

Qualitative Simulation of Genetic Regulatory Networks Using Piecewise-Linear Models

Hidde de Jong — Jean-Luc Gouzé — Céline Hernandez — Michel Page — Tewfik Sari —
Johannes Geiselman

N° 4407

Janvier 2002

THÈMES 4 et 3



*Rapport
de recherche*

Qualitative Simulation of Genetic Regulatory Networks Using Piecewise-Linear Models

Hidde de Jong , Jean-Luc Gouzé , Céline Hernandez , Michel Page , Tewfik Sari* , Johannes Geiselman†

Thèmes 4 et 3 — Simulation et optimisation
de systèmes complexes — Interaction homme-machine,
images, données, connaissances
Projets COMORE, HELIX

Rapport de recherche n° 4407 — Janvier 2002 — 31 pages

Abstract: In order to cope with the large amounts of data that have become available in genomics, mathematical tools for the analysis of networks of interactions between genes, proteins, and other molecules are indispensable. We present a method for the qualitative simulation of genetic regulatory networks, based on a class of piecewise-linear (PL) differential equations that has been well-studied in mathematical biology. The simulation method is well-adapted to state-of-the-art measurement techniques in genomics which often provide qualitative and coarse-grained descriptions of genetic regulatory networks. The method is able to deal with nontrivial mathematical problems induced by the discontinuous right-hand sides of the differential equations. Furthermore, it guarantees that the simulation covers all possible solutions of quantitative PL models corresponding to the qualitative PL model used by the method. The qualitative simulation method has been implemented in Java.

Key-words: genetic regulatory networks, mathematical modeling, piecewise-linear differential equations, discontinuous differential equations, qualitative simulation, mathematical biology, bioinformatics

* Laboratoire de Mathématiques, Université de Mulhouse.

† Laboratoire Plasticité et Expression des Génomes Microbiens, Université Joseph Fourier, Grenoble.

Simulation Qualitative de Réseaux de Régulation Génique en Utilisant des Modèles Linéaires par Morceaux

Résumé : Afin de traiter les grandes quantités de données génomiques disponibles aujourd'hui, des outils mathématiques pour l'analyse de réseaux d'interactions entre gènes, protéines, et d'autres molécules sont indispensables. Nous présentons une méthode pour la simulation qualitative de réseaux de régulation génique, basée sur une classe d'équations différentielles linéaires par morceaux (LM) qui a été bien étudiée en biologie mathématique. La méthode de simulation est bien adaptée à l'état de l'art des techniques expérimentales, qui donnent souvent des descriptions qualitatives et agrégées des réseaux de régulation génique. La méthode est capable de traiter des problèmes mathématiques non triviaux induits par des discontinuités dans le second membre des équations différentielles. En outre, elle garantit que la simulation couvre toutes les solutions possibles des modèles LM quantitatifs correspondants au modèle LM qualitatif utilisé par la méthode. La méthode de simulation a été implémentée en Java.

Mots-clés : réseaux de régulation génique, modélisation mathématique, équations différentielles linéaires par morceaux, équations différentielles discontinues, simulation qualitative, biologie mathématique, bioinformatique

1 Introduction

Recent progress in genomics has provided us with experimental tools that hold great promises for unraveling the networks of regulatory interactions between genes, proteins, and small molecules which underlie the functioning of living organisms. On the one hand, these techniques allow protein-DNA and protein-protein interactions to be identified, thus providing insight into the structure of genetic regulatory systems (e.g., [47, 53]). On the other hand, they allow the evolution of the state of the system to be characterized, by large-scale measurement of the level of gene expression and protein activity across time (e.g., [40, 68]).

In order to cope with the large amounts of data that have thus become available, formal methods for the representation and analysis of genetic regulatory networks are indispensable. Mathematical models allow networks of interactions to be described in a precise and unambiguous manner, while a large variety of analysis and simulation techniques exists to systematically derive behavioral predictions from the models. The application of formal methods, especially when supported by computer tools, may lead to a comprehension of the structure and functioning of large and complex networks of interactions that cannot be obtained through more intuitive approaches alone [14, 41].

The use of formal methods to study regulatory networks is currently subject to two major constraints [7]. First of all, the biochemical reaction mechanisms underlying the interactions are usually not or incompletely known. This prevents the formulation of detailed kinetic models, such as those developed for the genetic switch controlling phage λ growth [42] or the feedback mechanisms regulating tryptophan synthesis in *E. coli* [55]. A second constraint arises from the general absence of quantitative information on kinetic parameters and molecular concentrations. As a consequence, traditional methods for numerical analysis are difficult to apply.

Few of the modeling and simulation methods that have been developed thus far are capable of handling the above constraints. A notable exception is formed by approaches based on a class of *piecewise-linear* (PL) differential equation models originally proposed by Glass and Kauffman [23]. The state variables in the PL models correspond to the concentrations of proteins encoded by genes in the network, while the differential equations represent the interactions arising from the regulatory influence of some proteins on the synthesis and degradation of others. The regulatory interactions are modeled by means of step functions, which gives rise to the piecewise-linear structure of the differential equations. The use of step functions is motivated by the nonlinear, switch-like character of many of the interactions in gene expression and proteolysis [52, 67].

The PL models provide a coarse-grained description of genetic regulatory networks, well-adapted to state-of-the-art measurement techniques in genomics. Furthermore, the models have mathematical properties that favour qualitative analysis of the steady-state and transient behavior of regulatory systems [11, 12, 19, 20, 24, 25, 38, 44, 45, 48, 56, 57]. On a formal level, the PL models are related to a class of asynchronous logical models proposed by Thomas and colleagues [62, 63]. PL models and their logical relatives have been used for the study of a number of prokaryotic and eukaryotic regulatory networks [1, 10, 18, 43, 46, 51, 54, 61]. In addition, they have been used for modeling food webs [49], neural networks [39], and biological computers [5].

The discontinuous nature of the step functions in PL models brings about some nontrivial mathematical problems. Existing approaches either avoid these problems by restricting the analysis to a subclass of regulatory networks, or adopt solutions that have undesirable consequences for the predictiveness of the method. Recently, it has been shown that an approach capable of dealing with differential equations with discontinuous right-hand sides, widely used in control theory, allows the above-mentioned problems to be resolved in a mathematically proper and practically useful manner [27]. This approach, originally proposed by Filippov [15], is based on the generalization of the differential equations to differential inclusions.

In this paper we present a method for the *qualitative simulation* of genetic regulatory networks described by the generalized PL models. The method is obtained by formulating the analysis of PL

models in terms of concepts developed for the qualitative simulation of dynamical systems [10, 35, 36]. In particular, we supplement the PL differential equations with qualitative constraints on the parameter values in the form of algebraic inequalities that can be inferred from biological data. The method exploits the constraints to determine all qualitative states that are reachable from some initial qualitative state through one or more transitions. Each qualitative state corresponds to a region of the phase space where the system behaves in a qualitatively distinct way. The graph of qualitative states and transitions predicted by the method is guaranteed to cover all solutions of the differential equations consistent with the parameter constraints.

The simulation method has been implemented in a publicly-available computer tool, called *Genetic Network Analyzer (GNA)* [9]. In a companion paper we use the method and the tool to analyze a genetic regulatory network of biological interest, consisting of the genes and interactions that regulate the initiation of sporulation in *B. subtilis* [8]. The application shows that the simulation method can help to gain insight into the qualitative dynamics of complex regulatory networks involving dozens of genes.

In the next two sections of the paper, the mathematical framework underlying the simulation method will be reviewed. Sections 4 and 5 introduce the notions of qualitative PL model and qualitative state and behavior, respectively. Using a qualitative PL model of the regulatory network, supplemented by qualitative initial conditions, the simulation algorithm described in section 6 generates a graph of qualitative states and transitions between qualitative states. The soundness and completeness of the algorithm are investigated in section 7. In the final section of the paper, the method is discussed in the context of related work.

2 Piecewise-linear models of genetic regulatory networks

Figure 1 shows an example of a simple genetic regulatory network. The genes a and b , transcribed from separate promoters, encode the proteins A and B, each of which controls the expression of both genes.¹ Proteins A and B repress gene a and b at different concentrations. Repression of the genes is achieved by binding of the proteins to regulatory sites overlapping with the promoters. The pattern of interactions gives rise to one positive and two negative feedback loops.

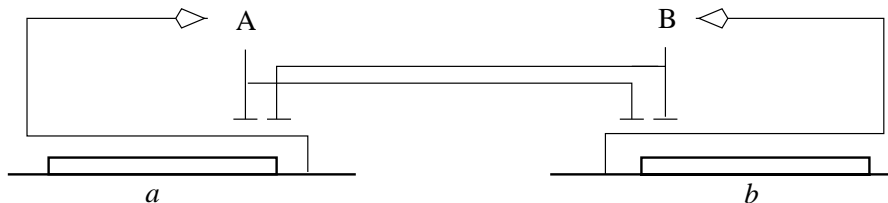


Figure 1: Example of a genetic regulatory network of two genes (a and b) coding for a regulatory protein (A and B). The notation follows, in a somewhat simplified form, the graphical conventions proposed by Kohn [34].

The dynamics of genetic regulatory networks can be modeled by a class of differential equations proposed by Mestl *et al.* [44], extending previous work by Glass and Kauffman [23] (see also the work of Snoussi and Thomas [56, 62] and Ratner and Tchouarev [59]). The equations have the general form

$$\dot{x}_i = f_i(\mathbf{x}) - g_i(\mathbf{x}) x_i, \quad x_i \geq 0, \quad 1 \leq i \leq n, \quad (1)$$

where $\mathbf{x} = (x_1, \dots, x_n)'$ is a vector of cellular protein concentrations. The *state equations* (1) define the rate of change of each concentration x_i as the difference of the rate of synthesis $f_i(\mathbf{x})$ and the rate

¹As a notational convention, names of genes are printed in italic and names of proteins start with a capital.

of degradation $g_i(\mathbf{x}) x_i$ of the protein. In vector notation, the system of differential equations (1) is written as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) - \mathbf{g}(\mathbf{x}) \mathbf{x}, \quad (2)$$

with $\mathbf{f} = (f_1, \dots, f_n)'$ and $\mathbf{g} = \text{diag}(g_1, \dots, g_n)$.

The function $f_i : \mathbb{R}_{\geq 0}^n \rightarrow \mathbb{R}_{\geq 0}$ is defined as

$$f_i(\mathbf{x}) = \sum_{l \in L} \kappa_{il} b_{il}(\mathbf{x}), \quad (3)$$

where $\kappa_{il} > 0$ is a (bounded) rate parameter, $b_{il} : \mathbb{R}_{\geq 0}^n \rightarrow \{0, 1\}$ a *regulation function* defined in terms of step functions, and L a possibly empty set of indices of regulation functions. A regulation function b_{il} is the arithmetic equivalent of a Boolean function expressing the logic of gene regulation [50, 56].

In the simplest case, $f_i(\mathbf{x}) = \kappa_i s^+(x_j, \theta_j)$, where the *step function* $s^+ : \mathbb{R}^2 \rightarrow \{0, 1\}$ is defined as follows (see also figure 7):

$$s^+(x_j, \theta_j) = \begin{cases} 1, & x_j > \theta_j, \\ 0, & x_j < \theta_j. \end{cases} \quad (4)$$

As a consequence, gene i is not expressed below a threshold concentration $\theta_j > 0$, whereas above this threshold it is expressed at a rate κ_i . If protein J is a negative regulator of gene i , we have $f_i(\mathbf{x}) = \kappa_i s^-(x_j, \theta_j)$, with $s^-(x_j, \theta_j) = 1 - s^+(x_j, \theta_j)$. More complex regulation functions can express the combined effect of several regulatory proteins. The use of step functions in gene regulation models has been motivated by the observation that the activation of a gene, as a function of the concentration of a regulatory protein, often follows a steep sigmoidal curve [52, 67].

The function g_i allows the regulation of protein degradation to be modeled. It is defined analogously to (3), except that we demand that $g_i(\mathbf{x})$ is strictly positive. In addition, in order to formally distinguish degradation rates from synthesis rates, we will denote the former by γ instead of κ . Notice that with the above definitions of f_i and g_i , the state equations (1) and (2) are *piecewise-linear (PL)*.

The PL models can be extended to take into account *input variables* $\mathbf{u} = (u_1, \dots, u_m)'$, representing the concentration of proteins and small molecules whose synthesis and degradation are regulated outside the system. This leads to models of the form:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) - \mathbf{g}(\mathbf{x}, \mathbf{u}) \mathbf{x}. \quad (5)$$

In what follows, we will assume that the input variables are constant, *i.e.*, $\dot{\mathbf{u}} = \mathbf{0}$. As a consequence, (5) can be reduced to (2) without loss of generality, by prior evaluation of the step function expressions in which input variables occur.

In figure 2 the state equations for the example network are shown. Gene a is expressed at a rate κ_a , if the concentration of protein A is below its threshold θ_a^2 and the concentration of protein B below its threshold θ_b^1 , that is, if $s^-(x_a, \theta_a^2) s^-(x_b, \theta_b^1) = 1$. Analogously, gene b is expressed at a rate κ_b , if the concentration of protein A is below the threshold θ_a^1 and the concentration of protein B below the threshold θ_b^2 . Degradation of the proteins A and B is assumed to be spontaneous, which gives rise to regulation functions having the value 1, independent of the concentrations of the proteins.

The dynamical properties of PL models of the form (2) can be analyzed in the n -dimensional phase space box $\Omega = \Omega_1 \times \dots \times \Omega_n$, where every Ω_i , $1 \leq i \leq n$, is defined as

$$\Omega_i = \{x_i \in \mathbb{R}_{\geq 0} \mid 0 \leq x_i \leq \text{max}_i\}. \quad (6)$$

max_i is a parameter denoting a maximum concentration for the protein. It can be shown, by generalizing the argument in [21], that if we choose $\text{max}_i = \max_{\mathbf{x} \geq \mathbf{0}} f_i(\mathbf{x})/g_i(\mathbf{x})$, all trajectories starting

State equation for gene a :

$$\dot{x}_a = \kappa_a s^-(x_a, \theta_a^2) s^-(x_b, \theta_b^1) - \gamma_a x_a$$

State equation for gene b :

$$\dot{x}_b = \kappa_b s^-(x_a, \theta_a^1) s^-(x_b, \theta_b^2) - \gamma_b x_b$$

Figure 2: State equations for the network of figure 1.

inside Ω will remain in it, while trajectories starting outside will enter the phase space box at some point.

