Parametric Analysis and Abstraction of Genetic Regulatory Networks

Michaël Adélaïde and Grégoire Sutre

LaBRI, UMR CNRS 5800
Domaine Universitaire, 351, cours de la Libération, Talence, France

{adelaide, sutre}@labri.fr

Abstract

This paper investigates the application of symbolic model checking techniques to the behavioral analysis of genetic regulatory networks. We use a natural description of regulatory networks in terms of hypergraphs with parameters (thresholds, production rates and a degradation rate). Their continuous semantics is given by so-called parametric attractor systems, a time-abstraction of the usual piecewise linear differential semantics. Parametric attractor systems turn out to be an interesting new class of hybrid systems. We show how to perform symbolic analysis of these systems using parametric linear constraints. This "symbolic model checking technology" has two main advantages over previous methods. First, it allows us perform symbolic reachability computations, e.g. compute the set of parameter values ensuring that the system satisfies a given bounded reachability property. Secondly, arbitrarily precise abstractions can be automatically computed (the coarsest one being the usual "discrete" abstraction), without any prior assumption on parameters, as the relevant parameter constraints are automatically inferred.

Key words: regulatory networks, parametric analysis, abstraction, refinement, parametric attractor systems.

1 Introduction

Computer scientists and biologists work together to model and analyse systems arising in living organisms. Actually, those systems have complex behaviors. For instance, in *biological regulatory networks*, the interactions between proteins combine discrete and continuous aspects, along with real constants. As those real constants are not known, they must be seen as parameters of the systems.

In this paper, we apply formal methods usually used for the verification of critical systems to perform a parametric analysis of biological regulatory networks.

This is a preliminary version. The final version will be published in Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

Final (pre–edited) version of the paper to appear in:
Proc. 2nd Workshop on Concurrent Models in Molecular Biology (BioCONCUR'04), London, UK, Aug. 2004,
Electronic Notes in Theor. Comp. Sci. Elsevier, 2004.

1.1 Modeling of regulatory networks

In a simplified view of cell biology, the DNA of a cell is divided into genes and binding sites. A gene encodes a protein, and proteins or complexes (which are composed of two or more proteins) can bind to binding sites. A gene is associated to one or many binding sites. The rate of transcription of DNA to RNA for a given gene depends on the states of the binding sites associated to this gene. This mechanism is called regulation of the transcription. The fluxes of information are not in a straight line from DNA to proteins. There is feedback from proteins to genes.

This approach mixes boolean dynamics — the dynamics of a binding site: a complex is attached to it or not — and continuous dynamics — the dynamics of the proteins concentrations. Such systems are called regulatory networks.

Most often, genetic regulatory networks provide qualitative information on their components. For instance, the regulatory diagram shown in Figure 1, and borrowed from [dJ02, page 81], expresses that: gene (i) produces protein (i) (for $i \in \{1, 2, 3\}$), protein (1) inhibits (symbol $-$) gene (3) while protein (2) activates (symbol $+$) gene (1), proteins (1) and (3) form a complex which inhibits gene (2).

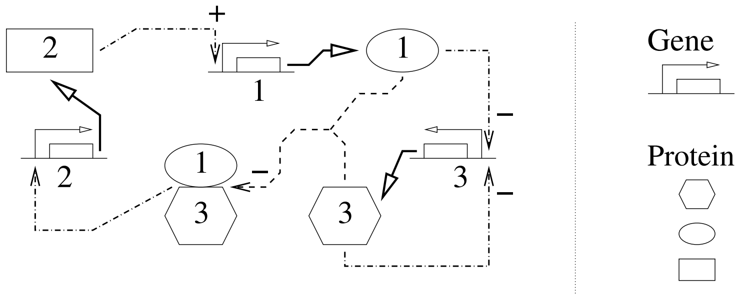

Figure 1. Introductory example

The continuous dynamics of a regulatory diagram is given by a differential system which defines the evolutions of protein concentrations. The differential system of the example given in Figure 1 is the following, where x_i denotes the concentration of protein (i), and where $s^+, s^- : \mathbb{R} \times \mathbb{R} \to \{0, 1\}$ are the usual *partial step functions* of Heaviside:

$$\begin{aligned}
\tfrac{d}{dt}x_1 &= \kappa_1 s^+(x_2, \theta_{1,2}) - \gamma_1 x_1 \\
\tfrac{d}{dt}x_2 &= \kappa_2 s^-(x_1, \theta_{2,1}) s^-(x_3, \theta_{2,3}) - \gamma_2 x_2 \\
\tfrac{d}{dt}x_3 &= \kappa_3 s^-(x_1, \theta_{3,1}) + \kappa_4 s^-(x_3, \theta_{4,3}) - \gamma_3 x_3
\end{aligned}$$

$$s^+(x, \theta) = \begin{cases} 0 \text{ if } x < \theta \\ 1 \text{ if } x > \theta \end{cases} \qquad s^- = 1 - s^+$$

This differential system considers production rates (the κ's), degradation rates (the γ's), and thresholds (the θ's) to indicate activation and inhibition; it is said to be a piecewise linear system because whenever no variable x_i is on a threshold $\theta_{i,j}$ — we say that the system is in a *box* — it gives a usual linear differential equation. The model is parametric since most constants are not known.

The step functions can be replaced by logoids functions [MPO95]. The logoids are piecewise affine continuous functions. Then, a step function is the simple limit of a sequence of logoids functions. Hill functions [dJ02,MPO95,Tho98] play the same role as logoids and step functions. All of them are used to model changes of mode which are discontinuous if we consider step functions, continuous but not differentiable with logoids, and infinitely differentiable with Hill functions. We only consider step functions in the rest of the paper.

