

# Marked Regulatory Graphs: A Formal Framework to Simulate Biological Regulatory Networks with Simple Automata

Vincent Bassano, Gilles Bernot  
LaMI, Université d'Evry,  
France  
{vbassano, bernot}@lami.univ-evry.fr

## Abstract

*In the field of biological regulation, models obtained from experimental biology are usually complex networks of induction and repression between genes. Due to the development of high throughput genomic, it is now necessary to treat large scale networks. The challenge is to automatically analyze their behaviour.*

*Here, we propose a formal framework to define this Biological Regulatory Networks.*

*Our model is derived from R. Thomas representation, where a biological regulatory network can be seen as a discrete model. We propose a formal definition of such representation. We separate the static part (description of the systems) from the dynamic one (we develop two semantics, synchronous or asynchronous, to illustrate our method).*

*A software environment to support this framework is also described. This software, developed to be used by biologists, can be used to rapidly obtain a prototype describing the behaviour of the system, to product simulations and to automatize proofs of properties.*

## Introduction

In biology, regulatory networks, as defined by the group of R. Thomas [15], allows to describe the influences between elements (or variables) inside biological systems. These variables can represent the expression level of genes. Influences are repression or induction of genes or genes products on others.

This approach has a significant impact on biological modelling and helps to understand the behaviour of biological systems ([3], [12]). We extend this approach into a formal framework to be more easily tractable by computer methods ([1], [2]).

We believe that this approach could facilitate many contribution of computer scientists to biology.

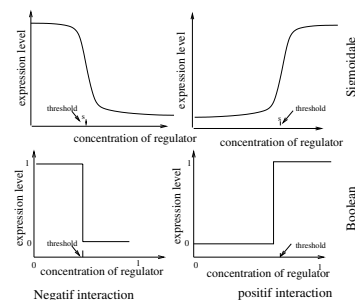
The first section introduces R. Thomas' regulatory networks and explains our choices. The first part of the second section describes the topological part of our framework, while the second part defines its parameterisation aspects. The third part deals with the dynamic: synchronous and asynchronous semantics and formal definitions to manipulate them. The last section describes the computer environment which supports the framework.

## 1 Logical modelling of regulatory networks

**Boolean modelling:** In the 70's R. Thomas [14] has defined a way to model regulatory networks based on boolean representations. This modelling has been widely used, tested and extended ([13], [16]).

In this part, we will give a brief computer oriented description of this modelling and make some commentary on it.

The approach is based on two classes of elementary regulatory interactions. The positive ones, where the regulator favours the expression of its target, and the negative ones, where the regulator inhibits the expression of its target .



As usual in biological system, most interactions between the elements of a regulatory network are non-linear and usually sigmoid shaped. Instead of describing these interactions by their continuous curves (involving differential equations), the boolean representations describe them by a step function with two logical values: True for an activated

variable, False for a deactivated one.

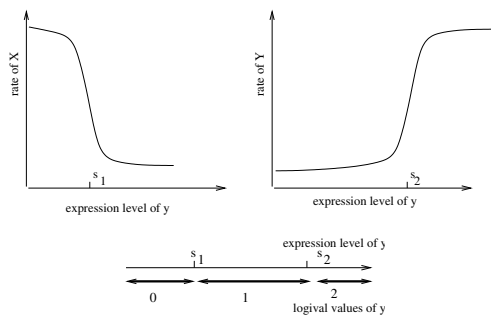
Another suited abstraction of this approach is to consider a discrete time.

At a time  $t$ , we can consider the state vector  $(x_t, y_t, \dots)$  where  $x, y$  are the discretised expression level of the variables of the system. Likewise we can define  $(X_t, Y_t, \dots)$  the *image vector*, representing the attractor state at the successor time  $succ(t)$ . The application giving the image vector from the state vector does not depend from a given time  $t$ .

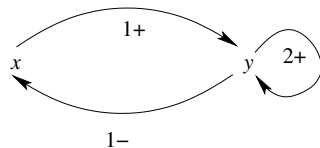
In R. Thomas representation, a network is represented as a labelled graph with variables as nodes and the existence of influence as arcs.

**René Thomas' multivalued regulatory network** An other innovation of R. Thomas is to consider that the range of a variable  $x$  is not necessarily reduced to boolean value, it can belong to the set of values  $\{0, s_1, 1, s_2, 2, \dots, n - 1, s_n, n\}$ , where the  $s_i$  are thresholds. Thus, we have  $x = 0$  if  $x < s_1$  and  $x = i$  if  $x > s_i$  and  $x < s_{i+1}$ .