In general, a protein encoded by a gene will be involved in different interactions at different threshold concentrations, which after ordering are denoted by $\theta_i^1, \dots, \theta_i^{p_i}$. The $n - 1$ -dimensional hyperplanes $x_i = \theta_i^{k_i}$, $1 \leq k_i \leq p_i$, divide Ω into hyperrectangular regions that are called *regulatory domains*. Within each region, the concentration of the proteins is bounded by thresholds. More precisely, a regulatory domain $D \subseteq \Omega$ is defined by $D = D_1 \times \dots \times D_n$, where every D_i , $1 \leq i \leq n$, is defined by one of the equations below:

$$\begin{aligned} D_i &= \{x_i \mid 0 \leq x_i < \theta_i^1\}, \\ D_i &= \{x_i \mid \theta_i^1 < x_i < \theta_i^2\}, \\ &\dots \\ D_i &= \{x_i \mid \theta_i^{p_i} < x_i \leq \max x_i\}. \end{aligned}$$

Δ_r denotes the set of regulatory domains in Ω . As can be easily verified, there are $|\Delta_r| = \prod_{i=1}^n (p_i + 1)$ regulatory domains in Ω .

In figure 3(a) the two-dimensional phase space box Ω for the example network is shown. As proteins A and B have two thresholds both, the phase space box is partitioned into 9 regulatory domains.

When evaluating the step function expressions in (3) in a regulatory domain, f_i and g_i reduce to sums of rate constants. More precisely, in every $D \in \Delta_r$, f_i reduces to some $\mu_i^D \in M_i \equiv \{f_i(\mathbf{x}) \mid \mathbf{0} \leq \mathbf{x} \leq \mathbf{max}\mathbf{x}\}$, and g_i to some $\nu_i^D \in N_i \equiv \{g_i(\mathbf{x}) \mid \mathbf{0} \leq \mathbf{x} \leq \mathbf{max}\mathbf{x}\}$. M_i and N_i collect the synthesis and degradation rates of the protein in different domains of Ω . Inside a regulatory domain D , the state equations thus simplify to linear and uncoupled differential equations

$$\dot{x}_i = \mu_i^D - \nu_i^D x_i, \quad 1 \leq i \leq n, \quad (7)$$

or, equivalently,

$$\dot{\mathbf{x}} = \boldsymbol{\mu}^D - \boldsymbol{\nu}^D \mathbf{x}, \quad (8)$$

where $\boldsymbol{\mu}^D = (\mu_1^D, \dots, \mu_n^D)'$ and $\boldsymbol{\nu}^D = \text{diag}(\nu_1^D, \dots, \nu_n^D)$.

Let $\boldsymbol{\theta} = (\dots, \theta_i^{k_i}, \dots)'$, $\boldsymbol{\kappa} = (\dots, \kappa_{il}, \dots)'$, and $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_n)'$ be numerical parameter values. Furthermore, let $\mathbf{x}(0) = \mathbf{x}_0$ be a point in Ω representing the initial conditions. A continuously differentiable function $\mathbf{x}(t) = \boldsymbol{\xi}(t, 0, \mathbf{x}_0, \boldsymbol{\theta}, \boldsymbol{\kappa}, \boldsymbol{\gamma})$ is a solution of (8) on a time-interval $[0, \tau[$, $\tau > 0$, if $\mathbf{x}(0) = \mathbf{x}_0$ and for all $t \in [0, \tau[$ it holds that $\mathbf{x}(t) \in D$ and $\dot{\mathbf{x}}(t) = \boldsymbol{\mu}^D - \boldsymbol{\nu}^D \mathbf{x}(t)$. For initial values $\mathbf{x}_0 \in D$, there exists a function $\mathbf{x}(t)$ and a $\tau > 0$, such that $\mathbf{x}(t)$ is the unique solution for (8) on $[0, \tau[$.

Let ϕ_i be a function from Δ_r to Ω_i , defined as $\phi_i(D) = \mu_i^D / \nu_i^D$. Analysis of (8) shows that all solution trajectories in D monotonically tend towards a *target equilibrium*, a stable equilibrium given

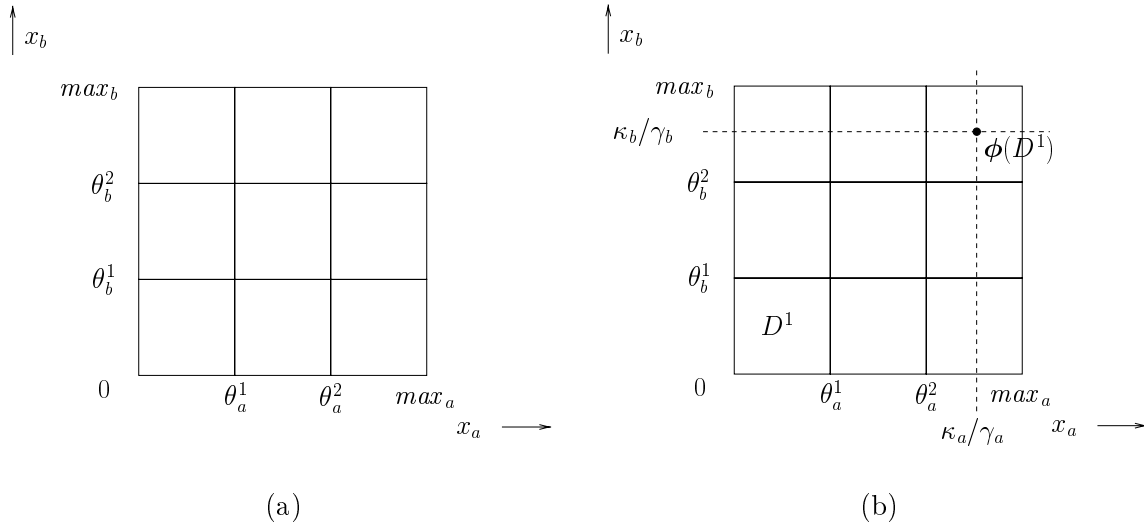


Figure 3: (a) Phase space box Ω for the genetic regulatory network in figure 2. (b) Phase space box with the target equilibrium for the regulatory domain $D^1 = \{(x_a, x_b) \in \mathbb{R}^2 \mid 0 \leq x_a < \theta_a^1, 0 \leq x_b < \theta_b^1\}$. The variables x_a and x_b tend towards equilibrium values κ_a/γ_a and κ_b/γ_b . The target equilibrium is assumed to lie in the upperright domain.

by $\mathbf{x} = \phi(D)$, with $\phi = (\phi_1, \dots, \phi_n)'$ [12, 20, 44, 48]. Intuitively speaking, the target equilibrium level $\phi_i(D)$ of x_i gives an indication of the strength of gene expression in the regulatory domain. If $\phi(D) \in D$, then for $t \rightarrow \infty$ all trajectories in D approach a target equilibrium in D , which represents a so-called *regular* steady state [57] of the system. If $\phi(D) \notin D$, all trajectories will at some point leave D . In what follows, we will make the generic assumption that target equilibria are not located in threshold planes.

In the example, as can be easily checked from the state equations, we have $M_a = \{0, \kappa_a\}$, $N_a = \{\gamma_a\}$ for protein A, and $M_b = \{0, \kappa_b\}$, $N_b = \{\gamma_b\}$ for protein B. In the regulatory domain D^1 in figure 3(b), the state equations simplify to

$$\begin{aligned}\dot{x}_a &= \kappa_a - \gamma_a x_a, \\ \dot{x}_b &= \kappa_b - \gamma_b x_b.\end{aligned}$$

As a consequence, the target equilibrium $\phi(D^1)$ of D^1 equals $(\kappa_a/\gamma_a, \kappa_b/\gamma_b)'$, which lies outside D^1 . The trajectories in D^1 will therefore leave the domain at some point. Different regulatory domains generally have different target equilibria. For instance, in the regulatory domain defined by $0 \leq x_a < \theta_a^1$ and $\theta_b^1 < x_b < \theta_b^2$, the target equilibrium is given by $(0, \kappa_b/\gamma_b)'$.

3 Extension of piecewise-linear models to deal with discontinuities

The global solution of (2) could be obtained by piecing together the local solutions in regulatory domains, in such a way as to guarantee continuity of the global solution across the threshold hyperplanes [12, 56]. This works fine as long as trajectories arriving at a threshold hyperplane can be continued in another regulatory domain, e.g., trajectories arriving at the threshold hyperplane $x_b = \theta_b^1$ from the regulatory domain D^1 (figure 4(a)). However, when the trajectories on both sides of a threshold hyperplane evolve towards this plane, as in the case of trajectories arriving at the threshold hyperplane $x_b = \theta_b^2$ from regulatory domain D^3 or D^5 (figure 4(b)), mathematical perplexities arise. In the

framework of the previous section, there is no indication on how the local solutions in D^3 and D^5 can be continued.²

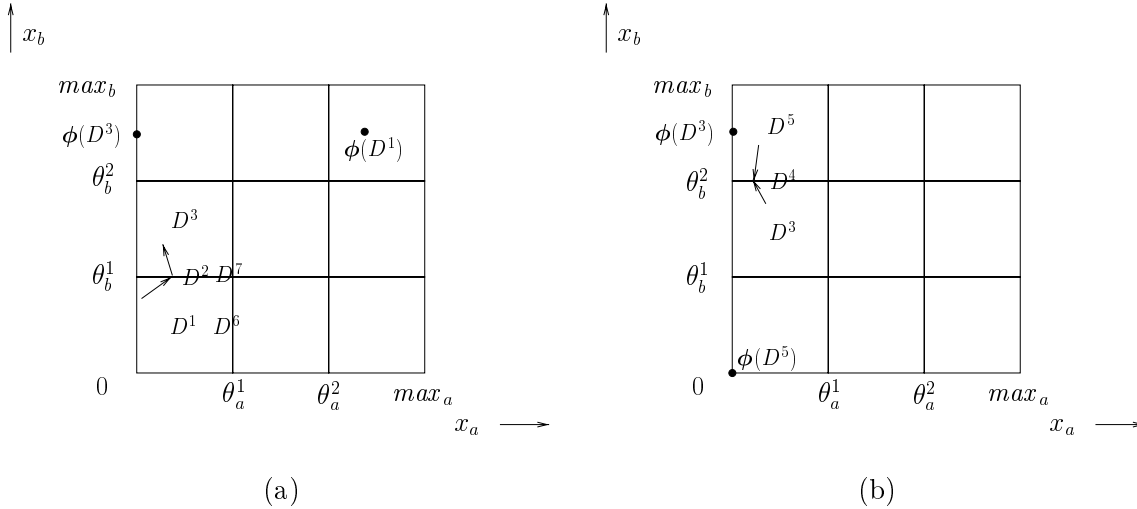


Figure 4: Examples of the behavior of the system of figure 1 at threshold hyperplanes. The figures show the regulatory domains D^1, \dots, D^5 , and the target equilibria $\phi(D^1)$, $\phi(D^3)$, $\phi(D^5)$. In (a) the solution trajectories in D^1 can be continued in D^3 , whereas in (b) the solution trajectories in D^3 cannot be continued in D^5 .

The troubles at the threshold hyperplanes are caused by discontinuities in the right-hand side of (2), due to the use of step functions. In order to deal with these discontinuities, we will use a method originally proposed by Filippov [15]. This method, recently applied by Gouzé and Sari [27] to PL systems of the form (2), consists of extending a system of differential equations with discontinuous right-hand sides into a system of differential inclusions.

We first refine the description of Ω by introducing *switching domains* located in the threshold hyperplanes. More precisely, a switching domain $D \subseteq \Omega$ is defined by $D = D_1 \times \dots \times D_n$, where every D_i , $1 \leq i \leq n$, is defined by one of the equations below:

$$\begin{aligned}
 D_i &= \{x_i \mid 0 \leq x_i < \theta_i^1\}, \\
 D_i &= \{x_i \mid x_i = \theta_i^1\}, \\
 D_i &= \{x_i \mid \theta_i^1 < x_i < \theta_i^2\}, \\
 D_i &= \{x_i \mid x_i = \theta_i^2\}, \\
 &\dots \\
 D_i &= \{x_i \mid x_i = \theta_i^{p_i}\}, \\
 D_i &= \{x_i \mid \theta_i^{p_i} < x_i \leq \max_i\},
 \end{aligned} \tag{9}$$

while for at least one i , it holds that there is a j , $1 \leq j \leq p_i$, such that $D_i = \{x_i \mid x_i = \theta_i^j\}$. The corresponding variables x_i are called *switching variables*.³ The *order* of a switching domain is a number between 1 and n , equal to the number of switching variables. Δ_s denotes the set of switching domains in Ω , while $\Delta = \Delta_r \cup \Delta_s$ denotes the set of all domains in Ω . As can be easily verified, $|\Delta| = \prod_{i=1}^n (2p_i + 1)$, which with $|\Delta_r| = \prod_{i=1}^n (p_i + 1)$ gives $|\Delta_s| = \prod_{i=1}^n (2p_i + 1) - \prod_{i=1}^n (p_i + 1)$

²In [44, 48], D^2 is called a *transparent wall* and D^4 a *black wall*. Another problem, not shown here, occurs when a switching domain is a *white wall*, that is, when the trajectories in two regulatory domains evolve from a common bounding threshold hyperplane.

³In [44] switching domains are called Δ -regions and switching variables are referred to as *primary variables*.

The phase space box in figure 3 consists of 9 regulatory and 16 switching domains. An example of a switching domain in figure 4 is the set $D^4 = \{(x_a, x_b) \in \mathbb{R}^2 \mid 0 \leq x_a < \theta_a^1, x_b = \theta_b^2\}$, which separates the regulatory domains D^3 and D^5 . Notice that x_b is the (only) switching variable in D^4 .

Let D be a switching domain of order k . Let C be the hyperplane of dimension $n - k$ containing D . The *boundary* of D in C is the set $B(D)$ of all points $\mathbf{x} \in C$, such that each ball $B_C(\mathbf{x}, \varepsilon)$ in C of center \mathbf{x} and radius $\varepsilon > 0$ intersects both D and $C \setminus D$ [33]. In the case that D is a regulatory domain, C equals Ω .

Now, for every $D \in \Delta$ we define the sets

$$A(D) = \{D' \in \Delta \mid D' \subseteq B(D)\}, \text{ and}$$

$$R(D) = \{D' \in \Delta \mid D' \text{ regulatory domain, } D \subseteq B(D')\}.$$

$A(D)$ contains the domains in the boundary of D , whereas $R(D)$ contains the regulatory domains that have D in their boundary.

In the case of the regulatory domain D^1 in figure 4, we find $A(D^1) = \{D^2, D^6, D^7\}$, while $A(D^2) = \{D^7\}$. Furthermore, $R(D^1) = \{\}$ and $R(D^2) = \{D^1, D^3\}$.