1.2 Analysis of regulatory networks

With the huge amount of genetic data now available, models for regulatory networks are becoming very large. Predicting their behaviors is usually challenging, as these networks may contain subtle interactions that are very hard to grasp. Thus, some form of mechanical assistance is greatly desired. We are mainly interested in behavioral properties of these networks:

- *stable states*: find stationary points and their attraction basin,
- *reachability problem*: determine which regions (in the concentration space) are reachable from a given region,
- *reverse engineering*: synthesize the parameters ensuring that the model satisfies a given behavioral property (usually simply a trace corresponding to an experiment).

Most previous approaches are based on symbolic computations or on discrete approximations of the continuous dynamics. The original work has been done by [TD90,TCT93,Tho91]. We review in more details related works in subsection 1.4.

1.3 Our contributions

Our approach is based on symbolic computation techniques [HM00] and abstraction interpretation techniques [CC79,LGS+95]. We apply these techniques to the so-called class of *parametric attractor systems* : a new class of hybrid systems arising from regulatory networks represented as hypergraphs. Parametric attractor systems are close to the parametric piecewise linear systems of [dJG+04,GS03]. In particular, we also handle thresholds in a rigorous way. The only restriction we do is that we assume a uniform degradation rate. Our main contributions are the following:

(i) we propose a global framework from regulatory networks given as labeled hypergraphs to parametric hybrid systems (with symbolic post/pre computations expressed in real arithmetic), and to predicate abstractions : the relevant relationships between parameters (given on the hypergraph) are automatically synthesized and we do not require any prior enumeration as it is done currently;

(ii) this gives us procedures to decide : bounded reachability (reachability in a bounded number of steps), bounded inevitability, attraction basins ;

(iii) our framework allows us to derive (arbitrarily precise) abstractions that are finer than the usual "discrete" abstraction.

(iv) we are able to synthesize constraints on parameters which could allow reverse engineering.

1.4 Related work

In order to analyse parametric systems, one usually first builds a partition of the parameter space, and then an abstraction is computed for each cell (of the partition). In [TD90], the parametric regulatory network is simplified into a parametric logical circuit: parameters and concentrations take qualitative values in finite domains. This model has been used to isolate regions that contain stationary points. The main difficulty concerns thresholds: for simplicity, the authors claimed that two thresholds cannot be crossed simultaneously (and that concentrations can not stay on a threshold).

This difficulty is overcome in [dJG+04], where concentrations and parameters take real values. Dynamics on thresholds is rigorously defined by differential inclusions. They show how to build finite abstractions of the continuous dynamics. However, they cannot refine their abstractions (as the partition on parameters is fixed), and they cannot perform bounded reachability. Our work is based and improves on [dJG+04].

The algorithm presented in [GTT03] builds abstract regions, using symbolic predecessor computations, and refines them until a deterministic abstraction is obtained. This technique is based on Lie derivatives: roughly speaking, the Lie derivative of a scalar field $V : \mathbb{R}^n \to \mathbb{R}$ with respect to a (smooth) vector field $f : \mathbb{R}^n \to \mathbb{R}^n$ gives the variations of V along a trajectory defined by $\frac{d}{dt}\overrightarrow{x} = f(\overrightarrow{x})$. Lie derivatives can be used to define more and more precise abstractions [TK02], however these techniques are not suited for bounded reachability.

Our approach is similar to the one used in [GTT03]. Theoretical computability issues may appear in the analysis done by [GTT03]. They do not have our uniform degradation restriction, but concentrations cannot stay on thresholds.

The class of parametric attractor systems that we study in this paper is incomparable with the usual class of rectangular hybrid automata [Hen96], but it is somewhat closer to the class of dynamical systems with piecewise-constant derivatives [AMP95], which could be seen as attractor systems whose attractors are "infinitely far".

1.5 Outline

The paper is organized as follows. In section 2, we formally present regulatory networks. In section 3, the semantics of regulatory networks is given by parametric attractor systems. In section 4, we show how to perform symbolic analysis and abstractions. We illustrate our approach in section 5 with a small case study, and we end the paper in section 6 with some conclusions.

2 Regulatory networks

We first present some basic definitions and notations that will be used in the paper. We denote by $\mathbb{R}$ (resp. $\mathbb{R}_{\geq 0}$, $\mathbb{R}_{>0}$, $\mathbb{Q}$) the set of *real numbers* (resp. *nonnegative real numbers*, *positive real numbers*, *rational numbers*). We write $\mathbb{P}(S)$ (resp. $\mathbb{P}_f(S)$) for the *set of subsets* (resp. *set of finite subsets*) of any set S. Given any finite set I, we denote by S^I the set of (I-indexed) *vectors* of elements of S. Observe that $\mathbb{R}^I$ is a vector space. The *convex hull* of any subset $S \subseteq \mathbb{R}^I$, written $hull(S)$, is the smallest convex subset of $\mathbb{R}^I$ containing S. For any two real vectors $\overrightarrow{x}$ and $\overrightarrow{y}$ in $\mathbb{R}^I$, we denote by $\overrightarrow{x} \cdot \overrightarrow{y}$ the componentwise product of $\overrightarrow{x}$ and $\overrightarrow{y}$.

2.1 Syntax in terms of directed hypergraphs

A *directed hypergraph* is a 4-tuple $\mathcal{H} = \langle V, E, in, out \rangle$ where V is a finite set of *vertices*, E is a finite set of *hyperarcs*, and $in : E \to \mathbb{P}(V)$ and $out : E \to V$ are *source* and *target* mappings. Observe that hyperarcs may have several source nodes, but have only one target node.