For example if  $y$  acts negatively on  $x$  and positively on itself as follows, we get



and if moreover  $x$  acts positively on  $y$ , we draw figure 1



**Figure 1.** An example of multivalued interaction graph

**Logical parameters** In order to compute the image vector  $(X_t, Y_t, \dots)$  from  $(x_t, y_t, \dots)$  R. Thomas introduced the so called *logical parameters*:

- $K_x \emptyset$  is the basal weight of  $x$ . It is the expression level of  $x$  in the absence of all activators and in the presence of all repressors.

- more generally  $K_x E$ , is the value of  $X_t$  if  $E$  is the set of variables  $v$  such that ,

- If  $v$  activate  $x$  then the level of  $v_t$  is greater than its threshold
- If  $v$  repress  $x$  then the level of  $v_t$  is lower than its threshold

**Synchronous and asynchronous approaches** Traditionally, biologist distinguish two kinds of dynamic to represent the temporal evolution of the system:

- Synchronous dynamic: all the values of the variables are updated simultaneously to the image value at predecessor time.  $(x_t, y_t, \dots) = (X_{pred(t)}, Y_{pred(t)}, \dots)_{pred(t)}$ .
- Asynchronous dynamic: only one variable is updated at a time (e.g. the fastest reaction) and other variables are left unchanged.

synchronous	asynchronous
$00 \xrightarrow{[01]} 11$ $11 \xrightarrow{[10]} 00$	$00 \xrightarrow{[01]} 11$ $11 \xrightarrow{[10]} 00$

**Discussion** In our perspective, the network of figure 1 can be read as

- $x$  receives a negative influence if  $y$  is at level 1, or in other words :  $y$  acts as activateur on  $x$  when absent
- $y$  receives a positive influence if  $x$  is at level 1
- $y$  receives a positive influence if  $y$  is at level 2

From a certain point of view, resources are transmitted threw transition, it reminds us strongly of resources sharing . One of the means to treat such systems is a bipartite place-transition graph (such as Petri net). Our work adapts these ideas to biological contingencies while taking into account some particularities:

- In a regulatory network a high expression level of a variable  $x$ , inducing a variable  $y$ , can be seen as a resource of  $y$ . Similarly, if  $y$  is repressed by  $x$ , a low expression level of  $x$  is considered as a resource of  $y$ .
- Transitions do not consume their resources by default.
- when several transitions are fired, the ressources are not added. So, we have taken into account a sort of "logical parameters" to compute the marking of the places (we can mention nevertheless [4] and [8] for modelling genetic networks by using classical Petri net methods and [7] and [10] for their application to metabolic networks and [6] for modelling biological system with statecharts , this works do not deals with Thomas theory).

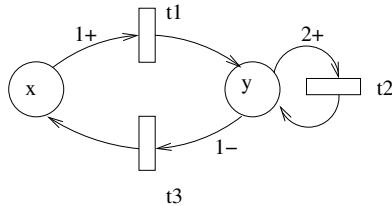
## 2 A formal description of biological regulatory networks

### 2.1 Static regulatory graphs

They describe the relations between the variables and the mean to represent them.

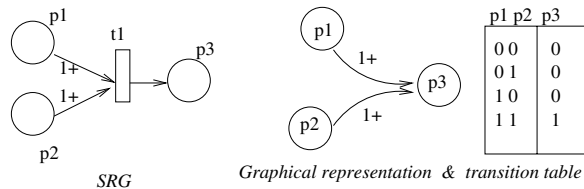
**Definition 1 :** [Static regulatory graph]. A *Static regulatory graph* (SRG for short) is a labelled, bipartite, directed graph  $\mathcal{G} = (V, T, Out, In)$  where :

- $V$  is the first set of nodes, such that each node of  $V$  is a *variable*  $x$  with an associated *bound*  $b_x$  belonging to  $\mathbb{N}^* = [1, 2, 3, \dots]$
- $T$  is the second set of nodes and each node  $\tau$  of  $T$  is called a *transition*
- $Out$  is the set of arcs from  $V$  to  $T$ , such that each arc  $(x \rightarrow \tau)$  is labelled by a couple  $(s, \varepsilon)$  where  $s$  belongs to the integer interval  $[1, b_x]$  and  $\varepsilon$  is either equal to “+” or to “-”
- $In$  is the set of arcs from  $T$  to  $V$  (with no labels).