The basic idea of the Filippov approach is to extend the differential equations (2) into differential inclusions

$$\dot{\mathbf{x}} \in \mathbf{H}(\mathbf{x}), \tag{10}$$

where $\mathbf{H} : \Omega \rightarrow S(\Omega)$ is a set-valued function.⁴

For $\mathbf{x} \in D$, and D a regulatory domain, we define $\mathbf{H}(\mathbf{x})$ simply as

$$\mathbf{H}(\mathbf{x}) = \{\boldsymbol{\mu}^D - \boldsymbol{\nu}^D \mathbf{x}\}. \tag{11}$$

Notice that, since the set $\mathbf{H}(\mathbf{x})$ contains a single element, the extension of the PL system agrees with the original system in the regulatory domains. If D is a switching domain, $\mathbf{H}(\mathbf{x})$ is defined by

$$\mathbf{H}(\mathbf{x}) = \overline{\text{co}}(\{\boldsymbol{\mu}^{D'} - \boldsymbol{\nu}^{D'} \mathbf{x} \mid D' \in R(D)\}). \tag{12}$$

The smallest closed convex set $\overline{\text{co}}(E)$ of a set E is the intersection of all closed convex sets containing E [15]. In the case of switching domains, $\mathbf{H}(\mathbf{x})$ will not generally be single-valued.

An absolutely continuous function $\mathbf{x}(t) = \boldsymbol{\xi}(t, 0, \mathbf{x}_0, \boldsymbol{\theta}, \boldsymbol{\kappa}, \boldsymbol{\gamma})$ is a solution of (10) in the sense of Filippov on $[0, \tau[$, $\tau > 0$, if $\mathbf{x}(0) = \mathbf{x}_0$ and for almost all $t \in [0, \tau[$ it holds that $\dot{\mathbf{x}}(t) \in \mathbf{H}(\mathbf{x}(t))$ [15]. The qualification ‘for almost all $t \in [0, \tau[$ ’ means that the set of time-points for which the condition does not hold is of measure 0. In particular, the condition is not satisfied at time-points when the solution arrives at a switching domain D , or leaves a switching domain D . If no misunderstanding is possible, we will often simply speak of ‘a solution of (10),’ instead of ‘a solution of (10) in the sense of Filippov.’ For all initial values $\mathbf{x}_0 \in \Omega$ there exists a solution of (10) on some $[0, \tau[$ [15]. However, this solution is not guaranteed to be unique, due to the generalization of the differential equations to differential inclusions.

In order to get an intuitive feeling for the meaning of the above concepts, consider again the examples in figure 4. In the first case, D^2 is the switching domain bounding the regulatory domains D^1 and D^3 . If $\mathbf{x} \in D^2$, then $\mathbf{H}(\mathbf{x})$ is the smallest closed convex set including the end-points of the vectors $\boldsymbol{\mu}^{D^1} - \boldsymbol{\nu}^{D^1} \mathbf{x}$ and $\boldsymbol{\mu}^{D^3} - \boldsymbol{\nu}^{D^3} \mathbf{x}$ starting at \mathbf{x} . This set is graphically represented in figure 5(a) by the linear segment connecting the end-points of the vectors. $\mathbf{H}(\mathbf{x})$ is multiple-valued but, as can be seen, $\dot{x}_b > 0$ for any $\dot{\mathbf{x}} \in \mathbf{H}(\mathbf{x})$, so that a solution trajectory arriving at \mathbf{x} crosses the threshold plane instantaneously. In the second example of figure 4, $\mathbf{H}(\mathbf{x})$ is a linear segment connecting the end-points of the vectors $\boldsymbol{\mu}^{D^3} - \boldsymbol{\nu}^{D^3} \mathbf{x}$ and $\boldsymbol{\mu}^{D^5} - \boldsymbol{\nu}^{D^5} \mathbf{x}$. However, this time $\mathbf{H}(\mathbf{x})$ intersects with

⁴For a set E , $S(E)$ represents the set of subsets of E .

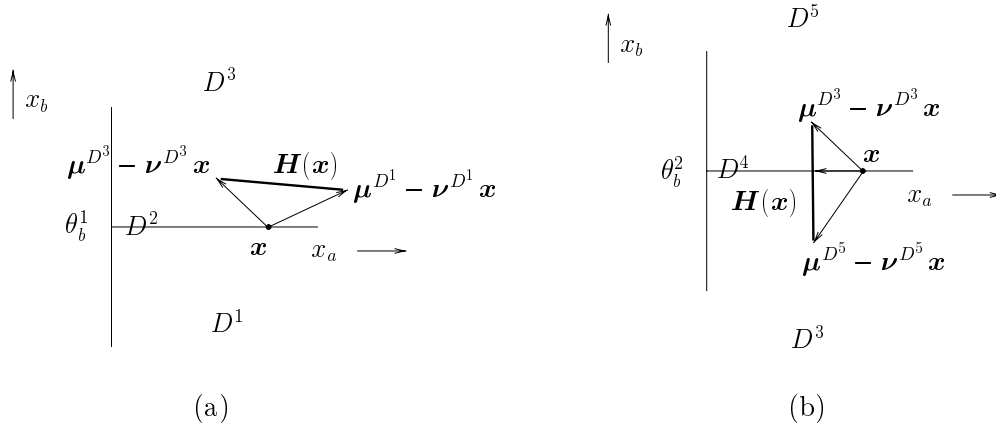


Figure 5: Behavior of the system of figure 1 at a point \mathbf{x} in a threshold hyperplane, when the differential equations are generalized into differential inclusions by the method of Filippov. Whereas in (a) solution trajectories cross D^2 instantaneously, in (b) they slide along D^4 .

the threshold plane (figure 5(b)). Because the vector fields in D^3 and D^5 are directed towards D^4 , a solution trajectory arriving at \mathbf{x} continues by sliding along the threshold plane with a derivative vector given by the intersection of $\mathbf{H}(\mathbf{x})$ and D^4 , displaying a so-called *sliding mode* behavior.

For every domain D , a so-called *target equilibrium set* $\Phi(D)$ can be defined. If D is a regulatory domain, then

$$\Phi(D) = \{\phi(D)\}. \quad (13)$$

If D is a switching domain, the definition is a little bit more complicated. Let D be a switching domain of order k , contained in the $n - k$ -dimensional hyperplane C . Then

$$\Phi(D) = C \cap \overline{\text{co}}(\{\phi(D') \mid D' \in R(D)\}). \quad (14)$$

That is, $\Phi(D)$ is the smallest closed convex set of the target equilibria of regulatory domains D' having D in their boundary, intersected with the hyperplane containing D .

As illustrated in figure 5, a solution may (a) instantaneously cross a switching domain or (b) remain in the domain for some time $\tau > 0$, sliding along the threshold hyperplane containing the domain. Obviously, the latter case is more interesting when analyzing the behaviour of regulatory systems. Gouzé and Sari [27] have shown that the latter *sliding mode* solutions exist in a switching domain D , iff

$$\Phi(D) \neq \{\}. \quad (15)$$

The sliding mode solutions monotonically tends towards the target equilibrium set $\Phi(D)$ [27]. Because $\Phi(D)$ does not generally include a single point, the behavior of the system is not uniquely determined by the differential inclusion (10). For example, if a solution trajectory rests in D on the interval $[0, \tau[$, it might tend towards one target equilibrium in $\Phi(D)$ on the subinterval $[0, \tau'[$, and to another target equilibrium in $\Phi(D)$ on the subinterval $[\tau', \tau[$ ($0 < \tau' < \tau$). In the special case that $\Phi(D) \cap D \neq \{\}$, there exist solutions in D that asymptotically approach a target equilibrium in D . The equilibrium represents a so-called *singular* steady state [57] of the system. Whether the equilibrium is stable or unstable must be determined through further analysis.

Consider the examples in figure 6. The target equilibrium set $\Phi(D^2)$ of the switching domain D^2 is defined, following (14), by the intersection of $\overline{\text{co}}(\{\phi(D^1), \phi(D^3)\})$ and the threshold hyperplane $x_b = \theta_b^1$. The smallest closed convex set consists of the linear segment connecting the points $(\kappa_a/\gamma_a, \kappa_b/\gamma_b)'$ and $(0, \kappa_b/\gamma_b)'$, as shown in (a). $\Phi(D^2)$ and the threshold boundary $x_b = \theta_b^1$ do not intersect in the

figure, so $\Phi(D^2) = \{\}$. Following criterion (15), there are no sliding mode solutions in D^2 . This is different in the case of D^4 . Here, the target equilibrium set $\Phi(D^2)$ is given by the intersection of $\overline{co}(\{\phi(D^3), \phi(D^5)\})$, the linear segment connecting the points $(0, \kappa_b/\gamma_b)'$ and $(0, 0)'$, and the threshold boundary $x_b = \theta_b^2$. Consequently, $\Phi(D^4)$ equals $\{(0, \theta_b^2)'\}$, and there exists a (unique) sliding mode solution in D^4 , tending towards $(0, \theta_b^2)$. Because the target equilibrium lies inside D^4 , it is also a singular steady state of the system. Closer analysis reveals that the equilibrium $(0, \theta_b^2)'$ is stable, since all trajectories in its neighbourhood end up in this point.

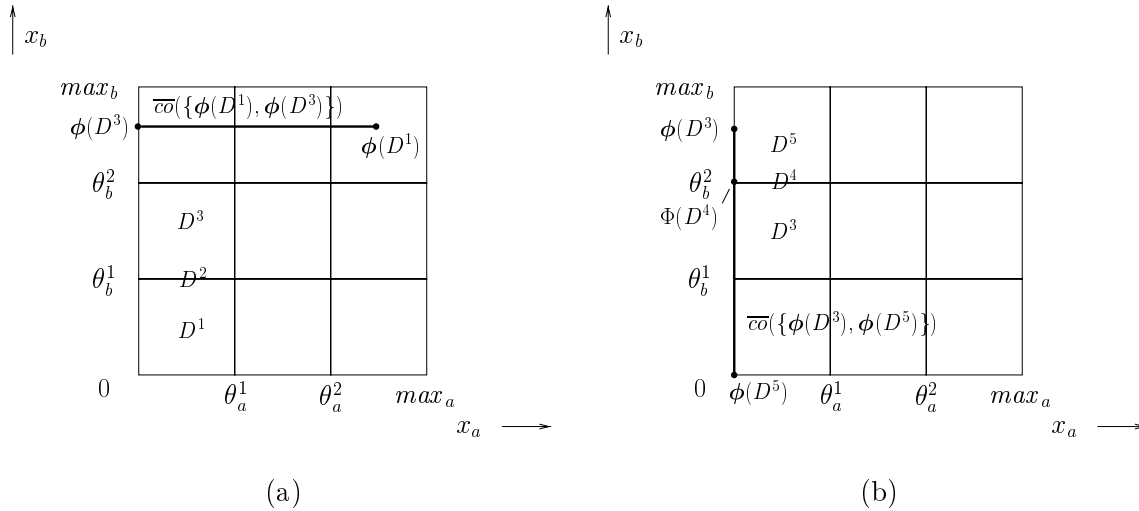


Figure 6: Determination of the target equilibrium sets (a) $\Phi(D^2) = \{\}$ and (b) $\Phi(D^4) = \{(0, \theta_b^2)'\}$.

In order to appreciate the intuitive validity of the Filippov approach, suppose that the step functions $s^+(x_j, \theta_j)$ in (2) are replaced by sigmoid functions $h^+(x_j, \theta_j, \sigma)$, defined by

$$h^+(x_j, \theta_j, \sigma) = \frac{x_j^\sigma}{x_j^\sigma + \theta_j^\sigma}. \quad (16)$$

The step functions and sigmoid functions (called *Hill functions*) are compared in figure 7. Replacing step functions by sigmoid functions results in differential equation models that are nonlinear but continuous. For steep sigmoid functions, the behavior of the continuous, nonlinear system resembles the behavior of the discontinuous, piecewise-linear system, as shown in figure 8 for the example system (see also [22, 23, 48]). In the continuous case, in the neighbourhood of the plane $x_b = \theta_b^2$, the solution trajectories tend towards an equilibrium located near $(0, \theta_b^2)'$. A mathematically rigorous comparison of the behavior of discontinuous systems and their continuous homologues can be found in the book of Filippov [15].



Figure 7: Different threshold functions: (a) step function $s^+(x_j, \theta_j)$ and (b) sigmoid function $h^+(x_j, \theta_j, \sigma)$. The sigmoid function approaches the step function as $\sigma \rightarrow \infty$.

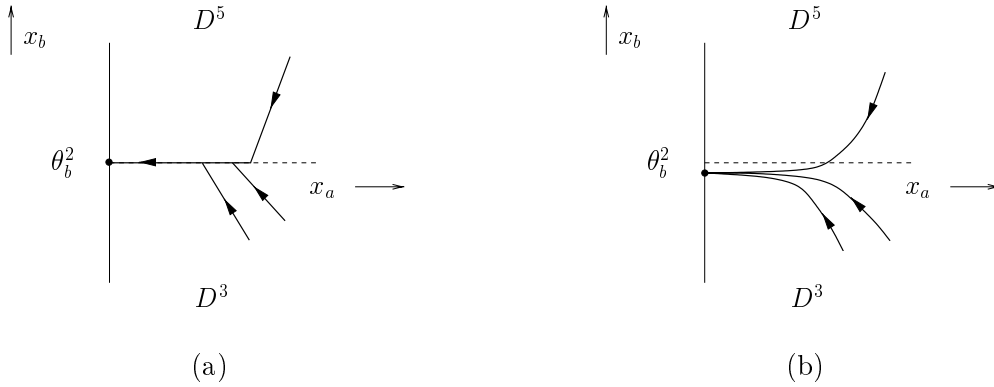


Figure 8: Behavior of the system of figure 1 near the threshold plane $x_b = \theta_b^2$. In (a) the system is described by the state equations in figure 2, whereas in (b) the system is described by the same state equations, but with the step functions replaced by the sigmoid functions.

4 Qualitative constraints on parameter values

Most of the time, precise numerical values for the threshold and rate parameters in (2) are not available. As a consequence, it is not possible to numerically simulate the behavior of a genetic regulatory network. Rather than numerical values, we will specify qualitative constraints on the parameter values. These constraints, having the form of algebraic inequalities, can usually be inferred from biological data. They are exploited by the simulation method to predict the qualitative dynamics of the regulatory system.

First of all, we can order the p_i threshold concentrations of gene i , yielding the so-called *threshold inequalities*

$$0 < \theta_i^1 < \dots < \theta_i^{p_i} < \max_i. \quad (17)$$

In the case of protein A, there are two threshold concentrations: θ_a^1 is the threshold for the repression of gene b , while θ_a^2 is the threshold for the repression of gene a . Assuming the first to be lower than the second, we obtain the threshold inequalities $0 < \theta_a^1 < \theta_a^2 < \max_a$. The ordering of the thresholds of protein B can be determined likewise, giving rise to $0 < \theta_b^1 < \theta_b^2 < \max_b$, with θ_b^1 being the threshold for gene a repression and θ_b^2 the threshold for gene b autorepression (figure 9).