Definition 2.1 A *regulatory network* is an 8-tuple $\mathcal{N} = \langle V, E, in, out, \gamma, \kappa, \theta, \varepsilon \rangle$ where:

- $\langle V, E, in, out \rangle$ is a directed hypergraph,
- $\gamma : V \to \mathbb{R}_{>0}$ is a *degradation* mapping,
- $\kappa : E \to \mathbb{R}_{\geq 0}$ is a *production* mapping, and
- $\theta : E \times V \to \mathbb{R}_{\geq 0}$ and $\varepsilon : E \times V \to \{-, +\}$ respectively are *threshold* and *sign* partial mappings defined on $\{(e, v) \in E \times V \mid v \in in(e)\}$.

The *state set* of a regulatory network $\mathcal{N} = \langle V, E, in, out, \gamma, \kappa, \theta, \varepsilon \rangle$ is the subset $\mathbb{R}^V_{\geq 0}$ of the vector space $\mathbb{R}^V$. Thus, a state $\overrightarrow{x} \in \mathbb{R}^V_{\geq 0}$ associates a nonnegative real value $\overrightarrow{x}(v)$ (e.g. a concentration) to each vertex v. States continuously change values according to the regulatory hyperarcs. Formally, the *dynamics* of the regulatory network $\mathcal{N}$ is given for each $v \in V$ by the following system of differential equations [dJ02]:

$$\frac{d}{dt}\overrightarrow{x}(v) \quad = \quad \sum_{e \in E \,\mid\, v = out(e)} \kappa(e) \prod_{w \in in(e)} s^{\varepsilon(e,w)}(\overrightarrow{x}(w), \theta(e, w)) \quad - \quad \gamma(v)\overrightarrow{x}(v)$$

2.2 Dynamics in terms of differential inclusions

Recall that partial step functions s^+ and s^- are undefined on all pairs (x, x) with $x \in \mathbb{R}$. Therefore, the above differential system is not defined at the thresholds. Thus, the state set $\mathbb{R}^V_{\geq 0}$ is divided into boxes and walls:

- a *box* is any Cartesian product of intervals I_v, where for every $v \in V$, the interval I_v is either an open interval between two consecutive thresholds (for v) or a left-closed interval between zero and the smallest threshold (for v).
- a *wall* is any non empty intersection of closures of boxes, and we denote by $\mathcal{B}(W)$ the set of adjacent boxes of any wall W.

Following [dJG+04], the dynamics within each box B can be rewritten as the linear differential system [1] : $\frac{d}{dt}\overrightarrow{x} = \overrightarrow{\kappa_B} - \overrightarrow{\gamma} \cdot \overrightarrow{x}$, where the vector $\overrightarrow{\kappa_B} \in \mathbb{R}^V_{\geq 0}$ is directly obtained from the previous differential equations. Then, there is a unique trajectory with origin $\overrightarrow{x_0}$ in $\mathbb{R}^V_{\geq 0}$, which is defined (for every $v \in V$) by:

$$\overrightarrow{x}(v)(t) \;=\; \left(\overrightarrow{x_0}(v) - \frac{\overrightarrow{\kappa_B}(v)}{\overrightarrow{\gamma}(v)}\right) e^{-\overrightarrow{\gamma}(v)t} \;+\; \frac{\overrightarrow{\kappa_B}(v)}{\overrightarrow{\gamma}(v)}$$

Hence the trajectory tends to the limit point $\overrightarrow{\varphi_B}$ with $\overrightarrow{\varphi_B}(v) = \frac{\overrightarrow{\kappa_B}(v)}{\overrightarrow{\gamma}(v)}$. This limit point, also called *image point*, can be outside of the box. When this is the case, there exists a time t at which the trajectory leaves the box and enters a wall.

Recall that a wall W is the intersection of closures of adjacent boxes, i.e. $W = \bigcap_{B \in \mathcal{B}(W)} \overline{B}$. In W, following [GS03], the dynamics is given by a differential inclusion:

$$\frac{d}{dt}\overrightarrow{x} \in hull(\{\overrightarrow{\kappa_B} - \overrightarrow{\gamma} \cdot \overrightarrow{x} \mid B \in \mathcal{B}(W)\})$$

Consequently, in both boxes and walls, the dynamics of the system is given by a differential inclusion:

$$\frac{d}{dt}\overrightarrow{x} \in H(\overrightarrow{x})$$

where $H(\overrightarrow{x})$ is the convex hull of a finite set of vectors (depending on $\overrightarrow{x}$).

Trajectories of such systems in the sense of Filippov are absolutely continuous functions $\overrightarrow{\chi} : [0, \tau] \to \mathbb{R}^V$ such that the set $\{v \mid \frac{d\overrightarrow{\chi}}{dt}(v) \notin H(\overrightarrow{\chi}(v))\}$ is of Lebesgue measure zero [dJG+04].

[1] Note that the degradation mapping $\gamma : V \to \mathbb{R}_{>0}$ can be seen as a real vector $\overrightarrow{\gamma}$ in $\mathbb{R}^V_{>0}$.

3 Parametric Attractor System

We present in this section the class of parametric attractor systems, a new class of hybrid systems that arises from regulatory networks where the degradation rates are equal.

3.1 Syntax

Given a set X of variables, we denote by $\mathbb{L}(X)$ the set of *linear expressions* over X with coefficients in $\mathbb{Q}$. A *linear constraint* is any boolean combination of constraints of the form $e \# e'$ where $e, e' \in \mathbb{L}(X)$ and $\# \in \{<, \leq, =, \geq, >\}$. For any set X, we denote by $\mathbb{P}(X)$ (resp $\mathbb{P}_f(X)$) the set of all subsets (resp. of all finite subsets) of X.

