a \leq max_a \\ 0 & \text{otherwise} \end{cases}$$

$\mu_b = 2$, $\nu = 1$, *and* $\theta_a = \theta_b = 1$. *Figure 4 shows the transition graph of* $Q(M)$. *Obviously, all paths from domain* D_0 *to domain* D_8 *traverse the second-order switching domain* D_4. *By Propositions 1 and 2,* D_8 *is reachable from* D_0 *in* $C(M)$, *which is not the case in the under-approximation* $C(M)/\mathcal{D}^{01}$.

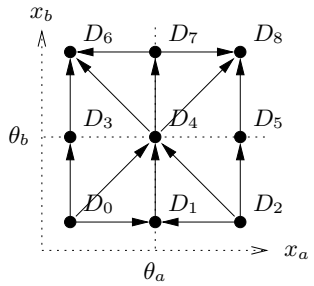

Figure 4: Reachability depends on higher-order switching domains being taken into account.

3 Compositional Reachability Analysis

The component model $C(M) = (\mathcal{D}, A, \rightsquigarrow)$ faithfully represents the qualitative model $Q(M) = (\mathcal{D}, \rightarrow)$, as discussed above, and can therefore serve as an input for a multitude of algorithms to analyze various properties. In the sequel, we will focus on the particular property of reachability. Based on the LTS $C(M)$, the compositional goal-directed simulation algorithm shown below allows to check for reachability of a goal domain, or set of domains, from an initial domain. The algorithm exhibits a path, if one is found, that solves the reachability problem.

In the sequel we consider a deterministic LTS $B = (Q, A, \rightarrow) = (B_1 \| \ldots \| B_N)/U$ with $B_i = (Q_i, A_i, \rightarrow_i)$, $i \in K = \{1, \ldots, N\}$, and U an action constraint. That is, we suppose the n proteins to be modeled with N ($1 \leq N \leq n$) components, according to Remark 1.

Let $path_k : Q_k \times \mathcal{P}(Q_k) \rightarrow 2^{A_k}$ be a function on the LTS B_k telling which action to take in order to bring component k from some current component state closer (in the number of transitions) to a state satisfying some predicate. This function can be computed locally: for any predicate $P \in \mathcal{P}(Q_k)$ and state $q \in Q_k$, let

$$path_k(q, P) = \{a \in A_k \mid \exists i \geq 0 \,.\, pre_a\big(pre_k^i(P)\big)(q) \ \wedge \ \forall j \in \{0, ..., i\} \,.\, \neg pre_k^j(P)(q)\}$$

That is, $path_k(q, P)$ contains an action a if and only if there exists some shortest path from q to P in the LTS B_k whose first transition is labeled with a.

Given a conjunction $c = c_1 \wedge ... \wedge c_N$ of predicates $c_i \in \mathcal{P}(Q_i)$, $i = 1, ..., N$, let $c[i] = c_i$ denote the *projection* of c on Q_i. For a set of actions A, let $enabling(A)$ be a list (disjunction) of predicates such that $\bigvee_{P \in enabling(A)} P = \bigvee_{a \in A} enabled(a)$, and each of the predicates in $enabling(A)$ is a conjunction $c_1 \wedge ... \wedge c_N$ of predicates on the component states. That is, $enabling(A)$ characterizes the states of B where some action in A is enabled, written in disjunctive form. Let $\oplus$ denote list concatenation. Given a non-empty list l, we write $l = e.l'$ where e is the first element, and l' the rest of the list. Given a list L of actions and a domain D, let $first\_enabled(L, D)$ be the first action a of L such that $\big(enabled(a)\big)(D)$, and $first\_enabled(L, D) = \bot$ ("undefined") if all actions are disabled.