Second, the possible target equilibrium levels of x_i in different regulatory domains $D \in \Delta_r$ can be ordered with respect to the threshold concentrations. The resulting *equilibrium inequalities* define the strength of gene expression in the domain in a qualitative way, on the scale of ordered threshold concentrations. More precisely, for every $\mu_i \in M_i$, $\nu_i \in N_i$, and $\mu_i, \nu_i \neq 0$, we specify one of the following pairs of inequalities:

$$\begin{aligned} 0 < \mu_i/\nu_i < \theta_i^1, \\ \theta_i^1 < \mu_i/\nu_i < \theta_i^2, \\ \dots \\ \theta_i^{p_i} < \mu_i/\nu_i < \max_i. \end{aligned} \quad (18)$$

The equilibrium inequalities for the example model are shown in figure 9. In the absence of protein B ($s^-(x_b, \theta_b^1) = 1$), while protein A has not yet reached its highest level ($s^-(x_a, \theta_a^2) = 1$), gene a is expressed at a rate κ_a . The corresponding target equilibrium value κ_a/γ_a of x_a must be above the second threshold θ_a^2 , otherwise the concentration of the protein would not be able to reach or maintain a level at which the observed negative autoregulation of gene a occurs (i.e., $\theta_a^2 < \kappa_a/\gamma_a < \max_a$).

State equation for gene a :

$$\dot{x}_a = \kappa_a s^-(x_a, \theta_a^2) s^-(x_b, \theta_b^1) - \gamma_a x_a$$

Threshold inequalities: $0 < \theta_a^1 < \theta_a^2 < \max x_a$

Equilibrium inequalities: $\theta_a^2 < \frac{\kappa_a}{\gamma_a} < \max x_a$

State equation for gene b :

$$\dot{x}_b = \kappa_b s^-(x_a, \theta_a^1) s^-(x_b, \theta_b^2) - \gamma_b x_b$$

Threshold inequalities: $0 < \theta_b^1 < \theta_b^2 < \max x_b$

Equilibrium inequalities: $\theta_b^2 < \frac{\kappa_b}{\gamma_b} < \max x_b$

Figure 9: State equations, threshold inequalities, and equilibrium inequalities for the network of figure 1.

In a similar way, the target equilibrium value κ_b/γ_b is positioned above θ_b^2 , again to ensure that the negative autoregulation of gene b at high concentrations of protein B can occur.

A *quantitative* PL model of a genetic regulatory network consists of the state equations (2) and numerical parameter values θ, κ, γ . In a *qualitative* PL model, on the other hand, the state equations are supplemented by parameter inequalities (17) and (18). Every quantitative PL model can be abstracted into a unique qualitative PL model. The values of θ can be uniquely ordered, giving rise to the threshold inequalities (17), while ratios of values of κ and γ can be uniquely ordered with respect to the threshold values, yielding the equilibrium inequalities. Conversely, a qualitative PL model corresponds to a set of quantitative PL models.

5 Qualitative states and behaviors

An intuitive qualitative description of the state of a regulatory system consists of the domain in which the system resides, supplemented by the position with respect to this domain of the target equilibrium set to which the state of the system tends. A qualitative behavior is then given by the sequence of qualitative states traversed by the system. This idea will be elaborated below.

We first define a function $v : \Delta \times \Omega \rightarrow \{-1, 0, 1\}^n$ that maps a domain D and a point e to a sign vector $v(D, e)$ describing the relative position of D and e . Let $v(D, e)_i$ denote the i th component of this vector. If x_i is a non-switching variable, then

$$v(D, e)_i = \begin{cases} 1 & , \text{ if } e_i \geq \sup D_i, \\ 0 & , \text{ if } \inf D_i < e_i < \sup D_i, \\ -1 & , \text{ if } e_i \leq \inf D_i. \end{cases}$$

On the other hand, if x_i is a switching variable, then $D_i = \{\theta_i^j\}$, and

$$v(D, \mathbf{e})_i = \begin{cases} 1 & , \text{ if } e_i > \theta_i^j, \\ 0 & , \text{ if } e_i = \theta_i^j, \\ -1 & , \text{ if } e_i < \theta_i^j. \end{cases}$$

Generalizing this definition, we obtain the set function $V : \Delta \times S(\Omega) \rightarrow S(\{-1, 0, 1\}^n)$ that maps a domain D and a set E to a set of sign vectors:

$$V(D, E) = \{v(D, e) \mid e \in E\}. \quad (19)$$

Let $\mathbf{x}(t) = \boldsymbol{\xi}(t, 0, \mathbf{x}_0, \boldsymbol{\theta}, \boldsymbol{\kappa}, \boldsymbol{\gamma})$ be the solution of a quantitative PL model describing a regulatory network on the time-interval $[0, \tau[$. Now suppose that for some t , $0 \leq t < \tau$, we have $\mathbf{x}(t) \in D$, $D \in \Delta$. The point $\mathbf{x}(t)$ corresponds to a *qualitative state* of the system defined by

$$QS(\mathbf{x}, t) = \langle D, V(D, \Phi(D)) \rangle. \quad (20)$$

The solution $\mathbf{x}(t)$ on $[0, \tau[$ remains in Ω (section 2), and hence passes through a sequence of domains D^0, \dots, D^m . The corresponding sequence of qualitative states is called the *qualitative behavior* of the system on the time-interval. More specifically, a qualitative behavior of the system is defined by

$$QB(\mathbf{x}, 0, \tau) = (\langle D^0, V(D^0, \Phi(D^0)) \rangle, \dots, \langle D^m, V(D^m, \Phi(D^m)) \rangle). \quad (21)$$

Consider the solution trajectory in figure 10, obtained for given parameter values, which moves from an initial state in D^1 towards a stable equilibrium in D^4 . Following the above definitions, the solution can be abstracted into a qualitative behavior, as shown in the figure. In the regulatory domain D^1 , we have $\Phi(D^1) = \{(\kappa_a/\gamma_a, \kappa_b/\gamma_b)'\}$. For the parameter values in figure 10, we find $\kappa_a/\gamma_a > \theta_a^2 > \theta_a^1$ and $\kappa_b/\gamma_b > \theta_b^2 > \theta_b^1$. As a consequence, $V(D^1, \Phi(D^1)) = \{(1, 1)\}$, and we have a qualitative state $QS^1 = \langle D^1, \{(1, 1)\} \rangle$. In the case of the switching domain D^2 , the smallest closed convex set of the target equilibria in D^1 and D^3 consists of the linear segment connecting $(\kappa_a/\gamma_a, \kappa_b/\gamma_b)'$ and $(0, \kappa_b/\gamma_b)'$. For the parameter values in figure 10, this segment does not intersect with $x_b = \theta_b^1$, so that $V(D^2, \Phi(D^2)) = \{\}$. The resulting qualitative state is $QS^2 = \langle D^2, \{\} \rangle$. The target equilibrium set of the regulatory domain D^3 is given by $\Phi(D^3) = \{(0, \kappa_b/\gamma_b)'\}$, which leads to $V(D^3, \Phi(D^3)) = \{(0, 1)\}$, and $QS^3 = \langle D^3, \{(0, 1)\} \rangle$. In the case of D^4 , the smallest closed convex set containing $\phi(D^3)$ and $\phi(D^5)$ consists of the linear segment connecting $(0, \kappa_b/\gamma_b)'$ and $(0, 0)'$. This set intersects with $x_b = \theta_b^2$ at $(0, \theta_b^2)'$, so that $V(D^4, \Phi(D^4)) = \{(0, 0)\}$. We have a qualitative state $QS^4 = \langle D^4, \{(0, 0)\} \rangle$.

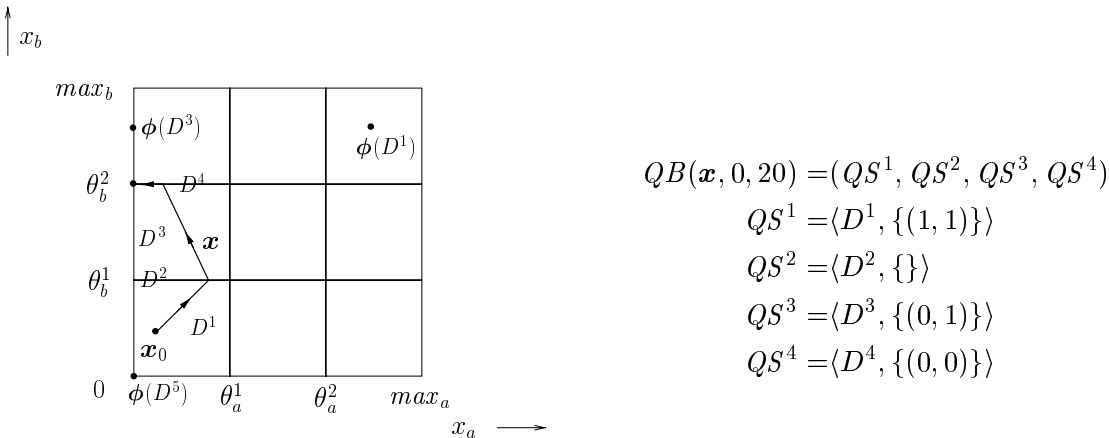


Figure 10: Abstraction of a solution of the PL model in figure 9 into a qualitative behavior. The solution trajectory has been obtained for the parameter values $\theta_a^1 = 4$, $\theta_a^2 = 8$, $\theta_b^1 = 3$, $\theta_b^2 = 7$, $\kappa_a = 20$, $\kappa_b = 16$, $\gamma_a = 2$, $\gamma_b = 2$, and the initial conditions $\mathbf{x}_0 = (1.25, 2)$ on the time-interval $[0, 20[$.

The example shows how a numerical solution of the system can be abstracted into a qualitative behavior consisting of a sequence of qualitative states. More generally, every solution \mathbf{x} of a quantitative PL model on a time-interval $[0, \tau[$ can be abstracted into a unique qualitative behavior $QB(\mathbf{x}, 0, \tau)$. The solution traverses a unique sequence of domains D^0, \dots, D^m , while to every domain D^j , $0 \leq j \leq m$, there corresponds a unique set of sign vectors $V(D^j, \Phi(D^j))$. Conversely, a single qualitative behavior usually corresponds to a set of solutions.

Some properties of the set of solutions abstracting to a qualitative behavior can be directly inferred from the behavior description. Let QB be a qualitative behavior containing a qualitative state $\langle D, V(D, \Phi(D)) \rangle$ associated with a domain $D \in \Delta$.

Proposition 1 Let D be a switching domain. There exist sliding mode solutions in D , iff $V(D, \Phi(D)) \neq \{\}$.

Proof. The condition $V(D, \Phi(D)) \neq \{\}$ means that $\Phi(D) \neq \{\}$. This is exactly condition (15) for the existence of sliding mode solutions in switching domains. \square

Proposition 2 There exist solutions in D that asymptotically approach an equilibrium point in D , iff $\mathbf{0} \in V(D, \Phi(D))$.

Proof. If D is a regulatory domain, then $\Phi(D) = \{\phi(D)\}$. All trajectories asymptotically approach an equilibrium point in D , iff $\Phi(D) \cap D \neq \{\}$. This is the case, iff for all x_i , $\inf D_i < \phi_i(D) < \sup D_i$, and hence $v(D, \phi(D)) = \mathbf{0}$. If D is a switching domain, then $\Phi(D) = C \cap \overline{co}(\{\phi(D') \mid D' \in R(D)\})$. $\Phi(D) \cap D \neq \{\}$, iff $\Phi(D) \neq \{\}$ and there is some $\mathbf{r} \in \Phi(D)$, such that for all non-switching variables x_i , $\inf D_i < r_i < \sup D_i$. That is, there is some $\mathbf{r} \in \Phi(D)$, such that $v(D, \mathbf{r}) = \mathbf{0}$. \square

6 Qualitative simulation

Given a qualitative PL model and initial conditions in a domain D^0 , one can ask what are the possible qualitative behaviors of the system. Determining these qualitative behaviors is the aim of *qualitative simulation*. Phrased in a different way, denoting by X the set of solutions $\mathbf{x}(t)$ on some time-interval $[0, \tau[$ of all quantitative PL models corresponding to the qualitative model, such that $\mathbf{x}(0) = \mathbf{x}_0 \in D^0$, the aim of qualitative simulation is to find the set of qualitative behaviors that abstract from some $\mathbf{x} \in X$. Although not essential for the sequel, we will make the assumption that τ is sufficiently large for obtaining an idea of the long-term behavior of the system.

The simulation algorithm described in this section generates a set of qualitative behaviors by recursively determining transitions from qualitative states, starting from the qualitative state associated with the initial domain D^0 . In order to achieve this, two issues need to be addressed. First of all, how can we determine the qualitative state associated with a domain from the constraints on the parameters? Second, how can we find the possible transitions from this qualitative state? In the next section, we will show that the set of qualitative behaviors generated by the simulation algorithm contains all qualitative behaviors abstracting from some $\mathbf{x} \in X$.

6.1 Computation of qualitative states

In order to determine the qualitative state associated with a domain D , we need to compute $V(D, \Phi(D))$. In the absence of numerical values for the parameters, we cannot give a numerical approximation of $\Phi(D)$ and then abstract this region into a set of sign vectors. Instead, we have to infer $V(D, \Phi(D))$ from the inequalities (17) and (18) constraining the parameter values. This is a difficult problem in general, because $\Phi(D)$ may be a complex polyhedron in Ω , and the relative position of $\Phi(D)$ with respect to the threshold planes underdetermined by the parameter inequalities.

Instead of $V(D, \Phi(D))$, we will calculate $V(D, \Psi(D))$, where $\Psi(D) \subseteq \Omega$ is the smallest closed hyperrectangle including $\Phi(D)$. Like $\Phi(D)$, $\Psi(D)$ is a convex set. If D is a regulatory domain, then $\Psi(D) = \{\phi(D)\}$. On the other hand, if D is a switching domain, we obtain

$$\Psi(D) = C \cap \overline{\text{rect}}(\{\phi(D') \mid D' \in R(D)\}), \quad (22)$$

where C is the $n-k$ -dimensional hyperplane containing D . $\overline{\text{rect}}(E)$ is the smallest closed hyperrectangle containing the set E . Although in general $\Psi(D)$ will be an overapproximation of $\Phi(D)$, $V(D, \Psi(D))$ is straightforward to compute from the parameter inequalities and uniquely determined by the latter. In the next section, we will discuss the consequences of the overapproximation of $\Phi(D)$ by $\Psi(D)$.