Let V and T denote two finite sets of variables ranging over $\mathbb{R}_{\geq 0}$. Variables in V are called *components* and variables in T are called *threshold parameters* (shortly *thresholds*). Thresholds are related to components through a *threshold parameter mapping* $th : V \to \mathbb{P}(T)$. We require that all thresholds for the same variable have distinct values. Formally, a *threshold parameter valuation* is any mapping θ in $\mathbb{R}^T_{\geq 0}$ such that θ is injective on $th(v)$ for all $v \in V$.

Given a component $v \in V$, a *location* of v is any partial mapping l from T to $\{-, 0, +\}$ defined on $th(v)$ and such that $l^{-1}(\{0\})$ is either empty or a singleton (i.e. no more than one threshold can be mapped to 0). Roughly speaking, $l(t)$ give the position of v with respect to t. Formally, we associate to each location l the linear constraint $C(l)$ defined by:

$$C(l) \quad = \quad \bigwedge_{t \in l^{-1}(\{+\})} v - t \geq 0 \quad \wedge \quad \bigwedge_{t \in l^{-1}(\{0\})} v - t = 0 \quad \wedge \quad \bigwedge_{t \in l^{-1}(\{-\})} v - t \leq 0$$

Given a threshold valuation θ, we denote by $[\![l]\!]_\theta$ the subset of $\mathbb{R}_{\geq 0}$ satisfying the constraint $C(l)$.

A *zone* is any mapping z from V to $\{-, 0, +\}^T$ such that $z(v)$ is a location of v for every $v \in V$. We denote by $Z(V, T)$ the set of all zones. Intuitively, a zone gives the position of each component with respect to their thresholds. Formally, the meaning of a zone z is given by its associated linear constraint $C(z)$ defined by: $C(z) = \bigwedge_{v \in V} C(z(v))$. Given a threshold valuation θ, we denote by $[\![z]\!]_\theta$ the subset of $\mathbb{R}^V_{\geq 0}$ satisfying the constraint $C(z)$. Observe that $[\![z]\!]_\theta$ is equal to $\{ \overrightarrow{x} \in \mathbb{R}^V_{\geq 0} \mid \forall v \in V, \overrightarrow{x}(v) \in [\![z(v)]\!]\}$.

Definition 3.1 A *parametric attractor system* is a 5-tuple $\mathcal{A} = \langle V, T, th, K, at \rangle$ where:

- V is a finite set of *components*,
- T is a finite set of *threshold parameters*,
- $th : V \to \mathbb{P}(T)$ is a *threshold mapping*,
- K is a finite set of *attractor parameters*, and

- $at : Z(V,T) \to \mathbb{P}_f(\mathbb{L}(K)^V)$ is an *attractor mapping*.

3.2 Semantics

We now give the operational semantics of a parametric attractor system in terms of a labeled transition system. Recall that a labeled transition system is a triple $\mathcal{S} = (Q, \Sigma, \to)$ where Q is a set of *configurations*, Σ is an alphabet of *labels*, and $\to \subseteq Q \times \Sigma \times Q$ is a *labeled transition relation*.

Intuitively, the system has two kinds of transitions: continuous steps and discrete steps. In a continuous step, all components move simultaneously in a straight line towards the attractors of the current zone, without leaving the zone. We will label this continuous step by the containing zone. Formally, the set of *attraction labels* is the set $\Delta(V,T)$ defined by: $\Delta(V,T) = \{\delta(z) \mid z \in Z(V,T)\}$. Here the $\delta(z)$ are symbols which mean "motion inside zone z".

In a discrete step, each component may either (1) stay in its current location, or (2) move to an "adjacent" location. We will label the component's move with $\perp$ in the first case, and with the threshold parameter that was crossed over in the second case. Formally, the set of *component cross labels* is the set $\Lambda(T)$ defined by:

$$\Lambda(T) \quad = \quad \{\perp\} \bigcup_{t \in T} \{- \xrightarrow{t} +,\ - \xrightarrow{t} 0,\ + \xrightarrow{t} -,\ + \xrightarrow{t} 0,\ 0 \xrightarrow{t} +,\ 0 \xrightarrow{t} -\}$$

Naturally, given a component $v \in V$, moves with respect to a threshold $t \in T$ with $t \notin th(v)$ will not be considered. A *cross label* is any vector σ in $\Lambda(T)^V$ such that for every $v \in V$, $\sigma(v)$ is either $\perp$ or a cross label with respect to a threshold in $th(v)$. The set of all cross labels is denoted by $\Lambda(V,T)$.

We first define transitions between zones, as a first step towards an operational semantics for parametric attractor systems. The *zone transition relation* $\rightsquigarrow_{V,T} \subseteq Z(V,T) \times \Lambda(V,T) \times Z(V,T)$ is defined by: $z \overset{\sigma}{\rightsquigarrow}_{V,T} z'$ if for all v in V we have:

(i) either $\sigma(v) = \perp$ and the locations $z(v)$ and $z'(v)$ are equal,

(ii) or $\sigma(v) = \varepsilon \xrightarrow{t} \varepsilon'$ and the locations $z(v)$ and $z'(v)$ are equal except on t, where $z(v)(t) = \varepsilon$ and $z'(v)(t) = \varepsilon'$.

The transition relation $\rightsquigarrow_{V,T}$ on zones can be seen as the discrete control flow semantics of parametric attractor systems.