Algorithm 1 Compositional goal-directed simulation. Initial call to construct a path σ from domain D_{init} to predicate P: $(D', \sigma, success) = \mathbf{move}(D_{init}, P, \langle\rangle)$.

$$\mathbf{move}\ (D, c.l, h) = \begin{cases} (D, \langle\rangle, true) & \text{if } c(D) \quad (1) \\ \big(a(D), \langle a \rangle, true\big) & \text{if } \neg c(D) \wedge a \neq \bot \quad (2) \\ (D'', \sigma \oplus \sigma', true) & \text{if } \neg\big(c(D) \vee a \neq \bot\big) \wedge goal \neq \emptyset \wedge goal\_enabled \wedge success \quad (3) \\ \mathbf{move}(D, l, h) & \text{if } \neg\big(c(D) \vee a \neq \bot \vee goal \neq \emptyset \wedge goal\_enabled \wedge success\big) \\ & \quad \wedge\ l \neq \langle\rangle \quad (4) \\ (D, \langle\rangle, false) & otherwise \quad (5) \end{cases}$$

where

$$a = first\_enabled(h, D)$$
$$goal = \bigcup_k path_k(D[k], c[k]) \setminus h$$
$$(D', \sigma, goal\_enabled) = \mathbf{move}\big(D, enabling(goal), h \oplus goal\big)$$
$$(D'', \sigma', success) = \mathbf{move}(D', c, h)$$

Algorithm 1 is constructive, that is, it establishes reachability from some initial domain D_{init} to a set of domains characterized by a predicate P by constructing a path from D_{init} to P. Function **move** works as follows. It takes as arguments the current domain D, the predicate P to be reached in the form of a list $d = \langle c_1, ..., c_m \rangle$ of conjunctions (such that $P = \bigvee_j c_j$), and an auxiliary list h of all actions requested to be executed, and returns a new domain, the part of the

path constructed so far, and a boolean indicating whether a path was found. The five cases are (1) if the current domain satisfies the predicate to be reached, then we are done. (2) Otherwise, execute the first action in h that is enabled. If there is none, compute the set *goal* of actions not considered so far that bring the components closer to the projection of the first element c of list d. (3) If *goal* is non-empty, recursively call **move** so as to reach some domain D' enabling some action in *goal*, then call **move** once more to continue moving towards c. (4) If reaching c fails, try the next conjunction of d. (5) If all above fails, then this call of **move** failed.

It can be shown that Algorithm 1 is guaranteed to terminate. It is not guaranteed to find a path even if one exists, though. Notice that it is straight-forward to make Algorithm 1 complete, by back-tracking when path search has failed for some choice of action a, and iterating over all remaining actions in h that are enabled in D. We have chosen, however, to iterate only over the conjunctions of the second argument of **move** (line (4)), as nested recursions would make the algorithm more difficult to read.

Theorem 2 (Correctness) *Algorithm 1 is correct, in the sense that if* $\mathbf{move}(D, P, \langle\rangle) = (D', \sigma, true)$*, then* $P(D')$*, and* σ *is a path from* D *to* D' *in* $(B_1 \| \ldots \| B_N)/U$.

Proof 6 *Given a list of actions* $\sigma = \langle a_0, ..., a_{n-1}\rangle$*, we write* $q_0 \xrightarrow{\sigma} q_n$ *if there are states* $q_1, ..., q_{n-1}$ *such that for any* $i \in \{0, ..., n-1\}$*,* $q_i \xrightarrow{a_i} q_{i+1}$*. We use structural induction to prove invariance of the implication* $\mathbf{move}(D, P, h) = (D', \sigma, true) \implies D \xrightarrow{\sigma} D' \wedge \big(P(D') \vee \exists a \in h\ \exists d \in \mathcal{D}\ \exists \sigma' \,.\, \sigma = \sigma' \oplus \langle a\rangle \wedge D \xrightarrow{\sigma'} d \xrightarrow{a} D'\big)$ *for the result of each call to* **move**. *It means that* σ *is a path from* D *to* D'*, and either the goal* P *has been reached, or the last transition of the path is labeled with an action in* h.