In the special case that D is a regulatory domain, $\Psi(D)$ is a single point in Ω , and $V(D, \Psi(D)) = \{v(D, \phi(D))\}$. The equilibrium inequalities tell us whether $\phi_i(D) \geq \sup D_i$, $\inf D_i < \phi_i(D) < \sup D_i$, or $\phi_i(D) \leq \inf D_i$, which directly gives the sign vector representing $v(D, \phi(D))$.

Consider the case of $V(D^1, \Psi(D^1)) = \{v(D^1, \phi(D^1))\}$ in the example system. $\phi(D^1)$ was shown to be $(\kappa_a/\gamma_a, \kappa_b/\gamma_b)'$. From the threshold and equilibrium inequalities in figure 9, we know that $\kappa_a/\gamma_a > \theta_a^1$ and $\kappa_b/\gamma_b > \theta_b^1$. As a consequence, $v(D^1, \phi(D^1)) = (1, 1)$. The resulting qualitative state is $\langle D^1, \{(1, 1)\} \rangle$.

If D is a switching domain, $\Psi(D)$ is a hyperrectangle in Ω . Because of the hyperrectangular shape of $\Psi(D)$, we can write $V(D, \Psi(D))$ as $V(D, \Psi(D))_1 \times \dots \times V(D, \Psi(D))_n$, where $V(D, \Psi(D))_i \subseteq \{-1, 0, 1\}$, $1 \leq i \leq n$. Let $v(D, \phi(D'))_i$ represent the i th component of the sign vector $v(D, \phi(D'))$. Now, if x_i is a non-switching variable, then $V(D, \Psi(D))_i$ is given by

$$V(D, \Psi(D))_i = \{r_i \in \{-1, 0, 1\} \mid \min_{D' \in R(D)} v(D, \phi(D'))_i \leq r_i \leq \max_{D' \in R(D)} v(D, \phi(D'))_i\}. \quad (23)$$

On the other hand, if x_i is a switching variable, $V(D, \Psi(D))_i$ is given by

$$V(D, \Psi(D))_i = \{r_i \in \{-1, 0, 1\} \mid \min_{D' \in R(D)} v(D, \phi(D'))_i < r_i < \max_{D' \in R(D)} v(D, \phi(D'))_i\}. \quad (24)$$

As an example, consider the computation of $V(D^2, \Psi(D^2))$, where D^2 is a switching domain and $R(D^2) = \{D^1, D^3\}$ (figure 6). By means of the parameter inequalities in figure 9, we find that $v(D^2, \phi(D^1)) = (1, 1)$ and $v(D^2, \phi(D^3)) = (0, 1)$. Using (23)-(24), bearing in mind that x_b is a switching variable in D^2 , we derive $V(D^2, \Psi(D^2)) = \{0, 1\} \times \{\} = \{\}$. A second example concerns the computation of $V(D^4, \Psi(D^4))$, where D^4 is a switching domain and $R(D^4) = \{D^3, D^5\}$ (figure 6). By means of the parameter inequalities in figure 9, $v(D^4, \phi(D^3))$ and $v(D^4, \phi(D^5))$ are calculated to be $(0, 1)$ and $(0, -1)$, respectively. In this case, we find a set $V(D^4, \Psi(D^4)) = \{0\} \times \{0\} = \{(0, 0)\}$. The qualitative states associated to D^2 and D^4 are $\langle D^2, \{\} \rangle$ and $\langle D^4, \{(0, 0)\} \rangle$, respectively.

6.2 Computation of state transitions

A qualitative state $QS = \langle D, V(D, \Psi(D)) \rangle$ describes a domain and the position of the target equilibrium set with respect to the domain. Since the trajectories in D tend towards the target equilibrium set, this information can be exploited to determine which domains in the boundary of D , or domains that have D in their boundary, are accessible from D . Since each of these domains is itself associated with a qualitative state, this amounts to computing the possible *transitions* between qualitative states of the system.

The possible transitions are defined by two *transition rules*. The relative position of the domains D and D' is given by $V(D, D')$. As can be easily verified, $V(D, D')$ always consists of a single sign vector, that is, $V(D, D') = \{\mathbf{w}\}$. The domains D and D' are associated with qualitative states QS and QS' , calculated to be $\langle D, V(D, \Psi(D)) \rangle$ and $\langle D', V(D', \Psi(D')) \rangle$, respectively.

Transition rule 1 Let $D' \in A(D)$. There is a transition from QS to QS' , if

1. $V(D, \Psi(D)) \neq \{\}$, and
2. if x_i is a switching variable in D' , but not in D , then there is some $\mathbf{v} \in V(D, \Psi(D))$, such that $v_i w_i = 1$.

Transition rule 2 Let $D \in A(D')$. There is a transition from QS to QS' , if

1. $V(D', \Psi(D')) \neq \{\}$, and
2. if x_i is a switching variable in D , but not in D' , then there is some $\mathbf{v}' \in V(D', \Psi(D'))$, such that $v'_i w_i \neq -1$.

Intuitively, the first transition rule says that, in order to enter a switching domain D' in the boundary of D , some trajectories must tend towards D' (condition 2). If D is a switching domain, then there must exist sliding mode trajectories in D (condition 1). The second transition rule says that, in order to enter a domain D' from a switching domain D in the boundary of D' , the trajectories in D' must not tend towards D (condition 2). If D' is a switching domain, then there must exist sliding mode trajectories in D' (condition 1).

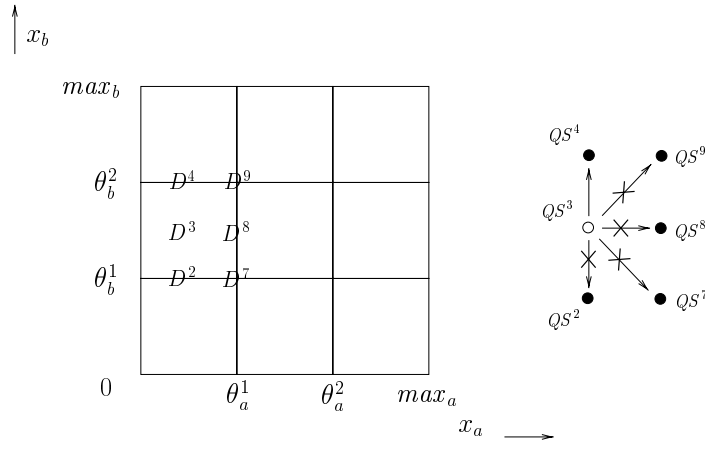
We will illustrate the rules by means of the two examples in figure 11, again derived from the model in figure 9. All qualitative states mentioned below have been computed in the way explained in section 6.1. Consider the possible transitions from the qualitative state QS^3 associated with regulatory domain D^3 to qualitative states associated with the boundary domains $A(D^3) = \{D^2, D^4, D^7, D^8, D^9\}$ in (a). We have to verify whether the conditions 1 and 2 of the first transition rule are verified. $V(D^3, \Psi(D^3))$ is calculated to be $\{(0, 1)\}$, while $V(D^3, D^4)$ equals $\{(0, 1)\}$. With x_b a switching variable in D^4 , but not in D^3 , we find that conditions 1 and 2 are satisfied. Consequently, there exists a transition from QS^3 to QS^4 . Transitions from QS^3 to the other candidate successor states are ruled out, because they violate condition 2. Because D^3 is not in the boundary of any domain, the second transition rule cannot be applied.

In figure 11(b), the transitions from the qualitative state $QS^4 = \langle D^4, \{(0, 0)\} \rangle$ to the states associated with the boundary domain $A(D^4) = \{D^9\}$ are investigated. The transition to $QS^9 = \langle D^9, \{\} \rangle$ is excluded, because the condition 2 of the first transition rule is not satisfied. In addition, we consider transitions to QS^4 from qualitative states associated with domains that have D^4 in their boundary. The second rule is valid for these cases. As can be verified in the figure, $D^4 \in A(D^3)$ and $D^4 \in A(D^5)$. D^3 and D^5 are regulatory domains, so the condition 1 is trivially satisfied. However, with x_b being a switching variable in D^4 , but not in D^3 and D^5 , condition 2 is satisfied in neither case. We therefore conclude that there are no transitions from QS^4 .

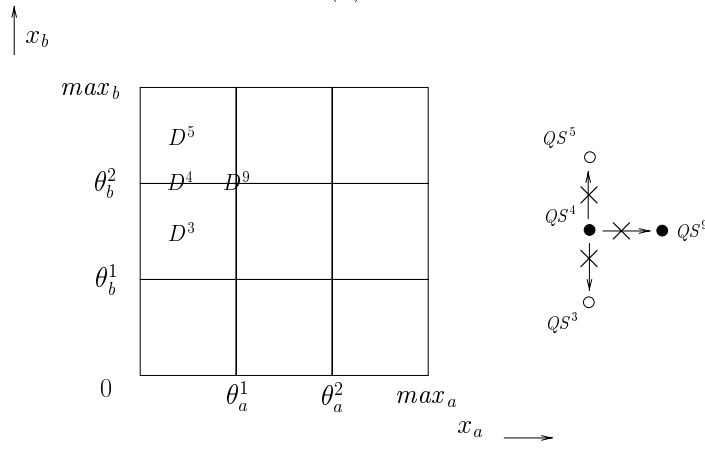
6.3 Simulation algorithm

The qualitative simulation algorithm can be summarized as follows. Given an initial domain D^0 , describing the initial protein concentrations \mathbf{x}_0 , the simulation algorithm computes the initial qualitative state QS^0 , and then determines all possible transitions from QS^0 to successor qualitative states by means of the rules above. The generation of successor states is repeated in a recursive manner until all qualitative states reachable from the initial qualitative state have been found.

The simulation algorithm is more precisely defined below. The *candidate* successor states of a qualitative state QS associated with a domain D are the qualitative states associated with domains D' , such that D' is in the boundary of D ($D' \in A(D)$) or D is in the boundary of D' ($D \in A(D')$). An *actual* successor state of QS is a candidate successor state QS' that satisfies the conditions specified in the transition rules.



(a)



(b)

Figure 11: Possible and impossible (\times) transitions from the qualitative states (a) QS^3 and (b) QS^4 , as determined by the transition rules.

```

determine qualitative state  $QS^0$  associated with  $D^0$ ;
push(stack,  $QS^0$ );
while not stack is empty do
  current qualitative state  $QS \leftarrow$  pop(stack);
  determine candidate successor states of  $QS$ ;
  for every candidate successor  $QS'$  do
    if  $QS'$  is actual successor state and not  $QS'$  reached before
    then mark  $QS'$  as reached;
    push(stack,  $QS'$ )
  end
end
end

```

The simulation algorithm generates a directed graph of qualitative states and transitions between qualitative states, a so-called *transition graph*. Let $QS = \langle D, V(D, \Psi(D)) \rangle$ be a qualitative state in the graph. If $\mathbf{0} \in V(D, \Psi(D))$, then QS is called a *qualitative equilibrium state* of the system. Qualitative equilibrium states may correspond to sinks of the graph, although this is not necessarily the case. A set

of qualitative states forming a cycle in the graph is called a *qualitative cycle* of the system. Since the number of possible qualitative states is finite, every path in the transition graph will reach a qualitative equilibrium state or a qualitative cycle at some point. Each path starting at the initial qualitative state QS^0 and ending at a qualitative equilibrium state or a qualitative cycle forms a possible qualitative behavior predicted by the simulation algorithm. The set of qualitative states included in any path leading to a qualitative equilibrium state or a qualitative cycle is called the *attraction set* of that state or cycle.

Figure 12(a) shows the transition graph for a qualitative simulation of the example system, starting in the regulatory domain D^1 , where both x_a and x_b lie below their first threshold. As can be seen, the simulation results in five qualitative behaviors leading to different qualitative equilibrium states. In QS^{16} , protein A is present at a high concentration ($x_a = \theta_a^2$), whereas protein B is present at a low concentration ($0 \leq x_b < \theta_b^1$). In QS^4 , protein A is present at a low concentration ($0 \leq x_a < \theta_a^1$) and protein B at a high concentration ($x_b = \theta_b^2$). In QS^7 , protein A and protein B are present at intermediate concentrations ($x_a = \theta_a^1$ and $x_b = \theta_b^1$). By simulating the regulatory system for every initial domain, and connecting the resulting transition graphs, the complete phase space behavior of the system can be analyzed. The results for the example system are shown in figure 12(b). They reveal that the system has three qualitative equilibrium states. The attraction set of the qualitative equilibrium state QS^4 is indicated in the figure.

The simulation results confirm the switch-like character of the network, established by earlier mathematical studies of genetic regulatory networks with the same or a similar structure (e.g., [6, 26, 31, 32, 62, 66]). The qualitative equilibrium states QS^4 and QS^{16} correspond to stable equilibria of the system, located at $(0, \theta_b^2)'$ and $(\theta_a^2, 0)'$, respectively. The qualitative equilibrium state QS^7 corresponds to an unstable equilibrium located at $(\theta_a^1, \theta_b^1)'$. This equilibrium lies on a separatrix dividing the initial regulatory domain D^0 into two parts. On one side of the separatrix, the trajectories move towards $(\theta_a^2, 0)'$, whereas on the other side of the separatrix the trajectories move towards $(0, \theta_b^2)'$. Only for the particular case that they start on the separatrix, will the trajectories end up in $(\theta_a^1, \theta_b^1)'$. The three cases are described by the qualitative behaviors (QS^1, QS^2, QS^3, QS^4) , $(QS^1, QS^6, QS^{11}, QS^{16})$, and (QS^1, QS^7) , respectively. The stable equilibria present the two functional states of the system: (1) gene a on and gene b off, (2) gene a off and gene b on.

The qualitative behaviors in a transition graph describe how protein concentrations evolve over time. In figure 13, one of the qualitative behaviors of the example system is explored in more detail. The figure shows how the bounds on the concentration of proteins A and B evolve over time, thus displaying the qualitative dynamics of the regulatory system.

The simulation method has been implemented in Java 1.3, in a program called *GNA (Genetic Network Analyzer)* [9].⁵ The program reads and parses input files specifying the model of the system (state equations, threshold and equilibrium inequalities) and the initial domain. From this information it produces a transition graph. GNA is accessible through a graphical user-interface, which allows the network of interactions between genes to be displayed, as well as the transition graph resulting from the simulation. In addition, the user can analyze the qualitative equilibrium states and qualitative cycles with their attraction sets, and focus on selected qualitative behaviors in order to study the temporal evolution of protein concentrations in more detail.

7 Properties of qualitative simulation

Given a qualitative PL model and an initial regulatory domain D^0 , what can be said about the correctness of the behaviors produced by qualitative simulation? We denote by X the set of solutions $\mathbf{x}(t)$ on $[0, \tau[$ of all quantitative PL models corresponding to the qualitative model, such that $\mathbf{x}(0) = \mathbf{x}_0 \in D^0$. In section 6, the aim of qualitative simulation has been described as finding the set of

⁵GNA is available for non-profit academic research purposes at <http://www-helix.inrialpes.fr/gna>.