Recall that a threshold parameter valuation is any mapping θ in $\mathbb{R}^T_{\geq 0}$ such that θ is injective on $th(v)$ for all $v \in V$. An *attractor parameter valuation* is any mapping κ in $\mathbb{R}^K_{\geq 0}$. Given a linear expression e in $\mathbb{L}(K)$, and a valuation $\kappa \in \mathbb{R}^K_{\geq 0}$, we denote by $[\![e]\!]_\kappa$ the value of e in $\mathbb{R}$ obtained by valuating each parameter according to κ. This notation is extended to $\mathbb{P}_f(\mathbb{L}(K)^V)$ in the obvious way.

Definition 3.2 The *operational semantics* of a parametric attractor system

$\mathcal{A} = \langle V, T, th, K, at \rangle$ is the labeled transition system $\mathcal{S}(\mathcal{A}) = (Q, \Sigma, \to)$ where:

- the set of configurations Q is the set of all quadruples $(z, \overrightarrow{x}, \theta, \kappa)$ where $z \in Z(V,T)$, $\overrightarrow{x} \in \mathbb{R}^V_{\geq 0}$, $\theta \in \mathbb{R}^T_{\geq 0}$, and $\kappa \in \mathbb{R}^K_{\geq 0}$, and such that: $\overrightarrow{x} \in [\![z]\!]_\theta$, and θ is injective on $th(v)$ for all $v \in V$,
- the set of labels is $\Sigma = \Delta(V,T) \cup \Lambda(V,T)$, and
- the transition relation $\to \subseteq Q \times \Sigma \times Q$ is defined by: $(z, \overrightarrow{x}, \theta, \kappa) \xrightarrow{\sigma} (z', \overrightarrow{x}', \theta', \kappa')$ if (1) $\theta = \theta'$ and $\kappa = \kappa'$, and (2) one of the two following conditions hold:
  - $\sigma = \delta(z)$, and $z = z'$, and $\overrightarrow{x}' \in hull(\{\overrightarrow{x}\} \cup [\![at(z)]\!]_\kappa)$ (C)
  - $\sigma \in \Lambda(V,T)$, and $z \overset{\sigma}{\rightsquigarrow}_{V,T} z'$, and $\overrightarrow{x} = \overrightarrow{x}'$ (D)

3.3 From regulatory networks to parametric attractor systems

In any regulatory network where γ is a constant mapping (i.e. all degradation rates are equal), trajectories in boxes are straight lines towards the corresponding limit point. In a wall, trajectories point toward the (convex hull of) the limit points of adjacent boxes. These limits are precisely the attractors of the zone corresponding to the wall.

Actually, we can prove that for every regulatory network $\mathcal{N}$, one can compute a parametric attractor system $\mathcal{A}$ such that $\mathcal{S}(\mathcal{A})$ is equal to the time abstract semantics of the differential semantics of $\mathcal{N}$.

4 Symbolic analysis and abstraction

We show in this section how symbolic model-checking technology can be applied to parametric attractor systems. We use the general framework of symbolic transitions systems introduced in [HM00]. Our definition slightly differs from the original one in [HM00], as we use labels instead of observables. We first need a few more basic notions on labeled transition systems.

Given a labeled transition system $\mathcal{S} = (Q, \Sigma, \to)$, the *successor* and *predecessor* functions $\mathrm{post}, \mathrm{pre} : \mathbb{P}(Q) \times \Sigma \to \mathbb{P}(Q)$ are defined by: $\mathrm{post}(U, \sigma) = \{q' \in Q \mid \exists\, q \in U, q \xrightarrow{\sigma} q'\}$ and $\mathrm{pre}(U, \sigma) = \{q \in Q \mid \exists\, q' \in U, q \xrightarrow{\sigma} q'\}$. These two functions are naturally extended to words: $\mathrm{post}(U, \varepsilon) = \mathrm{pre}(U, \varepsilon) = U$, $\mathrm{post}(U, w \cdot \sigma) = \mathrm{post}(\mathrm{post}(U, w), \sigma)$ and $\mathrm{pre}(U, w \cdot \sigma) = \mathrm{pre}(\mathrm{pre}(U, \sigma), w)$. A *trace* in $\mathcal{S}$ is any word $w \in \Sigma^*$ such that $\mathrm{post}(Q, w) \neq \emptyset$.

Definition 4.1 A *symbolic transition system* is a 4-tuple $\mathcal{S} = \langle Q, \Sigma, \to, R \rangle$ where $\langle Q, \Sigma, \to \rangle$ is a labeled transition system, and R is a subset of $\mathbb{P}(Q)$ such that:

- $Q \in R$, and R is closed under finite intersection and set difference, and
- R is closed under post and pre: $\mathrm{post}(r, \sigma) \in R$ and $\mathrm{pre}(r, \sigma) \in R$ for all $r \in R$ and $\sigma \in \Sigma$.

A symbolic transition system is called *effective* when (1) regions are finitely presented, and (2) all operations on regions (intersection, difference, emptiness, post and pre) are computable.

4.1 Symbolic analysis

This section is devoted to proving that parametric attractor systems can be embedded into effective symbolic transition systems. We will use formulas in real arithmetic to represent sets of real vectors. Recall that real arithmetic $\langle \mathbb{R}, +, \times, \leq \rangle$ is decidable.