In the case of (1) and (5), the implication holds trivially. In case (2), $a = first\_enabled(h, D)$*, and the implication follows. For (3),* $D \xrightarrow{\sigma} D' \xrightarrow{\sigma'} D''$ *by hypothesis of induction, and the implication holds again. In case (4), the implication is satisfied by induction since it holds for the right-hand side.*

Correctness follows by the fact that for the initial call, $h = \langle\rangle$. ■

If for a predicate P characterizing a set of non-transient states, a path $D \rightsquigarrow^* D'$ is found on $C(M)$ (thus, $P(D')$ holds), then Theorem 1 ensures that a path $D \rightarrow^* D'$ exists in the qualitative model $Q(M)$.

Algorithm 1 is compositional in the sense that it independently computes local paths through the state spaces of the components (line $goal = \bigcup_k path_k(D[k], c[k]) \setminus h$). A global path is then constructed from the local paths and the constraints between the components: when an action a to be executed is blocked by a constraint involving other components, the function **move** is called recursively to move the blocking components into a domain where a is enabled.

According to Proposition 3, reachability in the under-approximation implies reachability in the qualitative model. Therefore, reachability can be checked efficiently on the restricted model $C(M)/\mathcal{D}^{01}$. If no path is found, the search can be refined by checking with the more precise model $C(M)$.

Example 6 (Example 4 continued.) *The functioning of Algorithm 1 is illustrated by the path construction from domain* $D_{init} = (\theta_a^1 < x_a < \theta_a^2, \theta_b^1 < x_b < \theta_b^2)$ *to domain* $D_{goal} = (x_a = \theta_a^2, 0 \leq x_b < \theta_b^1)$ *representing a stable equilibrium. The subsequent calls of* **move** *are*

$$
\begin{array}{l}
\mathbf{move}\ (D_{init}, \langle D_{goal}\rangle, \langle\rangle) \\
\quad a = \bot,\ goal = \{inc_a, dec_b\} \\
\quad \mathbf{move}\ (D_{init}, \langle \theta_a^1 < x_a < \theta_a^2, \ldots\rangle, \langle inc_a, dec_b\rangle) \\
\qquad = (D_1 = (\theta_a^1 < x_a < \theta_a^2,\ x_b = \theta_b^1), \langle dec_b\rangle, true) \qquad (2) \\
\quad \mathbf{move}\ (D_1, \langle D_{goal}\rangle, \langle\rangle) \\
\qquad a = \bot,\ goal = \{inc_a, dec_b\} \\
\qquad \mathbf{move}\ (D_1, \langle \theta_a^1 < x_a < \theta_a^2, \ldots\rangle, \langle inc_a, dec_b\rangle)
\end{array}
$$

$$= (D_2 = (\theta_a^1 < x_a < \theta_a^2,\ 0 \le x_b < \theta_b^1), \langle dec_b \rangle, \mathit{true}) \quad (2)$$
move $(D_2, \langle D_{goal} \rangle, \langle\rangle)$
$a = \bot,\ goal = \{inc_a\}$
move $(D_2, \langle x_a < \theta_a^2,\ 0 \le x_b < \theta_b^1, \dots \rangle, \langle inc_a \rangle)$
$= (D_{goal}, \langle inc_a \rangle, \mathit{true})$ (2)
move $(D_{goal}, \langle D_{goal} \rangle, \langle\rangle) = (D_{goal}, \langle\rangle, \mathit{true})$
$= (D_{goal}, \langle inc_a \rangle, \mathit{true})$ (3)
$= (D_{goal}, \langle dec_b, inc_a \rangle, \mathit{true})$ (3)
$= (D_{goal}, \langle dec_b, dec_b, inc_a \rangle, \mathit{true})$ (3)

Thus, D_{goal} is reached from D_{init} by decrementing $level_b$ twice and then incrementing $level_a$.

We have implemented Algorithm 1 in the compositional analysis tool PROMETHEUS [29]. A query language allows for interactive queries to the algorithm, and navigation through the state space.