**Figure 2.** Example of a SRG

**Remark:** Obviously every graph of R. Thomas can be represented by a SRG. The reverse is false: as an example, the SRG representation of figure 3 (a simple AND relation) can be directly seen on the topology of the graph. On the



**Figure 3.** Comparison of SRG and boolean representations of a same AND relation

contrary, to represent the relation between  $p_1$  and  $p_2$  in the

modelling of Thomas we have to introduce the transition table.

Owing to explicit transitions, our formalism enables to clearly express cooperation of variables at the graphic level.

**Notation** Given a SRG  $\mathcal{G} = (V, T, Out, In)$ :

- for every variable  $x \in V$ ,  $x_T^{-1} = \{\tau \in T | (\tau \rightarrow x) \in In\}$  denotes the set of inducing transitions of the variable  $x$ ;
- for every transition  $\tau \in T$ ,  $\tau_V^{-1} = \{x \in V | (x \rightarrow \tau) \in Out\}$  denotes the set of inducing variables of the transition  $\tau$ .

### 2.2 Marking and Parameters

In this part we define: the nature of the objects involved in the dynamic of the system, the fixed rules governing them, and the variations of their expression level. It finally leads to define the quantitative aspects of regulatory networks.

**Definition 2 :** [Marking]. Given a SRG  $\mathcal{G} = (V, T, Out, In)$ , a *marking* on  $\mathcal{G}$  is a total function  $\mu : V \rightarrow \mathcal{P}(T)$ , (where  $\mathcal{P}(T)$  is the powerset of  $T$ ) such that, for every  $x \in V$  we have  $\mu(x) \subseteq x_T^{-1}$ .

The resources of a variable are transitions which influence its level of expression.

Let us define the way to deduce the expression level of a variable  $x$  from its marking via a parameterisation function (analogous to logical parameters).

**Definition 3 :** [Parameters and PRG]. Given a SRG  $\mathcal{G} = (V, T, Out, In)$ , a *parameterisation function* on  $\mathcal{G}$  is a partial function  $\mathcal{K} : V \times \mathcal{P}(T) \rightarrow \mathbb{N}$  such that:

- $\mathcal{K}(x, S)$  is defined if and only if  $S \subseteq x_T^{-1}$
- if  $\mathcal{K}(x, S)$  is defined then  $0 \leq \mathcal{K}(x, S) \leq b_x$

We call *Parameterised Regulatory Graph* (PRG) a couple  $(\mathcal{G}, \mathcal{K})$  where  $\mathcal{G}$  is a SRG and  $\mathcal{K}$  is a parameterisation function on  $\mathcal{G}$ .

It will be convenient later to consider that undefined values of  $\mathcal{K}$  are equal to 0. For the example of Figure 2 one possible instantiation of the parameterisation function among others is:  $\mathcal{K}_x\{t3\} = 1$ ,  $\mathcal{K}_y\{t1\} = 1$ ,  $\mathcal{K}_y\{t2\} = 2$ ,  $\mathcal{K}_y\{t1, t2\} = 2$ . The possible instantiations of  $\mathcal{K}$  are not unique and have a strong biological sense.

## 2.3 Dynamic aspects of marked regulatory graphs

We now have to express rules to describe the temporal evolution of the system (more particularly its marking).

**Definition 4 :** [Enabled Transition]. Given a PRG  $(\mathcal{G}, \mathcal{K})$ , and a marking  $\mu$  on  $\mathcal{G}$ , a transition  $\tau$  is *enabled* according to  $\mu$  if for all  $(v \rightarrow \tau) \in Out$  labelled by  $(s, \varepsilon)$  we have

- $\mathcal{K}(v, \mu(v)) \geq s$  if  $\varepsilon = +$
- $\mathcal{K}(v, \mu(v)) < s$  if  $\varepsilon = -$

In other words, a transition is enabled if all its inducing variables are greater than their thresholds and all its repressing variables are lower than their thresholds: enabled transitions always help the target variables to be expressed. So, contrarily to Petri nets, a non enabled transition can have an action: to repress the variables (see Def 6).