Consider a parametric attractor system $\mathcal{A} = \langle V, T, th, K, at \rangle$, along with operational semantics $\mathcal{S}(\mathcal{A}) = (Q, \Sigma, \rightarrow)$. We define the set of regions $R_{\mathcal{A}}$ as the set of subsets $r \subseteq Q$ such that for every zone $z \in Z(V, T)$, the set $\{(\overrightarrow{x}, \theta, \kappa) \in \mathbb{R}^V_{\geq 0} \times \mathbb{R}^T_{\geq 0} \times \mathbb{R}^K_{\geq 0} \mid (z, \overrightarrow{x}, \theta, \kappa) \in r\}$ is definable in real arithmetic. In other words, regions can be seen as finite unions of pairs (z, f) where z is a zone and f is real arithmetic formula with free variables in $V \cup T \cup K$.

Proposition 4.2 *For every parametric attractor system $\mathcal{A}$, the pair $\langle \mathcal{S}(\mathcal{A}), R_{\mathcal{A}} \rangle$ is an effective symbolic transition system.*

Proof It is readily seen that regions are finitely presented, that $Q \in R$, and that $R_{\mathcal{A}}$ is effectively closed under finite intersection and set difference. Let us prove that $R_{\mathcal{A}}$ is effectively closed under post and pre. It is sufficient to show that for every label σ in Σ, the binary relation $\xrightarrow{\sigma}$ is effectively definable in real arithmetic, by a formula with free variables z, $\overrightarrow{x}$, t, k, z', $\overrightarrow{x}'$, t' and k'. There are two cases:

- if $\sigma \in \Delta(V, T)$ then $\xrightarrow{\sigma}$ is defined in real arithmetic by the following formula:

$$z \overset{\sigma}{\rightsquigarrow}_{V,T} z' \quad \wedge \quad \overrightarrow{x} = \overrightarrow{x}' \quad \bigwedge_{y \in V \mid \sigma(v) \neq \perp} x(v) = t(\sigma(v)) \quad \bigwedge_{t \in T} t = t' \quad \bigwedge_{k \in K} k = k'$$

  where $t(\sigma(v))$ denotes the threshold parameter of the component cross label $\sigma(v)$.
- if $\sigma = \delta(z'')$, with $z'' \in Z(V, T)$ then $\xrightarrow{\sigma}$ is defined in real arithmetic by the following formula:

$$z = z' = z'' \ \wedge \ Hull[at(z) \cup \{\overrightarrow{x}\}](\overrightarrow{x}') \quad \bigwedge_{t \in T} t = t' \quad \bigwedge_{k \in K} k = k'$$

  where $Hull[\{\overrightarrow{x_1}, \overrightarrow{x_2}, \ldots, \overrightarrow{x_n}\}](\overrightarrow{x'})$ is a standard real arithmetic formula expressing that $\overrightarrow{x}'$ belongs to the convex hull of $\{\overrightarrow{x_1}, \overrightarrow{x_2}, \ldots, \overrightarrow{x_n}\}$.

$\square$

As a direct consequence of the previous proposition, we obtain that the set of configurations satisfying a given bounded reachability property is an effectively computable region.

4.2 Symbolic analysis in the parametric linear real arithmetic

All the operations used previously for the analysis of parametric attractor systems are defined in real arithmetic $\langle \mathbb{R}, +, \times, \leq \rangle$. They involve *quantifier elimination* (given an existentially quantified formula $\exists x.\varphi$, where φ is quantifier-free, find a quantifier-free formula ψ such that $\exists x.\varphi$ and ψ are equivalent) and *satisfiability* (is a given formula φ satisfiable?). The most efficient known algorithms for these two problems are exponential in the number of variables [Mis97]. Nevertheless, some algorithms which are doubly exponential in the number of variables, such as [CH91], have a better practical complexity. Furthermore, observe that the complexity of the satisfiability problem is polynomial [Kar84] for linear formulas of the form $A\overrightarrow{x} \leq \overrightarrow{b}$ where A and $\overrightarrow{b}$ have rational coefficients. This is the reason why linear variables in arbitrary (non linear) formulas are treated separately [LW93].

We can show that our symbolic regions are linear in the variables V (but not necessarily linear in the parameters $K \cup T$): they are called *parametric linear formulas*. Moreover the computations of regions can be done only eliminating linear variables. Actually, the original zones are linear in the variables and the computation of predecessors and successors can be done eliminating linear variables (with a well chosen order). Consequently, a satisfiability solver is needed only for answering the satisfiability of (non-linear) conditions on the parameters.

4.3 Abstraction

We may now use our symbolic analysis technology to define arbitrarily precise abstractions. Any abstraction necessarily has more traces than the abstracted system. Our objective is to obtain abstractions that do not have too many "spurious" traces. The abstractions that we consider are basically predicate abstractions [GS97] where predicates are defined by regions.

Recall that given two labeled transition system $\mathcal{S}_1 = (Q_1, \Sigma, \rightarrow_1)$ and $\mathcal{S}_2 = (Q_2, \Sigma, \rightarrow_2)$, a *simulation* from $\mathcal{S}_1$ to $\mathcal{S}_2$ is any relation $\rho \subseteq Q_1 \times Q_2$ such that $\rho^{-1} \rightarrow_1 \subseteq \rightarrow_2 \rho^{-1}$. We say that $\mathcal{S}_1$ is an *abstraction* of $\mathcal{S}_2$ when there exists a simulation from $\mathcal{S}_2$ to $\mathcal{S}_1$.