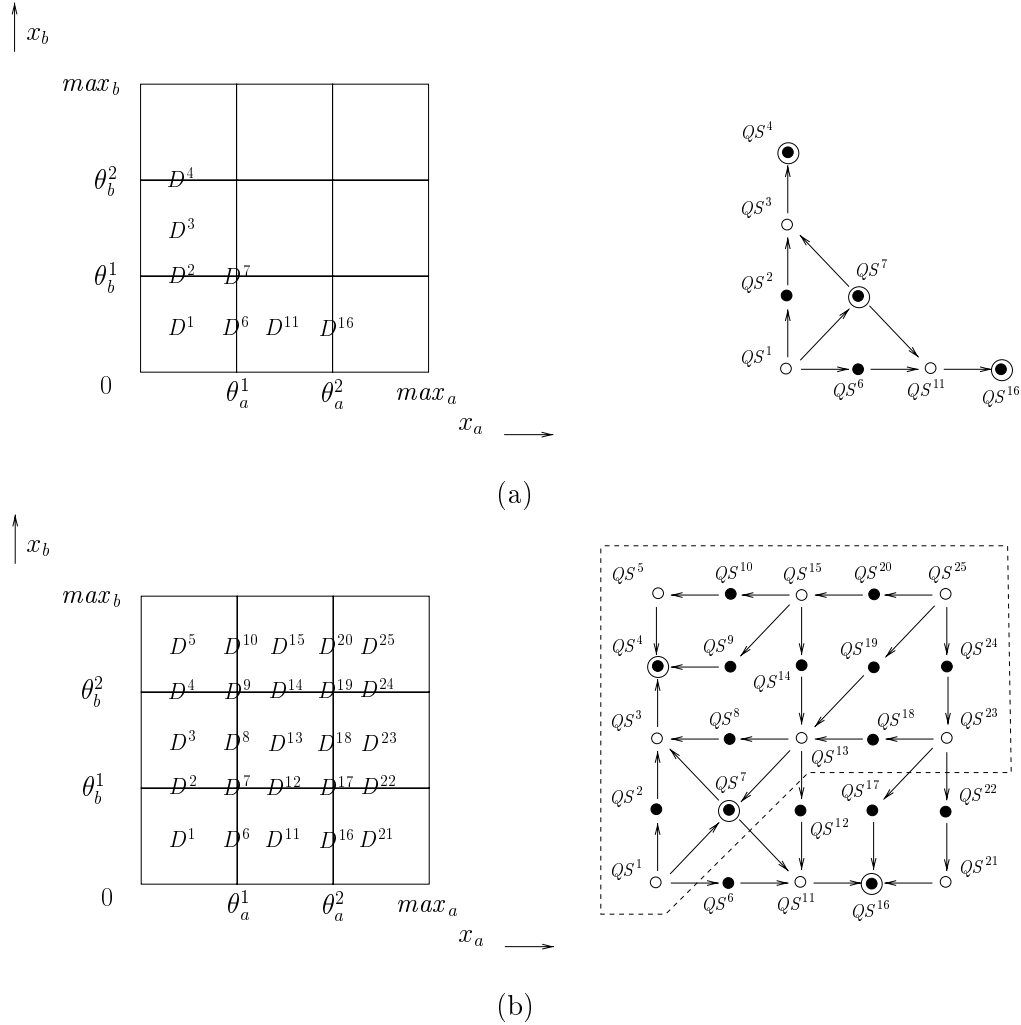


Figure 12: Phase spaces and transition graphs for the model describing the network in figure 9. Qualitative states associated with regulatory domains and switching domains are indicated by unfilled and filled dots, respectively. Qualitative equilibrium states are circled in addition. (a) Phase space and transition graph resulting from a simulation of the example system starting in the domain D^1 . (b) Phase space and transition graph showing all qualitative states of the system and their mutual transitions. The attraction set of the qualitative equilibrium state QS^4 is marked by the dotted border.

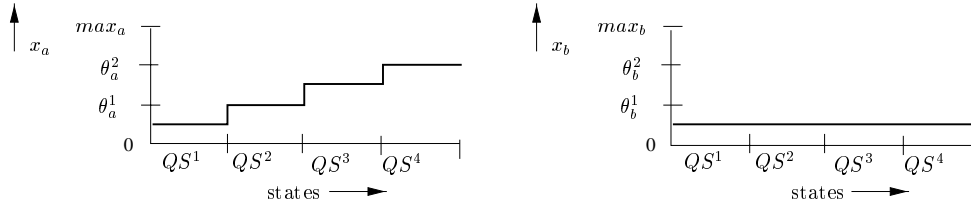


Figure 13: Detailed description of the qualitative behavior (QS^1, QS^2, QS^3, QS^4) in figure 12.

all qualitative behaviors that abstract from some $\mathbf{x} \in X$. More precisely, we would like that (1) for every $\mathbf{x} \in X$, the transition graph contains a qualitative behavior QB , such that $QB = QB(\mathbf{x}, 0, \tau)$ (*soundness*), and (2) for every qualitative behavior QB contained in the transition graph, there is some $\mathbf{x} \in X$, such that $QB = QB(\mathbf{x}, 0, \tau)$ (*completeness*).

7.1 Soundness

Theorem 1 The qualitative simulation algorithm is sound.

Proof. Let \mathbf{x} be a solution in X . On $[0, \tau[$, $\mathbf{x}(t)$ traverses a sequence of domains D^0, \dots, D^m , where $\mathbf{x}_0 \in D^0$. Like in the soundness proof of the qualitative simulation algorithm QSIM [35], we will show by induction that the qualitative behavior $QB(\mathbf{x}, 0, \tau)$ is generated by the algorithm.

The qualitative state $\langle D^0, V(D^0, \Psi(D^0)) \rangle$ is generated as described in section 6.1. Since $\Psi(D^0)$ includes $\Phi(D^0)$, $V(D^0, \Psi(D^0))$ is guaranteed to include $V(D^0, \Phi(D^0))$. A necessary condition for solution trajectories in D^0 to enter D^1 is either that D^1 is included in the boundary of D^0 , *i.e.* $D^1 \in A(D^0)$, or that D^0 is included in the boundary of D^1 , *i.e.* $D^0 \in A(D^1)$.

(1) If $D^1 \in A(D^0)$, then the transition from the qualitative state associated with D^0 to the qualitative state associated with D^1 is generated by means of transition rule 1. For, in order to reach the switching domain D^1 , \mathbf{x} must remain in D^0 for some time. That is, D^0 must be a regulatory domain or \mathbf{x} must be a sliding mode solution in D^0 . Moreover, the solution trajectory in D^0 must tend towards D^1 . Together with $V(D^0, \Psi(D^0)) \supseteq V(D^0, \Phi(D^0))$, this implies that the two conditions of transition rule 1 are satisfied.

(2) If $D^0 \in A(D^1)$, then the transition from the qualitative state associated with D^0 to the qualitative state associated with D^1 is generated by means of transition rule 2. For, in order to enter D^1 from D^0 , \mathbf{x} must remain in D^1 for some time. That is, D^1 must be a regulatory domain or \mathbf{x} must be a sliding mode solution in D^1 . Moreover, the solution trajectory in D^1 must point away from D^0 . As a consequence, with $V(D^0, \Psi(D^0)) \supseteq V(D^0, \Phi(D^0))$, it follows that the two conditions of transition rule 2 are satisfied.

We conclude that the transition from the qualitative state associated with D^0 to the qualitative state associated with D^1 is generated by the simulation algorithm. Now suppose that the qualitative state associated with the domain D^j , $0 < j < m$, has been generated. By an argument parallel to that given above, it can be shown that the qualitative state associated with the domain D^{j+1} will be generated by the algorithm as well. Consequently, the qualitative behavior $QB(\mathbf{x}, 0, \tau)$ is generated step by step. \square

The soundness of the qualitative simulation algorithm in section 6 has important consequences. It implies that, whatever the exact numerical values for the parameters may be, if these values are consistent with the threshold and equilibrium inequalities specified in the qualitative PL model, the solutions of the quantitative PL model are covered by the transition graph generated by the algorithm. More in particular, an equilibrium attained by some solution of a quantitative PL model consistent with the parameter inequalities necessarily corresponds to some qualitative equilibrium state in the transition graph. For, by proposition 2, we know that, if domain D contains an equilibrium, then $\mathbf{0} \in$

$V(D, \Phi(D))$. Soundness of the simulation algorithm guarantees that the qualitative state associated with D is actually generated, and that $\mathbf{0} \in V(D, \Psi(D))$.

7.2 Incompleteness

On the other hand, the qualitative simulation algorithm is not complete, as will be illustrated by means of a counterexample. The network in figure 14 consists of two genes, a and b , which encode the proteins A and B, respectively. The proteins form a heterodimer A·B repressing the expression of both genes. The qualitative PL model is shown in figure 15. It is assumed that the heterodimer represses the two genes above the same threshold concentration

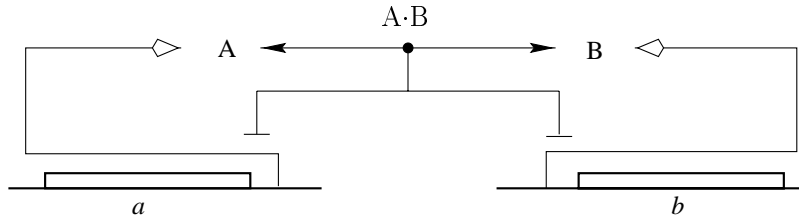


Figure 14: Example of a genetic regulatory network of two genes (a and b) coding for a regulatory protein (A and B). The regulatory proteins form a heterodimer A·B.

The phase space associated with the model consists of four regulatory domains and five switching domains (figure 16). When simulating the network from an initial state associated with the regulatory domain D^1 , in which both x_a and x_b are below their threshold concentration, we obtain the transition graph shown in the figure. The simulation gives rise to three different qualitative equilibrium states, while transitions between the qualitative equilibrium states are predicted as well.

State equation for gene a :

$$\dot{x}_a = \kappa_a (1 - s^+(x_a, \theta_a^1) s^+(x_b, \theta_b^1)) - \gamma_a x_a$$

Threshold inequalities: $0 < \theta_a^1 < \max x_a$

Equilibrium inequalities: $\theta_a^1 < \frac{\kappa_a}{\gamma_a} < \max x_a$

State equation for gene b :

$$\dot{x}_b = \kappa_b (1 - s^+(x_a, \theta_a^1) s^+(x_b, \theta_b^1)) - \gamma_b x_b$$

Threshold inequalities: $0 < \theta_b^1 < \max x_b$

Equilibrium inequalities: $\theta_b^1 < \frac{\kappa_b}{\gamma_b} < \max x_b$

Figure 15: State equations, threshold inequalities, and equilibrium inequalities for the proteins in the network of figure 14.

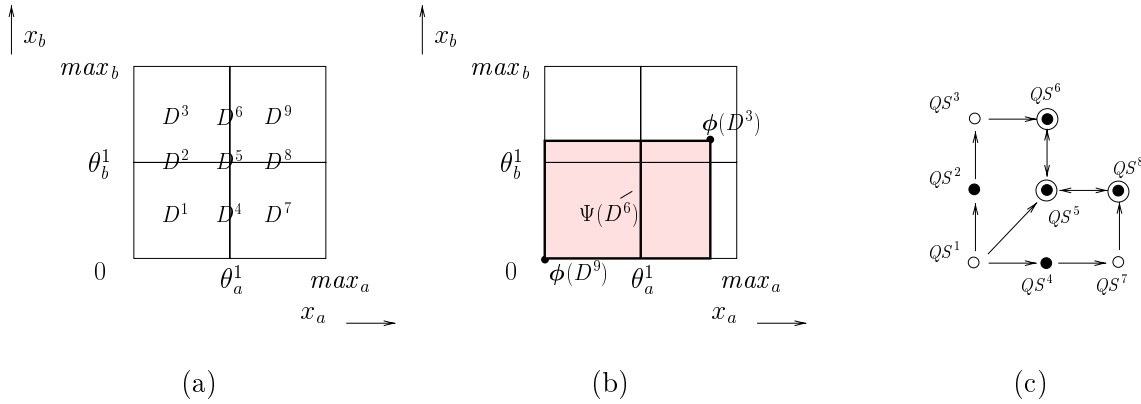


Figure 16: (a) Phase space box and transition graph for the genetic regulatory network described by the model in figure 15. (b) Determination of $\Psi(D^6)$. (c) Transition graph resulting from a simulation of the system starting in D^1 .

In order to explain this result, consider the computation of the qualitative state associated with the switching domain D^6 . $R(D^6) = \{D^3, D^9\}$, and $\phi(D^3) = (\kappa_a/\gamma_a, \kappa_b/\gamma_b)'$ and $\phi(D^9) = (0, 0)'$. Using the equilibrium inequalities in figure 15, we compute $V(D^6, \Psi(D^6))$ as described in section 6.1. We find $V(D^6, \Psi(D^6)) = \{(0, -1), (0, 0)\}$, where $\Psi(D^6)$ is the intersection of the smallest rectangle including $\phi(D^3)$ and $\phi(D^9)$, and the threshold boundary $x_a = \theta_a^1$ (figure 16(b)). QS^6 is a qualitative equilibrium state, because $(0, 0) \in V(D^6, \Psi(D^6))$. However, $(0, -1) \in V(D^6, \Psi(D^6))$ allows a transition to QS^5 , associated with $D^5 \in A(D^6)$, to occur as well. The transitions from QS^5 and from QS^8 are explained analogously.