4 Case Study: Delta-Notch Cell Differentiation

Cell differentiation by *Delta-Notch lateral inhibition* is an important step in the embryonic development of many species, as it causes initially uniform cells to assume different functions. Examples are the emergence of ciliated cells in *Xenopus* embryonic skin [30], or neurogenesis in *Drosophila*. *Delta-Notch* lateral inhibition is a well-studied phenomenon, see e.g. [30, 31, 11]. *Delta* is a transmembrane protein that binds and activates its receptor, the transmembrane protein *Notch*, in neighboring cells. High *Notch* levels in a cell inhibit its *Delta* production. Thus, a cell with a high *Delta* concentration prevents its immediate neighbors from adopting the same fate; those neighbors then take a different fate or remain undetermined. Figure 5 illustrates these interactions, which are the same as in [11].

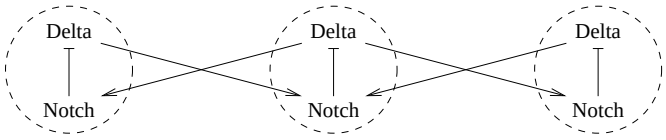

Figure 5: Interactions within and between neighbor cells.

For our case study, we consider networks consisting of 19 and 37 cells, and the networks of 49 and 245 cells shown in Figure 6. For each protein we partition the continuous state space into two intervals and one threshold value: $\mathcal{D}_D = \{[0, \theta_D), \{\theta_D\}, (\theta_D, max_D]\}$ and $\mathcal{D}_N = \{[0, \theta_N), \{\theta_N\}, (\theta_N, max_N]\}$. Cells with low *Delta* and high *Notch* levels ($0 \le Delta < \theta_D$, $\theta_N < Notch \le max_N$) are undifferentiated, whereas cells with high *Delta* and low *Notch* concentrations ($\theta_D < Delta \le max_D$, $0 \le Notch < \theta_N$) are differentiated, that is, set to adopt a different fate than undifferentiated cells. The actual values of the thresholds θ_D and θ_N are unknown; possible ranges have been derived in [31]. We suppose the focal point ϕ to satisfy

$$\begin{aligned}
0 \le \phi_{Delta_i} < \theta_D \text{ if } & \ Notch_i > \theta_N \\
\theta_D < \phi_{Delta_i} \le max_D \text{ if } & \ Notch_i < \theta_N \\
0 \le \phi_{Notch_i} < \theta_N \text{ if } & \ \max\{Delta_j \mid j \in neighbors(i)\} < \theta_D \\
\theta_N < \phi_{Notch_i} \le max_N \text{ if } & \ \max\{Delta_j \mid j \in neighbors(i)\} > \theta_D
\end{aligned}$$

In the following, we will use our implementation of Algorithm 1 to benchmark the impact of three parameters on the cost of reachability analysis: the size of the model M, the use of $C(M)$ versus the under-approximation $C(M)/\mathcal{D}^{01}$, and the degree of modularity of the model.

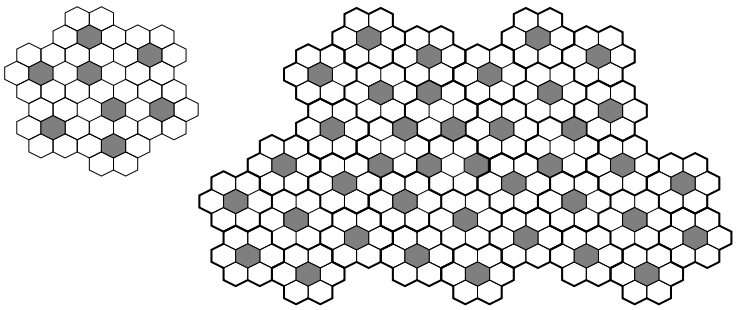

Figure 6: Examples of stable equilibrium states involving 49 cells (left) and 245 cells (right). Dark cells are differentiated.