### 2.3.1 Synchronous Semantics of a MRG

Here, all the transitions occur at the same time. The process is deterministic and each marking has only one successor marking.

**Definition 5 :** [Successor marking (synchronous)]. Given a PRG  $(\mathcal{G}, \mathcal{K})$ , and a marking  $\mu$  on  $\mathcal{G}$  the *successor marking of  $\mu$  by  $\mathcal{K}$*  is the marking  $\mu'$ , defined for all  $u$  in  $V$  by

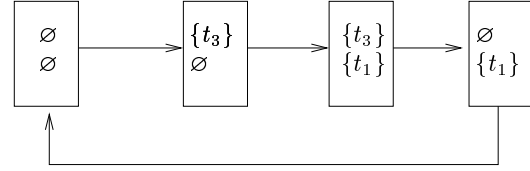
$$\mu'(u) = \{\tau \in u_T^{-1} \mid \tau \text{ is enabled according to } \mu\}$$

The marking of a variable  $u$  is at any time the whole set of the transitions which help the expression of  $u$ .

**Example** For the example of figure 2 with the parameterisation function defined at the end of section 2.2, for the initial marking  $\mathcal{M}_0(x) = \emptyset$ ,  $\mathcal{M}_0(y) = \emptyset$ ,  $\mathcal{M}_1$ , successor marking of  $\mathcal{M}_0$  by  $\mathcal{K}$ , is build as follow:

- $t_3 \in \mathcal{M}_1(x)$  if  $t_3$  is enabled, thus if  $\mathcal{K}_y\{\mathcal{M}_0(y)\} < 1$  (it is a negative interaction), meaning if  $\mathcal{K}_y\emptyset < 1$ : it is true
- $t_2 \in \mathcal{M}_1(y)$  if  $t_2$  is enabled, thus if  $\mathcal{K}_y\{\mathcal{M}_0(y)\} \geq 2$ , meaning if  $\mathcal{K}_y\emptyset \geq 2$ : it is false
- $t_1 \in \mathcal{M}_1(y)$  if  $t_1$  is enabled, thus if  $\mathcal{K}_x\{\mathcal{M}_0(x)\} \geq 1$ , meaning if  $\mathcal{K}_x\emptyset \geq 1$ : it is false

So, the marking  $\mathcal{M}_1$  is  $\mathcal{M}_1(x) = \{t_3\}$ ,  $\mathcal{M}_1(y) = \emptyset$ . Similarly, we can also define  $\mathcal{M}_2(x) = \{t_3\}$ ,  $\mathcal{M}_2(y) = \{t_1\}$  and  $\mathcal{M}_3(x) = \emptyset$ ,  $\mathcal{M}_3(y) = \{t_1\}$  whose successor marking  $\mathcal{M}_4$  is equal to  $\mathcal{M}_0$ :



### 2.3.2 Asynchronous Semantics of a MRG

With respect to our asynchronous semantics, only one transition at a time is considered to pass from a marking to its successor. Thus a given marking can have more than one possible successor.

**Definition 6 :** [Successor Marking (asynchronous)]. Given a PRG  $(\mathcal{G}, \mathcal{K})$ , and a marking  $\mu$  on  $\mathcal{G}$ , let  $\tau$  be a transition of  $\mathcal{G}$ , the successor marking  $\mu'$  of  $\mu$  by  $\tau$  is defined as:

For all  $(\tau \rightarrow u) \in In$ ,

- if the transition is enabled then  $\mu'(u) = \mu(u) \cup \tau$
- else  $\mu'(u) = \mu(u) \setminus \tau$

and for all other variables  $v$ ,  $\mu'(v) = \mu(v)$ .

the operator  $\setminus$ , is the subtraction of set theory:  $\mu(u) \setminus \tau$  is the set  $\mu(u)$  except  $\tau$ . Note that if  $\tau \notin \mu(u)$ ,  $\mu(u) \setminus \tau$  is  $\mu(u)$  itself.

As every transition can be chosen, there are *a priori*  $n$  possible successors of  $\mu$ , where  $n$  is the number of transitions.

### 2.3.3 State automata

Successor marking being defined (synchronous or asynchronous) we can build the automata representing the evolution of the marking of the system.