$\Psi(D^6)$ overestimates $\Phi(D^6)$, as can be seen in figure 17. The smallest closed convex set $\overline{\text{co}}(\{\phi(D^3), \phi(D^9)\})$ is not $\overline{\text{rect}}(\{\phi(D^3), \phi(D^9)\})$, but rather the linear segment connecting the two target equilibria. As a consequence, $\Phi(D^6)$ is a single point in Ω , located at the intersection of the linear segment connecting $\phi(D^3)$ and $\phi(D^9)$, and the threshold boundary $x_a = \theta_a^1$. Depending on the exact position of $\phi(D^3)$ in D^9 , the linear segment crosses (a) D^6 , (b) D^5 , or (c) D^8 . In (a) we have $V(D^6, \Phi(D^6)) = \{(0, 0)\}$, in (b) $V(D^6, \Phi(D^6)) = \{(0, -1)\}$, and in (c) $V(D^6, \Phi(D^6)) = \{(0, -1)\}$. In other words, the parameter inequalities in figure 15 are consistent with two different qualitative states for QS^6 , both of which are covered by the qualitative state actually determined by the algorithm. Each configuration in figure 17(a)-(c) implies an additional constraint on the parameter values. In particular, we have

$$(a) \frac{\gamma_a \kappa_b}{\kappa_a \gamma_b} > \frac{\theta_b^1}{\theta_a^1}, \quad (b) \frac{\gamma_a \kappa_b}{\kappa_a \gamma_b} = \frac{\theta_b^1}{\theta_a^1}, \quad \text{and} \quad (c) \frac{\gamma_a \kappa_b}{\kappa_a \gamma_b} < \frac{\theta_b^1}{\theta_a^1}.$$

With the above constraints, we should have obtained the three different transition graphs shown in figure 17(d)-(f). Each of these is a subgraph of the transition graph actually generated by the simulation algorithm. Notice that the transition graphs in figure 17 contain a single qualitative equilibrium state, which is a sink of the graph.

The example illustrates that the transition graph resulting from a simulation may contain more qualitative behaviors than necessary to cover the trajectories corresponding to solutions in X . As a consequence, the algorithm is incomplete. Among other things this implies that, although we are sure to never miss a qualitative equilibrium state, not all qualitative equilibrium states in the transition graph need to correspond to some equilibrium of a quantitative PL model corresponding to the qualitative PL model. The incompleteness of the simulation algorithm raises issues for further research that will be addressed in the next section.

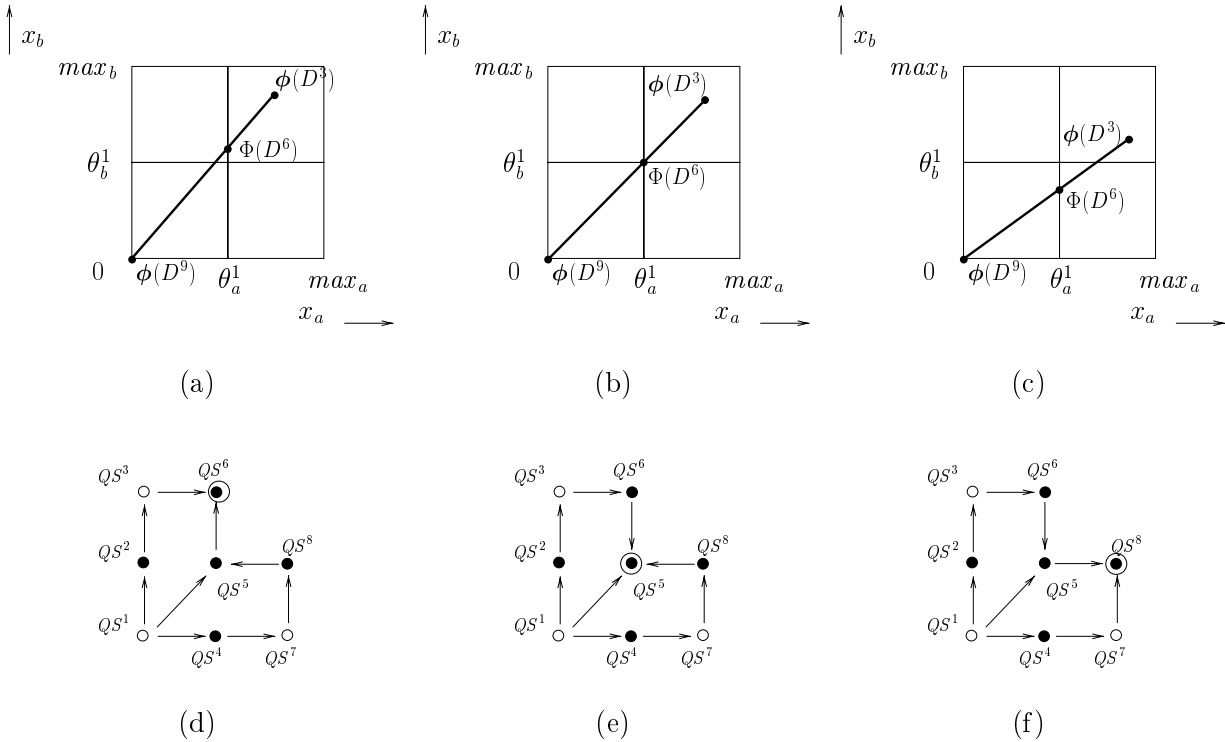


Figure 17: (a)-(c) Depending on the exact position of $\phi(D^3)$ in Ω , different values for $V(D^6, \Phi(D^6))$ will be found: (a) $V(D^6, \Phi(D^6)) = \{(0, 0)\}$, (b) $V(D^6, \Phi(D^6)) = \{(0, -1)\}$, and (c) $V(D^6, \Phi(D^6)) = \{(0, -1)\}$. The set $V(D^6, \Psi(D^6))$ actually calculated by the simulation algorithm, shown to be $\{(0, 0), (0, -1)\}$ in figure 16, is an overestimation of $V(D^6, \Phi(D^6))$. (d)-(f) The transition graphs that should have been generated for the cases (a)-(c), taking into account the additional constraints on the position of $\phi(D^3)$.

8 Discussion

We have presented a method for the qualitative simulation of genetic regulatory networks described by a class of piecewise-linear (PL) differential equations that has been well-studied in mathematical biology. The method allows the behavior emerging from large and complex networks of genetic regulatory interactions to be predicted in a qualitative manner. In the accompanying paper, we describe a model of the network underlying the initiation of sporulation in *B. subtilis*, and we compare predictions obtained through simulation with observations of the behavior of wild type and mutant bacteria [8]. The application of the qualitative simulation method is supported by a computer tool, called Genetic Network Analyzer (GNA) [9].

The PL models employed in this paper are based on step function approximations of the regulatory interactions involved in the synthesis and degradation of proteins. The step functions provide a succinct description of the regulatory logic, while abstracting from the details of molecular interactions. The biological validity of the step function expressions derives from experimental evidence that the activation of a gene, as a function of the concentration of a regulatory protein, often follows a steep sigmoidal curve [52, 67]. That is, below a certain threshold concentration of the protein, the gene will be hardly expressed at all, whereas above this threshold its expression rapidly saturates. Recent experimental studies have shown that some aspects of the qualitative dynamics of genetic regulatory networks synthesized *in vivo* correspond well with the predictions obtained from mathematical models based on switch-like approximations of regulatory interactions [4, 13, 17].

The use of step functions gives rise to discontinuities in the right-hand side of the differential equations, which may lead to nontrivial mathematical problems, as illustrated in figure 4. Several ways

to deal with the step function discontinuities have been proposed in the literature. The application of the PL models can be restricted to systems without autoregulation, which excludes situations of the type described in figure 4(b) [20]. Alternatively, when a trajectory arrives at a switching domain from which it cannot be continued, it may simply be stipulated to come to a dead stop [48]. Another solution, based on an idea of Plahte *et al.* [44, 48], consists of avoiding the discontinuities altogether, by replacing the step functions $s^+(x_j, \theta_j)$ by so-called *logoid* functions $l^+(x_j, \theta_j, \delta)$ that monotonically increase from 0 to 1 in a δ -interval around the threshold θ_j . The logoid functions approach their step function homologues as $\delta \rightarrow 0$.

Each of the above solutions is unsatisfactory in some way. In the first place, autoregulation is an ubiquitous feature in genetic regulatory networks [60]. (In fact, the network shown in figure 1 is a simplified version of the molecular switch determining the response of *E. coli* to phage λ infection [52].) Ignoring trajectories that cannot be continued in a switching domain will cause important behavioral properties of regulatory systems to be missed, like the singular steady states in the example. The use of logoid functions instead of step functions is attractive at first sight, but leads to nonlinear differential equation models that are difficult to treat in a qualitative way (see an earlier version of the simulation method in [10]). Here, we have adopted another solution, based on an approach to deal with differential equations with discontinuous right-hand sides originally proposed by Filippov [15]. This approach, recently applied to PL models of the form (2), has the advantage of putting no restrictions on the class of genetic regulatory networks that can be handled, while explicitly defining the behavior of the system in the threshold planes by means of simple-to-analyze PL models [27].

The simulation method described in this paper allows genetic regulatory networks to be simulated in the absence of numerical values for the kinetic parameters and molecular concentrations. The possible qualitative behaviors of the system are determined by means of qualitative constraints on the threshold and rate parameters. Whereas exact numerical values for the parameters are usually not available, the choice of appropriate threshold and equilibrium inequalities can be based on biological data, or is at least strongly constrained by the latter. If the choice of parameter inequalities is not unambiguously determined by the data, the consequences of opting for one combination of inequalities rather than another can be explored by simulating the system for each of the alternatives. Notice that in this case, instead of continuous ranges of numerical values, only a limited and often restricted number of alternative parameter inequalities need to be considered. The underdetermination of initial conditions by available biological data can be handled in the same way.

The predictions obtained through qualitative simulation have the form of a graph of qualitative states and transitions between qualitative states. Each qualitative state in the graph corresponds to a domain in the phase space where the system behaves in a qualitatively homogeneous manner, while a transition between two qualitative states corresponds to solution trajectories that leave one domain and enter another. Qualitative simulation makes predictions about the qualitative shape of solutions, illustrated in figure 13, as opposed to the quantitative predictions obtained through numerical simulation. The qualitative nature of the predictions is well-adapted to current measurement techniques in genomics, which have limited quantitative precision, but are able to detect qualitative changes in gene expression over time.

The soundness property of qualitative simulation guarantees that no solution of a quantitative PL model consistent with the qualitative PL model is omitted. In fact, to each such solution there corresponds a qualitative behavior in the transition graph. The soundness of qualitative simulation provides for a straightforward check of the robustness of simulation results to changes in parameter values. If a certain behavior is not observed for given parameter inequalities, one can be sure that it will not occur for any of the parameter values consistent with the threshold and equilibrium inequalities. That is, the behavioral variety of the system, given the constraints expressed in the parameter inequalities, is completely covered by the transition graph. This is a powerful result that may guide computationally-expensive simulations to test the sensitivity of the results to changes in parameter

values, as for example carried out in studies of the segment polarity network in *Drosophila* by von Dassow *et al.* [65] and simple metabolic pathways by Alves and Savageau [2, 3].

The lack of quantitative information on kinetic parameters and molecular concentrations has stimulated an interest in methods for modeling and simulation developed in the field of qualitative reasoning (QR), most notably QSIM [35] and QPT [16]. QR methods have been applied to the regulation of tryptophan synthesis [29] and λ phage growth [28] in *E. coli*, and to the regulation of the transcription factor families AP-1 and NF- κ B in different classes of animals [64]. A major problem with existing QR methods is their lack of upscalability, which causes the applicability of the methods to be limited to small regulatory systems of modest complexity.

As its application to the sporulation example in the accompanying paper shows, the qualitative simulation method presented here is able to deal with large and complex networks. Upscaling of the method has been achieved by using PL models that strongly constrain the local dynamics of the system. Moreover, the representation of qualitative states and the transition rules have been tailored to this class of models, in order to maximally exploit their favorable mathematical properties. Thus, in comparison with existing qualitative simulation methods like QSIM and QPT, the expressivity and generality of the modeling formalism have been traded for upscalability. This is particularly clear for the description of qualitative states, which is achieved on a higher level of abstraction in our method. Domains containing a hyperplane $x_i = \phi_i(D) = \mu_i^D / \nu_i^D$ give rise to a single qualitative state in our case, whereas in a method like QSIM several qualitative states must be defined, corresponding to the part of the domain located above, below, and inside the hyperplane [35].

Qualitative methods for the analysis of genetic regulatory systems have been developed in mathematical biology as well, the best-known example being Boolean networks [30, 58]. Simulation of Boolean networks rests on the assumption that a gene is either active or inactive, and that genes change their activation state synchronously. For the purpose of modeling actual genetic regulatory networks, these assumptions are usually too strong. Thomas and colleagues [62, 63] have proposed a generalized logical method that permits multivalued activation states and asynchronous transitions. On the formal level, the method of Thomas and colleagues is related to the approach presented in this paper. In fact, Snoussi has demonstrated that the logical equations can be interpreted as an abstraction of a special case of (1), where in the production term $f_i(\mathbf{x}) = \sum_{l \in L} \kappa_{il} b_{il}(\mathbf{x})$ it holds that either $b_{il}(\mathbf{x}) = s^+(x_j, \theta_j)$ or $b_{il}(\mathbf{x}) = s^-(x_j, \theta_j)$, while in the degradation term $g_i(\mathbf{x}) = \sum_{l \in L} \gamma_{il} b_{il}(\mathbf{x})$ it holds that $b_{il}(\mathbf{x}) = 1$ [56].

Although some ideas of the generalized logical method have been retained in the method presented here, in particular the parameter constraints of section 4, which are related to the logical parameters in [62, 63], we have opted for differential equation models. We believe that the latter formalism is intuitively clear and of large generality. In particular, it allows for a transparent description of the behavior of the system in the threshold planes. Although for the class of PL models covered by the generalized logical method, certain patterns of logical states can be interpreted as indicating singular steady states of the system [57], a general description of the behavior of the system in the threshold planes is currently missing. The differential equation formalism has the additional advantage of facilitating the integration of any quantitative data becoming available through improvements of current measurement techniques.

The simulation of genetic regulatory networks may lead to unwieldy transition graphs that are difficult to interpret. The availability of computer tools, allowing one to identify and to focus upon interesting parts of the transition graph, forms a prerequisite for the simulation of large and complex regulatory systems [9]. In addition, a weaker notion of soundness might be chosen. We call an algorithm *almost sound*, if for every $\mathbf{x} \in Y \subset X$, the transition graph contains a qualitative behavior QB , such that $QB = QB(\mathbf{x}, 0, \tau)$, while for every $\mathbf{x} \in X/Y$, the transition graph does not contain a qualitative behavior QB , such that $QB = QB(\mathbf{x}, 0, \tau)$. Moreover, the set X/Y is of measure 0 with respect to the set X . That is, the size of the set X/Y of trajectories not covered by a qualitative behavior in the transition graph is negligible in comparison with the size of the total set of trajectories X .

The simulation algorithm could be made almost sound by adapting the transition rules in section 6.2. For example, transitions in which the order of the domain of the qualitative states changes by more than one might be excluded. In fact, these transitions correspond to sets of trajectories lying on a separatrix in a domain, like in the transition from QS^1 to QS^7 in the example of figure 12. Although it still needs to be proven that the resulting algorithm is almost sound, there is some intuitive support for omitting these transitions. Biologically speaking, they represent the rare event that two or more protein concentrations reach a threshold at exactly the same moment [62]. Excluding these transitions would have the advantage of reducing the connectivity of the transition graph, and thus increasing their legibility, while retaining the transitions that are essential.