Influence of size and precision In this section, we fix the granularity of the model and choose to represent each cell by one component.

For a piecewise linear model M of n cells, and thus a $2n$-dimensional global state space, both the qualitative model $Q(M)$ and the component model $C(M)$ encompass 9^n domains, whereas the under-approximation $C(M)/\mathcal{D}^{01}$ has a state space of 4^n regulatory domains and $2n \times 2^{2n-1}$ first-order switching domains.

We check reachability of a given stable equilibrium from the initial state where all cells are non differentiated, for the models of 19, 37, 49, and 245 cells. Table 1 shows the state spaces and execution times for each of the models. "Exact" and "under-approximation" mean that the model $C(M)$ and its restriction $C(M)/\mathcal{D}^{01}$ were used, respectively. The subsequent columns show the number of dimensions, the number of domains of the model, and the times for constructing the component model and a path to the final state using Algorithm 1. All measurements have been made on the same machine, a Pentium4 at 3 GHz with 512 MB of memory.

number of cells	dim.	state space	model	reachability
19 (under-approx.)	38	5.5×10^{12}	0.01 s	0.68 s
19 (exact)	38	1.4×10^{18}	18.4 s	10 min 38 s
37 (under-approx.)	74	7.2×10^{23}	0.04 s	2.7 s
49 (under-approx.)	98	1.6×10^{31}	0.08 s	4.0 s
245 (under-approx.)	490	7.9×10^{149}	2.4 s	46 min

Table 1: Performance on models of different size and precision.

In all cases, the results reported by PROMETHEUS are consistent with the actual, experimentally observed behavior [30]. For the case of 245 cells and the state shown in Figure 6, PROMETHEUS finds, in spite of the considerable size of the model, a path of length 152 reaching the state. The benchmarks suggest the use of the under-approximation as long as the model remains precise enough to establish the desired property.

Influence of modularity In order to evaluate the performance increase due to modularity, we now fix the size of M to 49 cells and use the under-approximation $C(M)/\mathcal{D}^{01}$. The only parameter that varies is the granularity of the components, where extreme cases are given by the model of 98 components each modeling one protein concentration, and the model consisting of one single component. According to Remark 1, the choice of the structure of the component model does not influence its behavior. All models are "flat", in the sense that components are not decomposed into sub-components. On each instance of the system we apply Algorithm 1 to find a path from the initial, undifferentiated state to the state of Figure 6 (left).

cells per component	0.5	1	3	7	10	49
reachability	10.7 s	7.5 s	8.4 s	35.5 s	(*)	(*)

Table 2: Benchmarks for different levels of modularity of Delta-Notch 49. (*): computation interrupted after 12 hours.

The measured performances are shown in Table 2. For this example, the optimal degree of granularity lies around one component per cell. It should be noted that the optimal partitioning of proteins into components depend on the system, and cannot be easily generalized. For a higher degree of modularity (1 component per protein), the algorithm performs somewhat slower, probably due to an overhead in coordination between closely interacting components. As the component size increases, complexity of the (non compositional) path construction within the components exponentially blows up.

This implies that the path construction in larger models is only feasible by exploiting the hierarchical structure of the components, and using compositionality on each level. For instance, our model of 245 cells discussed above consists of 5 identical top-level components, each of which is composed of 7 components, which are in turn composed of 7 bottom-level components each modeling one cell.

To complete the benchmarks, Table 3 shows the comparison with two other tools, GNA 5.5 and the model checking tool CADP [32]. Both tools are targeted at different functionalities, and do not compare directly to our approach. However, this comparison helps to confront the performance of compositional analysis with non-compositional techniques. The second column of Table 3 shows the time required by GNA to explore the state space reachable from the initial state in the model $Q(M)$. The last two columns measure the performance of reachability analysis using CADP on the models $C(M)/\mathcal{D}^{01}$ and $C(M)$ previously generated by PROMETHEUS. Notice that although CADP allows for compositional verification, we did not use this feature as it currently does not support action constraints.