Given $k \in \mathbb{N}$ and a parametric attractor system $\mathcal{A} = \langle V, T, th, K, at \rangle$, the *k-precise abstraction* of $\mathcal{S}(\mathcal{A}) = (Q, \Sigma, \rightarrow)$ is the finite labeled transition system $\mathcal{S}_k^\sharp(\mathcal{A}) = (Q_k^\sharp, \Sigma, \rightarrow_k^\sharp)$ where:

- the set of abstract states $Q_k^\sharp$ is the partition of Q induced by the following set of regions: $P_k^\sharp = \{\text{pre}(Q, w) \mid w \in \Delta(V,T)(\Lambda(V,T)\Delta(V,T))^i, i \leq k\}$.
- the abstract transition relation $\rightarrow_k^\sharp \subseteq Q_k^\sharp \times \Sigma \times Q_k^\sharp$ is defined by: $r \xrightarrow{\sigma}{}_k^\sharp r'$ if $\text{pre}(r', \sigma) \cap r \neq \emptyset\}$.

Intuitively, $\mathcal{S}_k^\sharp(\mathcal{A})$ is the predicate abstraction of $\mathcal{S}(\mathcal{A})$ w.r.t. the set of predicates $P^\sharp$. Observe that our 0-precise abstraction is actually equivalent to the one presented in [dJG+04], as abstract states of $\mathcal{S}_0^\sharp(\mathcal{A})$ are precisely sets of states contained in the same zone.

The following proposition shows that $\mathcal{S}_k^\sharp(\mathcal{A})$ is an abstraction of $\mathcal{S}(\mathcal{A})$ that has the same traces of length at most k as $\mathcal{S}(\mathcal{A})$.

Proposition 4.3 *Let $\mathcal{A}$ be a parametric attractor system. For every $k \in \mathbb{N}$, we have:*

- *$\mathcal{S}_k^\sharp(\mathcal{A})$ is an abstraction of $\mathcal{S}(\mathcal{A})$, and*
- *for every $w \in \Delta(V,T)(\Lambda(V,T)\Delta(V,T))^i$ with $i \leq k+1$, w is a trace of $\mathcal{S}(\mathcal{A})$ iff w is a trace of $\mathcal{S}_k^\sharp(\mathcal{A})$.*

Note that our notion of precision can be seen as abstraction refinement: $\mathcal{S}_{k+1}^\sharp(\mathcal{A})$ is a refinement of $\mathcal{S}_k^\sharp(\mathcal{A})$ as it is based on more predicates.

Remark 4.4 We may define a similar abstraction using post instead of pre. This "forward" abstraction would also satisfy the previous proposition.

The abstraction $\mathcal{S}_k^\sharp(\mathcal{A})$ captures all possible abstractions corresponding to different parameter values. In particular, we don't have to enumerate all orderings between parameters to compute the 0-precise abstraction: the relevant parameter constraints are automatically synthesized.

5 Case study with a simple example

In this last part, we illustrate the interest of our method with an example. Let us consider the regulatory network given by Figure 2.

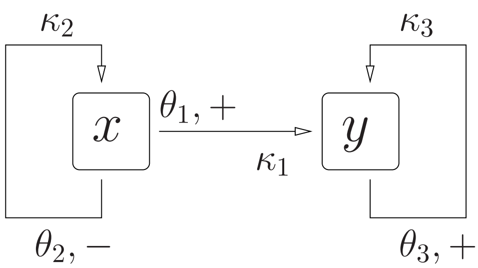

Figure 2. Case study example

We assume that $\gamma_x = \gamma_y = 1$. Then the differential system is:

$$\begin{aligned} \tfrac{d}{dt}x &= \kappa_2 s^-(x,\theta_2) - x \\ \tfrac{d}{dt}y &= \kappa_1 s^+(x,\theta_1) + \kappa_3 s^+(y,\theta_3) - y \end{aligned}$$

Figure 3 gives the positions of the attractors for values satisfying $0 < \kappa_2 < \theta_1$, $0 < \theta_3 < \kappa_3 < \kappa_1$ and $\theta_1 < \frac{\kappa_1(\theta_1-\kappa_2)}{\kappa_1-\theta_3} + \kappa_2 < \theta_2$. The last conjunct comes from the separation of box B (using Thales Theorem in the triangle $(\kappa_2, 0)$,

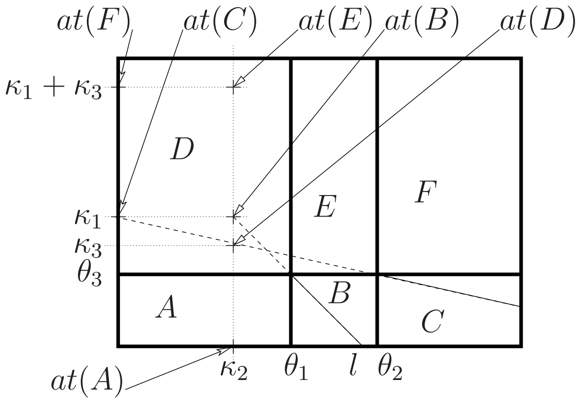

Figure 3. Positions of the attractors

$(l, 0)$ and (κ_2, κ_3)). These constraints are automatically synthesized by our abstraction algorithm.

Observe that zone B is cut when computing the predecessors in B of $A \cap B$, which are given by the lower triangle of B. The 1-precise abstraction restricted to parameters satisfying the above constraint is given in Figure 4. In this abstraction, there are two stationary states represented by circles: one in zone A, the other in zone D.