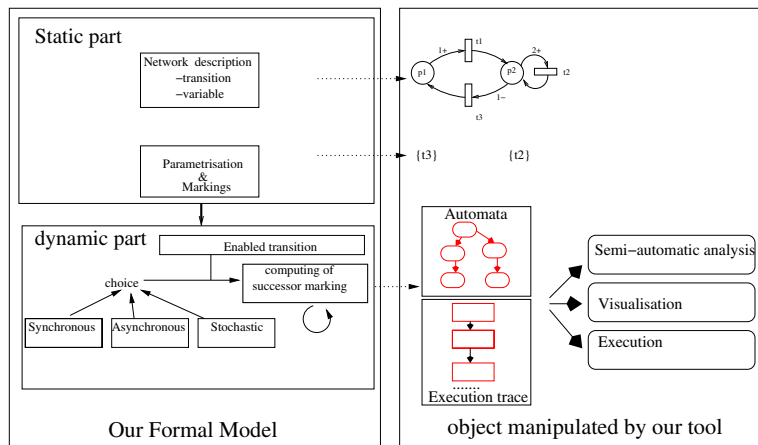
**Definition 7 :** [State automata]. Given a PRG  $(\mathcal{G}, \mathcal{K})$ , its state automata is the labelled directed graph  $\mathcal{A} = (N, A)$  where

- $N$  is the set of nodes and contains all possible markings on  $\mathcal{G}$
- $A$  is the set of arcs, such that  $(\mu \rightarrow \mu') \in A$  if and only if there exist  $\tau$  such that  $\mu'$  is the successor marking of  $\mu$  by  $\tau$ .

## 3 $M_eR_{ovin}G_{ien}$ : An environment to analyse MRG

The main goal of our formal approach of R. Thomas modelling is to allow to implement well founded platforms for regulatory networks analysis.

Here, we outline the specification of our environment whose main goal is to help biologists to make a prototype of a regulatory network, to modify it, to explore its behaviour, to



**Figure 4.** The architecture of our approach and the tool.

validate its parameters, etc. according to their knowledge and biological experiments. Well chosen tools, integrated within the environment, are needed for:

- the simulation and the observation of a system behaviour, the representation of its dynamic and the semi-automatic analysis of its traces
- network structural analysis (feedback loops, patterns, etc.)
- the visualisation and representation of static regulatory networks
- the creation of the state automata of the system and its visualisation
- as well as more general tools such as model checking and formal proofs of properties in temporal logic [9].

The role of the environment is also to hide the internal representation of the implantation of our model to allow an easy plug of new tools in a “user oriented” manner.

Our platform is divided in three main categories of tools corresponding more or less to the static/ parameters/ dynamic structuration of the formal framework:

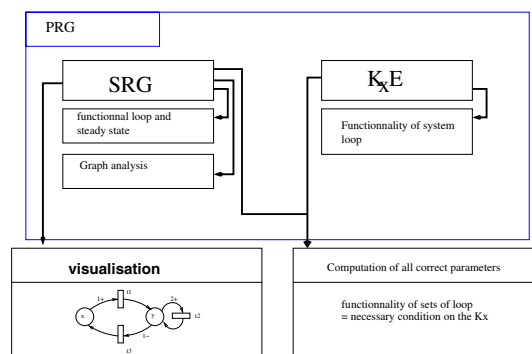
- description tools : the creation of a prototype of the regulatory networks and the study of all properties deducible without a temporal evolution
- automata tools: the creation of state automaton and the validation and verification of system properties
- simulation trace analysis tools : generation and study of execution traces

### 3.1 Description tools

Such tools allow us to describe (and construct) our networks. They correspond in our formal framework to the definition of the static part of a system. Here, two kinds of object are manipulated

- those concerning the topological description : the equivalent of the *SRG*
- those concerning the description of the rule governing the evolution of the system : implementation of the parameterisation function

This category of tools has to include static analysis of regulatory networks.



Manipulation and visualisation of graphs form the basic library that will be used. The visualisation part is not only a graphical tool. In the long run, it has to be able to include the notion of “black-box” approach for some subgraphs. It is a tool to understand, analyse and represent topological properties of the system.