	GNA ($Q(M)$)	CADP ($C(M)/\mathcal{D}^{01}$)	CADP ($C(M)$)
Delta-Notch 19	(*)	9.8 s	(*)
Delta-Notch 37	(*)	(*)	(*)

Table 3: Comparison with other tools. (*): computation interrupted after 24 hours.

5 Discussion

Regulatory networks of biological interest are usually composed of a large number of genes, proteins, and metabolites. There has been a wide variety of modeling approaches based on differential equations. However, simulation and verification of the continuous model are expensive, and many properties are not even decidable in this framework. The approach of [3] based on the approximation of nonlinear models by piecewise linear differential inclusions, uses a discrete abstraction preserving the qualitative dynamics of networks. This approach has been applied to the analysis of several models of networks controlling the stress response of bacteria. As [3] approximates the continuous behavior with a monolithic discrete transition system, it still suffers from two drawbacks: construction is not modular, and faces the problem of state space explosion, which limits subsequent analysis to models of limited size.

The work presented here addresses both issues. A discrete abstraction of the network dynamics, defined by a system of piecewise linear differential equations, is constructed component-wise. We have shown the equivalence of the component model and the discrete abstraction of [3] for biologically sound models. On the obtained model, reachability properties can be analyzed con-

structively by a compositional goal-directed simulation algorithm, allowing to deal with complex, high-dimensional systems. Our approach can help in coping with the fast growing complexity of models, as it is able to take advantage of the natural, spatial and functional modularity of genetic networks. The benchmarks clearly show its efficiency. Besides reachability analysis, goal-directed compositional simulation allows the user to navigate through huge state spaces of highly non-deterministic behaviors. We believe that a powerful simulation technique is crucial as an interface between biologists and other disciplines.

We intend to apply the technique to genetic networks involving a hierarchy of communicating functional modules, and to models of not yet fully understood networks. In order to further improve precision, we are studying the integration of the qualitative framework of [33] distinguishing *flow domains* in our framework.

This focus shift from the study of narrow aspects of cellular behavior towards the integrated view of systems biology, aiming at a system-level understanding of cells [34], is creating a strong need to jointly model and analyze different, interacting fragments of behavior, for instance, to study regulated metabolic networks. Component-based modeling is crucial to overcome the complexity of systems. However, the *heterogeneous* nature of the models is a major obstacle to be addressed. Currently, formal component frameworks for systems biology, capable of capturing, composing, and jointly analyzing these different fragments, in order to help understanding interacting aspects in systems biology, are badly missing. We intend to take advantage of the heterogeneous modeling capabilities of the component framework BIP [35, 36] in order to study the modeling and analysis of interacting biological functions. This work is a first step in the development of a methodology and tool platform for modeling, composition, and validation of complex heterogeneous models.

Acknowledgment. The author thanks Hidde de Jong for many fruitful discussions and comments on earlier versions of this work.

References

[1] Z. Oltvai and A. Barabási, "Life's complexity pyramid," *Science*, vol. 298, pp. 763–764, 2002.

[2] O. Resendis-Antonio, J. Freyre-González, R. Menchaca-Méndez, R. Gutiérrez-Ríos, A. Martínez-Antonio, C. Avila-Sánchez, and J. Collado-Vides, "Modular analysis of the transcriptional regulatory network of e. coli," *Trends in Genetics*, vol. 21, no. 1, pp. 16–20, 2005.

[3] H. de Jong, J.-L. Gouzé, C. Hernandez, M. Page, T. Sari, and J. Geiselmann, "Qualitative simulation of genetic regulatory networks using piecewise-linear models," *Bulletin of Mathematical Biology*, vol. 66, pp. 301–340, 2004.

[4] G. Gössler, "Compositional reachability analysis of genetic networks," in *CMSB'06*, ser. LNBI, C. Priami, Ed., vol. 4210. Springer, 2006, pp. 212–226.