The simulation algorithm is sound but incomplete, which means that the transition graph may contain qualitative behaviors that do not correspond to any solution of a quantitative PL model satisfying the parameter constraints in the qualitative PL model. An illustration of this point was given in section 7.2, when discussing the results of the simulation of the genetic regulatory network in figure 14. The example reveals that the transition graph generated by the algorithm should be refined into three subgraphs, corresponding to alternative implicit additional constraints on the parameter values. This conclusion required graphical and algebraic considerations that are far beyond the scope of the current simulation algorithm. An issue for further research would be to exploit methods for the symbolic analysis of sets of algebraic inequalities, in order to obtain more precise estimates of the position of a target equilibrium set $\Phi(D)$ with respect to its domain D .

Proposition 2 expresses a relation between the existence of an equilibrium in a domain D and the qualitative state associated with D . Previous work on PL models of the form (2) has led to theorems that can be reformulated as stating a relation between the existence of a limit cycle in Ω and a certain type of qualitative cycle in the transition graph [24, 25, 45, 56]. Given the complexity of a general analysis of the occurrence of limit cycles for equations of the form (2), the relation is much weaker than its counterpart for equilibrium points. An interesting issue for further work would be to try to generalize the relation between limit cycles and qualitative cycles, using the literature on periodic solutions in differential equations with discontinuous right-hand sides (see [37] for a review).

The qualitative simulation method provides predictions of the possible qualitative behaviors of a genetic regulatory network. The interest of these predictions is that they can be directly compared with gene expression profiles obtained by means of quantitative RT-PCR or DNA microarrays. The use of predicted qualitative behaviors in combination with observed gene expression profiles allows hypothesized models of regulatory networks to be rapidly tested, even when only imprecise data is available. Along these lines, we are currently working on extensions of the method to validate and identify models of genetic regulatory networks using gene expression data. Incorporation of these extensions in the computer tool described in section 6.2 would allow the simulation method to evolve into a more general approach towards the computer-supported analysis of genetic regulatory networks.

Acknowledgments The authors would like to thank Grégory Batt for comments on previous versions of this report.

References

- [1] R. Alur, C. Belta, F. Ivančić, V. Kumar, M. Mintz, G.J. Pappas, H. Rubin, and J. Schlug. Hybrid modeling and simulation of biomolecular networks. In M.D. Di Benedetto and A. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control (HSCC 2001)*, volume 2034 of *Lecture Notes in Computer Science*, pages 19–32. Springer-Verlag, Berlin, 2001.
- [2] R. Alves and M.A. Savageau. Comparing systemic properties of ensembles of biological networks by graphical and statistical methods. *Bioinformatics*, 16(6):527–533, 2000.

- [3] R. Alves and M.A. Savageau. Systemic properties of ensembles of metabolic networks: Application of graphical and statistical methods to simple unbranched pathways. *Bioinformatics*, 16(6):534–547, 2000.
- [4] A. Becskei and L. Serrano. Engineering stability in gene networks by autoregulation. *Nature*, 405:590–591, 2000.
- [5] A. Ben-Hur and H.T. Siegelmann. Computation in gene networks. In M. Margenstern and Y. Rogozhin, editors, *Machines, Computations, and Universality: Third International Conference (MCU 2001)*, volume 2055 of *Lecture Notes in Computer Science*, pages 11–24. Springer-Verlag, Berlin, 2001.
- [6] J.L. Cherry and F.R. Adler. How to make a biological switch. *Journal of Theoretical Biology*, 203:117–133, 2000.
- [7] H. de Jong. Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 9(1):69–105, 2001.
- [8] H. de Jong, J. Geiselmann, G. Batt, C. Hernandez, and M. Page. Qualitative simulation of the initiation of sporulation in *B. subtilis*. in preparation, 2002.
- [9] H. de Jong, J. Geiselmann, C. Hernandez, and M. Page. Genetic Network Analyzer: A tool for the qualitative simulation of genetic regulatory networks. Technical Report RR-4262, INRIA Rhône-Alpes, Montbonnot Saint-Martin, 2001.
- [10] H. de Jong, M. Page, C. Hernandez, and J. Geiselmann. Qualitative simulation of genetic regulatory networks: Method and application. In B. Nebel, editor, *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, IJCAI-01*, pages 67–73, San Mateo, CA, 2001. Morgan Kaufmann.
- [11] R. Edwards and L. Glass. Combinatorial explosion in model gene networks. *Chaos*, 10(3):691–704, 2000.
- [12] R. Edwards, H.T. Siegelmann, K. Aziza, and L. Glass. Symbolic dynamics and computation in model gene networks. *Chaos*, 11(1):160–169, 2001.
- [13] M.B. Elowitz and S. Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403:335–338, 2000.
- [14] D. Endy and R. Brent. Modelling cellular behavior. *Nature*, 409:391–395, 2001.
- [15] A.F. Filippov. *Differential Equations with Discontinuous Righthand Sides*. Kluwer Academic Publishers, 1988.
- [16] K.D. Forbus. Qualitative process theory. *Artificial Intelligence*, 24:85–168, 1984.
- [17] T.S. Gardner, C.R. Cantor, and J.J. Collins. Construction of a genetic toggle switch in *Escherichia coli*. *Nature*, 403:339–342, 2000.
- [18] R. Ghosh and C.J. Tomlin. Lateral inhibition through Delta-Notch signaling: A piecewise affine hybrid model. In M.D. Di Benedetto and A. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control (HSCC 2001)*, volume 2034 of *Lecture Notes in Computer Science*, pages 232–246. Springer-Verlag, Berlin, 2001.
- [19] L. Glass. Classification of biological networks by their qualitative dynamics. *Journal of Theoretical Biology*, 54:85–107, 1975.

- [20] L. Glass. Combinatorial and topological methods in nonlinear chemical kinetics. *Journal of Chemical Physics*, 63(4):1325–1335, 1975.
- [21] L. Glass. Global analysis of nonlinear chemical kinetics. In B. Berne, editor, *Statistical Mechanics, Part B: Time Dependent Processes*, pages 311–349. Plenum Press, New York, 1977.
- [22] L. Glass and S.A. Kauffman. Co-operative components, spatial localization and oscillatory cellular dynamics. *Journal of Theoretical Biology*, 34:219–237, 1972.
- [23] L. Glass and S.A. Kauffman. The logical analysis of continuous non-linear biochemical control networks. *Journal of Theoretical Biology*, 39:103–129, 1973.
- [24] L. Glass and J.S. Pasternack. Prediction of limit cycles in mathematical models of biological oscillations. *Bulletin of Mathematical Biology*, 40:27–44, 1978.
- [25] L. Glass and J.S. Pasternack. Stable oscillations in mathematical models of biological control systems. *Journal of Mathematical Biology*, 6:207–223, 1978.
- [26] B.C. Goodwin. *Temporal Organization in Cells*. Academic Press, New York, N.Y., 1963.
- [27] J.-L. Gouzé and T. Sari. A class of piecewise linear differential equations arising in biological models. Technical Report RR-4207, INRIA Sophia-Antipolis, Sophia-Antipolis, 2001.
- [28] K.R. Heidtke and S. Schulze-Kremer. Design and implementation of a qualitative simulation model of λ phage infection. *Bioinformatics*, 14(1):81–91, 1998.
- [29] P.D. Karp. Design methods for scientific hypothesis formation and their application to molecular biology. *Machine Learning*, 12:89–116, 1993.
- [30] S.A. Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, New York, 1993.
- [31] A.D. Keller. Specifying epigenetic states with autoregulatory transcription factors. *Journal of Theoretical Biology*, 170:175–181, 1994.
- [32] A.D. Keller. Model genetic circuits encoding autoregulatory transcription factors. *Journal of Theoretical Biology*, 172:169–185, 1995.
- [33] J.L. Kelley. *General Topology*. Van Nostrand, New York, 1969.
- [34] K.W. Kohn. Molecular interaction maps as information organizers and simulation guides. *Chaos*, 11(1):1–14, 2001.
- [35] B. Kuipers. *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*. MIT Press, Cambridge, MA, 1994.
- [36] B.J. Kuipers. Qualitative reasoning: Modeling and simulation with incomplete knowledge. *Automatica*, 25(4):571–585, 1989.
- [37] R.I. Leine, D.H. van Campen, and B.L. van de Vrande. Bifurcations in nonlinear discontinuous systems. *Nonlinear Dynamics*, 23:105–164, 2000.
- [38] J.E. Lewis and L. Glass. Steady states, limit cycles, and chaos in models of complex biological networks. *International Journal of Bifurcation and Chaos*, 1(2):477–483, 1991.
- [39] J.E. Lewis and L. Glass. Nonlinear dynamics and symbolic dynamics of neural networks. *Neural Computation*, 4:621–642, 1992.

- [40] D.J. Lockhart and E.A. Winzeler. Genomics, gene expression and DNA arrays. *Nature*, 405:827–836, 2000.
- [41] H.H. McAdams and A. Arkin. Simulation of prokaryotic genetic circuits. *Annual Review of Biophysics and Biomolecular Structure*, 27:199–224, 1998.
- [42] H.H. McAdams and L. Shapiro. Circuit simulation of genetic networks. *Science*, 269:650–656, 1995.
- [43] L. Mendoza, D. Thieffry, and E.R. Alvarez-Buylla. Genetic control of flower morphogenesis in *Arabidopsis thaliana*: A logical analysis. *Bioinformatics*, 15(7-8):593–606, 1999.
- [44] T. Mestl, E. Plahte, and S.W. Omholt. A mathematical framework for describing and analysing gene regulatory networks. *Journal of Theoretical Biology*, 176:291–300, 1995.
- [45] T. Mestl, E. Plahte, and S.W. Omholt. Periodic solutions in systems of piecewise-linear differential equations. *Dynamics and Stability of Systems*, 10(2):179–193, 1995.
- [46] S.W. Omholt, X. Kefang, Ø. Andersen, and E. Plahte. Description and analysis of switchlike regulatory networks exemplified by a model of cellular iron homeostasis. *Journal of Theoretical Biology*, 195:339–350, 1998.
- [47] A. Pandey and M. Mann. Proteomics to study genes and genomes. *Nature*, 405:837–846, 2000.
- [48] E. Plahte, T. Mestl, and S.W. Omholt. Global analysis of steady points for systems of differential equations with sigmoid interactions. *Dynamics and Stability of Systems*, 9(4):275–291, 1994.
- [49] E. Plahte, T. Mestl, and S.W. Omholt. Stationary states in food web models with threshold relationships. *Journal of Biological Systems*, 3(2):569–577, 1995.
- [50] E. Plahte, T. Mestl, and S.W. Omholt. A methodological basis for description and analysis of systems with complex switch-like interactions. *Journal of Mathematical Biology*, 36:321–348, 1998.
- [51] E.I. Prokudina, R.Y. Valeev, and R.N. Tchuraev. A new method for the analysis of the dynamics of the molecular genetic control systems. II. Application of the method of generalized threshold models in the investigation of concrete genetic systems. *Journal of Theoretical Biology*, 151:89–110, 1991.
- [52] M. Ptashne. *A Genetic Switch: Phage λ and Higher Organisms*. Cell Press & Blackwell Science, Cambridge, MA, 2nd edition, 1992.
- [53] B. Ren, F. Robert, J.J. Wyrick, O. Aparicio, E.G. Jennings I. Simon, J. Zeitlinger, J. Schreiber, N. Hannett, E. Kanin, T.L. Volkert, C.J. Wilson, S.P. Bell, and R.A. Young. Genome-wide location and function of DNA binding proteins. *Science*, 290:2306–2309, 2000.
- [54] L. Sánchez and D. Thieffry. A logical analysis of the *Drosophila* gap genes. *Journal of Theoretical Biology*, 211:115–141, 2001.
- [55] M. Santillán and M.C. Mackey. Dynamic regulation of the tryptophan operon: A modeling study and comparison with experimental data. *Proceedings of the National Academy of Sciences of the USA*, 98(4):1364–1369, 2001.
- [56] E.H. Snoussi. Qualitative dynamics of piecewise-linear differential equations: A discrete mapping approach. *Dynamics and Stability of Systems*, 4(3-4):189–207, 1989.
- [57] E.H. Snoussi and R. Thomas. Logical identification of all steady states: The concept of feedback loop characteristic states. *Bulletin of Mathematical Biology*, 55(5):973–991, 1993.

- [58] R. Somogyi and C.A. Sniegoski. Modeling the complexity of genetic networks: Understanding multigenic and pleiotropic regulation. *Complexity*, 1(6):45–63, 1996.
- [59] R.N. Tchuraev and V.A. Ratner. A continuous approach with threshold characteristics for simulation of gene expression. In K. Bellmann, editor, *Molecular Genetic Information Systems: Modelling and Simulation*, pages 64–80. Akademie-Verlag, Berlin, 1983.
- [60] D. Thieffry, A.M. Huerta, E. Pérez-Rueda, and J. Collado-Vides. From specific gene regulation to genomic networks: A global analysis of transcriptional regulation in *Escherichia coli*. *BioEssays*, 20:433–440, 1998.
- [61] D. Thieffry and R. Thomas. Dynamical behaviour of biological networks: II. Immunity control in bacteriophage lambda. *Bulletin of Mathematical Biology*, 57(2):277–297, 1995.
- [62] R. Thomas and R. d’Ari. *Biological Feedback*. CRC Press, Boca Raton, FL, 1990.
- [63] R. Thomas, D. Thieffry, and M. Kaufman. Dynamical behaviour of biological regulatory networks: I. Biological role of feedback loops and practical use of the concept of the loop-characteristic state. *Bulletin of Mathematical Biology*, 57(2):247–276, 1995.
- [64] R.B. Trelease, R.A. Henderson, and J.B. Park. A qualitative process system for modeling NF- κ B and AP-1 gene regulation in immune cell biology research. *Artificial Intelligence in Medicine*, 17:303–321, 1999.
- [65] G. von Dassow, E. Meir, E.M. Munro, and G.M. Odell. The segment polarity network is a robust developmental module. *Nature*, 406:188–192, 2000.
- [66] D.M. Wolf and F.H. Eeckman. On the relationship between genomic regulatory element organization and gene regulatory dynamics. *Journal of Theoretical Biology*, 195:167–186, 1998.
- [67] G. Yagil and E. Yagil. On the relation between effector concentration and the rate of induced enzyme synthesis. *Biophysical Journal*, 11:11–27, 1971.
- [68] H. Zhu and M. Snyder. Protein arrays and microarrays. *Current Opinion in Chemical Biology*, 5:40–45, 2001.



Unité de recherche INRIA Rhône-Alpes

655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique

615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur

INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)

<http://www.inria.fr>

ISSN 0249-6399