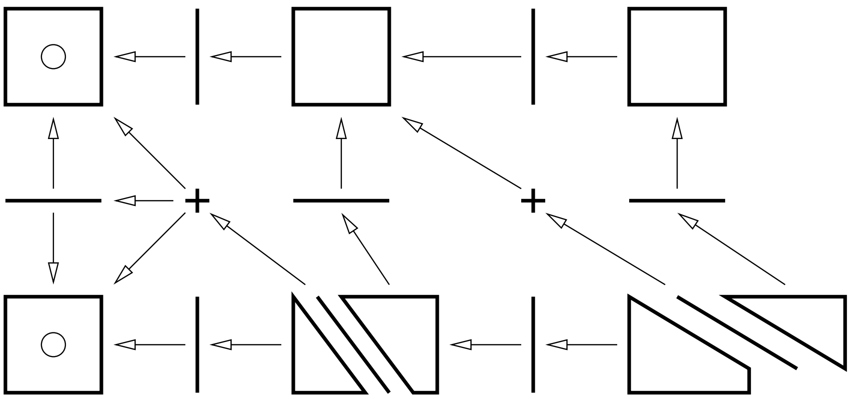

Figure 4. 1-precise abstraction

Recall that our 0-precise abstraction is equivalent to the abstraction that would be constructed by [dJG+04]: regions are zones and no zone is divided. Consequently, in the abstraction made by [dJG+04], from zone C, it is possible to go to both stationary states. In our 1-precise abstraction, trajectories starting in zone C always lead to zone D.

6 Conclusion

We have presented a new class of hybrid systems, the so-called parametric attractor systems, which gives a time-abstract semantics for regulatory networks having a uniform degradation rate. We show that parametric attractor systems are effective symbolic transition systems, which regions and operations can be performed in parametric linear real arithmetic. This allows us to define arbitrarily precise predicate abstractions for these systems.

Our main restriction is the assumption that γ is a constant mapping. However, without this assumption, continuous steps could involve radicals not definable in real arithmetic.

We intend to investigate the practical feasibility of our approach using the satisfiability solver (for real arithmetic) presented in [CH91].

Acknowledgement

We wish to thank Macha Nikolski and Pascal Durrens for insightful discussions on biological regulatory networks, and the anonymous referees for their helpful comments.

References

[AMP95] Eugene Asarin, Oded Maler, and Amir Pnueli. Reachability analysis of dynamical systems having piecewise-constant derivatives. *Theoretical Computer Science*, 138(1):35–65, 1995.

[CC79] P. Cousot and R. Cousot. Systematic design of program analysis frameworks. In *Proc. 6th ACM Symp. Principles of Programming Languages, San Antonio, TX, USA*, pages 269–282. ACM Press, 1979.

[CH91] George E. Collins and H. Hong. Partial cylindrical algebraic decomposition for quantifier elimination. *JSC*, 12(3):299–328, 1991.

[dJ02] Hidde de Jong. Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 9(1):67–103, 2002.

[dJG+04] Hidde de Jong, Jean-Luc Gouze, , Celie Hernandez, Michel Page, Tewfik Sari, and Johannes Geiselmann. Qualitative simulation of genetic regulatory networks using piecewise-linear models. *Bulletin of Mathematical Biology*, 66:301–340, 2004.

[GS97] S. Graf and H. Saïdi. Construction of abstract state graphs with PVS. In *Proc. 9th Int. Conf. Computer Aided Verification (CAV'97), Haifa, Israel, June 1997*, volume 1254 of *Lecture Notes in Computer Science*, pages 72–83. Springer, 1997.

[GS03] J-L. Gouzé and T. Sari. A class of piecewise linear differential equations arising in biological models. *Dynamical systems*, 17:299–316, 2003.

[GTT03] R. Ghosh, A. Tiwari, and C. Tomlin. Automated symbolic reachability analysis with application to delta-notch signaling automata. In O. Maler and A. Pnueli, editors, *Hybrid Systems: Computation and Control HSCC*, volume 2623 of *LNCS*, pages 233–248. Springer, April 2003.

[Hen96] Thomas A. Henzinger. The theory of hybrid automata. In *LICS*, pages 278–292, 1996.

[HM00] T. A. Henzinger and R. Majumdar. A classification of symbolic transition systems. In *Proc. 17th Ann. Symp. Theoretical Aspects of Computer Science (STACS'2000), Lille, France, Feb. 2000*, volume 1770 of *Lecture Notes in Computer Science*, pages 13–34. Springer, 2000.

[Kar84] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. In *STOC*, pages 302–311, 1984.

[LGS+95] C. Loiseaux, S. Graf, J. Sifakis, A. Bouajjani, and S. Bensalem. Property preserving abstractions for the verification of concurrent systems. *Formal Methods in System Design*, 6(1):1–44, 1995.

[LW93] Rüdiger Loos and Volker Weispfenning. Applying linear quantifier elimination. *The Computer Journal*, 36(5):450–462, 1993.

[Mis97] Mishra. Computational real algebraic geometry. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Discrete and Computational Geometry*. CRC Press, 1997.

[MPO95] Th. Mestl, E. Plahte, and S.W. Omholt. A mathematical framework for describing ans analysins gene regulatory networks. *Journal of Theoretical Biology*, 176:291–300, 1995.

[TCT93] D. Thieffry, D. Collet, and R. Thomas. Formalisation of regulatory networks: a logical method and its automatization. *Math. Modelling and Computing*, 2:144–151, 1993.

[TD90] R. Thomas and R. D'Ari. *A logical and differential description of regulatory networks involving intertwined circuits.* CRC Press, 1990.

[Tho91] R. Thomas. Regulatory networks seen as asynchronous automata: a logical description. *Journal of Theoritical Biology*, 153:1–23, 1991.

[Tho98] R. Thomas. Laws for the dynamics of regulatory networks. *International Journal Developmental Biology*, 42:479–485, 1998.

[TK02] A. Tiwari and G. Khanna. Series of abstractions for hybrid automata. In C. J. Tomlin and M. R. Greenstreet, editors, *Hybrid Systems: Computation and Control HSCC*, volume 2289 of *LNCS*, pages 465–478. Springer, March 2002.