[5] H. de Jong, "Modeling and simulation of genetic regulatory systems: A literature review," *Journal of Computational Biology*, vol. 9, no. 1, pp. 69–105, 2002.

[6] R. Blossey, L. Cardelli, and A. Phillips, "A compositional approach to the stochastic dynamics of gene networks," *Trans. on Comput. Syst. Biol.*, vol. 4, pp. 99–122, 2006.

[7] L. Glass and S. Kauffman, "The logical analysis of continuous, non-linear biochemical control networks," *Journal of Theoretical Biology*, vol. 39, no. 1, pp. 103–129, 1973.

[8] A. Tiwari and G. Khanna, "Series of abstractions for hybrid automata," in *proc. HSCC'02*, ser. LNCS, C. Tomlin and M. Greenstreet, Eds., vol. 2289. Springer-Verlag, 2002, pp. 465–478.

[9] R. Ghosh, A. Tiwari, and C. Tomlin, "Automated symbolic reachability analysis; with application to delta-notch signaling automata," in *proc. HSCC'03*, ser. LNCS, O. Maler and A. Pnueli, Eds., vol. 2623. Springer-Verlag, 2003, pp. 233–248.

[10] R. Alur, T. Dang, and F. Ivancić, "Reachability analysis of hybrid systems via predicate abstraction," *Trans. on Embedded Computing Systems*, 2004.

[11] R. Ghosh and C. Tomlin, "Symbolic reachable set computation of piecewise affine hybrid automata and its application to biological modeling: Delta-Notch protein signaling," *IEE Trans. on Systems Biology*, vol. 1, no. 1, pp. 170–183, 6 2004.

[12] C. Piazza, M. Antoniotti, V. Mysore, A. Policriti, F. Winkler, and B. Mishra, "Algorithmic algebraic model checking I: Challenges form systems biology," in *proc. CAV'05*, ser. LNCS, vol. 3576. Springer-Verlag, 2005, pp. 5–19.

[13] R. Thomas, "Boolean formalisation of genetic control circuits," *J. Theor. Biol.*, vol. 42, pp. 565–583, 1973.

[14] G. Bernot, J.-P. Comet, A. Richard, and J. Guespin, "Application of formal methods to biological regulatory networks: Extending Thomas' asynchronous logical approach with temporal logic," *Journal of Theoretical Biology*, vol. 229, no. 3, pp. 339–348, 2004.

[15] M. Heiner and I. Koch, "Petri net based model validation in systems biology," in *proc. ICATPN'04*, ser. LNCS, J. Cortadella and W. Reisig, Eds., vol. 3099. Springer-Verlag, 2004, pp. 216–237.

[16] C. Chaouiya, E. Remy, P. Ruet, and D. Thieffry, "Qualitative modelling of genetic networks: From logical regulatory graphs to standard petri nets," in *proc. ICATPN'04*, ser. LNCS, J. Cortadella and W. Reisig, Eds., vol. 3099. Springer-Verlag, 2004, pp. 137–156.

[17] E. Simão, E. Remy, D. Thieffry, and C. Chaouiya, "Qualitative modelling of regulated metabolic pathways: application to the tryptophan biosynthesis in e.coli," *Bioinformatics*, vol. 21, no. Suppl. 2, pp. ii190–ii196, 2005.

[18] J.-P. Comet, H. Klaudel, and S. Liauzu, "Modeling multi-valued genetic regulatory networks using high-level petri nets," in *proc. ICATPN'05*, ser. LNCS, G. Ciardo and P. Darondeau, Eds., vol. 3536. Springer-Verlag, 2005, pp. 208–227.

[19] S. Eker, M. Knapp, K. Laderoute, P. Lincoln, and C. Talcott, "Pathway logic: Executable models of biological networks," *ENTCS*, vol. 71, 2002.

[20] F. Fages, S. Soliman, and N. Chabrier-Rivier, "Modelling and querying interaction networks in the biochemical abstract machine BIOCHAM," *J. of Biological Physics and Chemistry*, vol. 4, no. 2, pp. 64–72, 2004.

[21] A. Naldi, D. Thieffry, and C. Chaouiya, "Decision diagrams for the representation and analysis of logical models of genetic networks," in *proc. CMSB 2006*, ser. LNBI, M. Calder and S. Gilmore, Eds., vol. 4695. Springer, 2007, pp. 233–247.

[22] H. de Jong, J. Geiselmann, C. Hernandez, and M. Page, "Genetic Network Analyzer: Qualitative simulation of genetic regulatory networks," *Bioinformatics*, vol. 19, no. 3, pp. 336–344, 2003.

[23] G. Batt, D. Bergamini, H. de Jong, H. Garavel, and R. Mateescu, "Model checking genetic regulatory networks using GNA and CADP," in *proc. SPIN'04*, ser. LNCS, vol. 2989. Springer-Verlag, 2004, pp. 158–163.

[24] A. Larrinaga, A. Naldi, L. Sánchez, D. Thieffry, and C. Chaouiya, "GINsim: a software suite for the qualitative modelling, simulation and analysis of regulatory networks," *Biosystems*, 2005.

[25] N. Chabrier and F. Fages, "Symbolic model checking of biochemical networks," in *proc. CMSB'03*, 2003.

[26] A. Filippov, "Differential equations with discontinuous righthand side," *Mathematics and its Applications*, vol. 18, 1988.

[27] J.-L. Gouzé and T. Sari, "A class of piecewise linear differential equations arising in biological models," *Dynamical Systems*, vol. 17, no. 4, pp. 299–316, 2003.

[28] G. Gössler and J. Sifakis, "Component-based construction of deadlock-free systems (extended abstract)," in *proc. FSTTCS'03*, ser. LNCS, vol. 2914. Springer-Verlag, 2003.

[29] G. Gössler, "Componend-based design of heterogeneous reactive systems in PROMETHEUS," INRIA, Research Report 6057, 2006.

[30] G. Marnellos, G. Deblandre, E. Mjolsness, and G. Kintner, "Delta-Notch lateral inhibitory patterning in the emergence of ciliated cells in *Xenopus*: Experimental observations and a gene network model," in *proc. PSB'00*, vol. 5. World Scientific Publishing, 2000, pp. 326–337.

[31] R. Ghosh and C. Tomlin, "Lateral inhibition through delta-notch signaling: A piecewise affine hybrid model," in *proc. HSCC'01*, ser. LNCS, M. D. Benedetto and A. Sangiovanni-Vincentelli, Eds., vol. 2034. Springer-Verlag, 2001, pp. 232–246.

[32] J.-C. Fernandez, H. Garavel, A. Kerbrat, R. Mateescu, L. Mounier, and M. Sighireanu, "CADP: A protocol validation and verification toolbox," in *Proc. CAV '96*, ser. LNCS, R. Alur and T. Henzinger, Eds., vol. 1102. Springer-Verlag, 1996, pp. 437–440.

[33] G. Batt, H. de Jong, J. Geiselmann, and M. Page, "Symbolic reachability analysis of genetic regulatory networks using qualitative abstractions," *Automatica*, 2007.

[34] H. Kitano, "Looking beyond the details: a rise in system-oriented approaches in genetics an molecular biology," *Current Genetics*, vol. 41, pp. 1–10, 2002.

[35] G. Gössler and J. Sifakis, "Composition for component-based modeling," *Science of Computer Programming*, vol. 55, no. 1-3, pp. 161–183, 2005.

[36] S. Bliudze and J. Sifakis, "The algebra of connectors — structuring interaction in BIP," in *Proc. EMSOFT'07*, 2007, pp. 11–20.

Centre de recherche INRIA Grenoble – Rhône-Alpes
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399