One has also to perform classical graph analysis : strongly

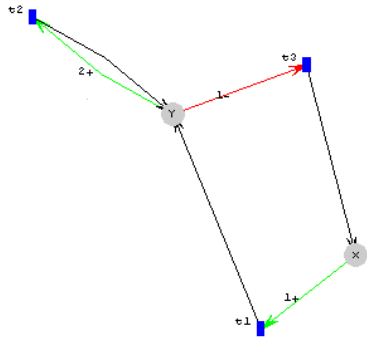


Figure 5. visualisation of a SRG

connected components, pattern with strong biological behaviour [11], common subgraph between different species etc. are useful properties directly deductible from graph theory. An example of a graph with this tool can be seen in Figure 5

Tools to manage the parametrisation function are able to manipulate several versions of the parameter values, to classify them, etc. Those classifications of the possible parameterisation function are strongly linked to the functionality of the feedback loops (cf [16])

One of the main problems for making a simulation or for determining the behaviour of a system was the determination of the correct parameters. Here we provide a tool answering this problem. Parameter values are not usually known in a biological system. However, experimentally or by hypothesis, we can know the functionality of loops. This tool takes a set of loops and gives all the set of parameters allowing the functionality of these loops.

When we want to make an exhaustive analysis of a system, it is no longer necessary to study all the possible values for the parameters, we just have to study the correct subset. Here are the parameters and their set of values obtained for having the two loops functional in Figure 5

$$\{\{K_x\}, K_x\{t3\}\}, \{K_y\}, K_y\{t2\}, K_y\{t1, t2\}, K_y\{t1}\}\} \\ \{\{0\}, \{1\}\}, \{\{0\}, \{2\}, \{2\}, \{1, 2\}\}$$

### 3.2 Automata tools

The goal of these tools is to generate and extract information from the state automata associated with a given PRG. The generation tool contains a main function to allow to find the successor of a marking.

In order to make this operation, we have to specify the kind of dynamic. With our framework, it is easy to keep a strong modularity on this subject. We can switch the dynamic used by switching the function of successor marking.

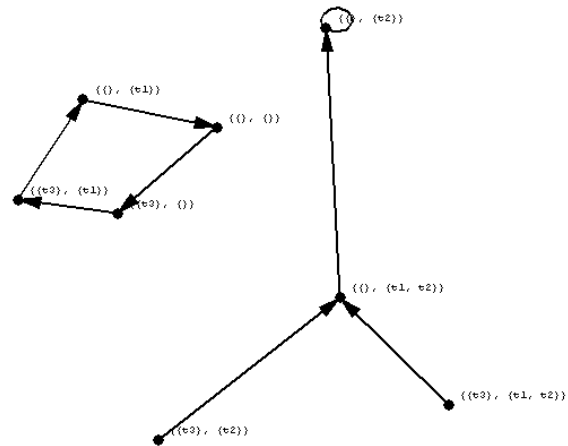
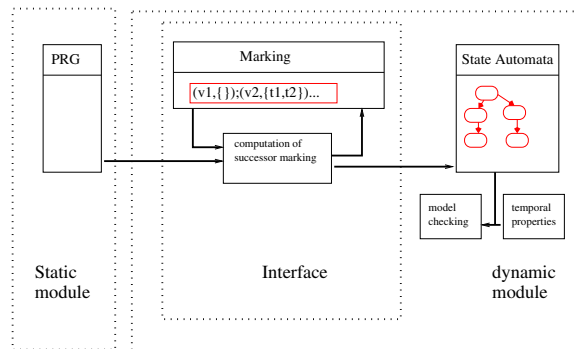


Figure 6. State automata obtained



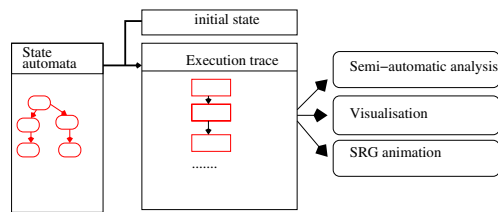
We present the result obtained from figure 5 with the parameterisation  $K_x\{t3\} = 1$ ,  $K_y\{t1\} = 1$ ,  $K_y\{t2\} = 2$ ,  $K_y\{t1, t2\} = 2$  in figure 6

An interface is provided which automatically computes the dynamical evolution of a marking of the system. As from each marking of a variable we can deduce its expression level, we can easily deduce a dual automata where a node is marked by its expression level. It allows us to have a representation more similar to the one of R. Thomas modelling.

### 3.3 Trace analysis

The last category of tools helps the manipulation of execution traces. An execution trace is a sequence of the successive marking of an execution. It allows the visualisation of a simulation. With trace tool, we are able to make a temporal animation of the SRG graph during an execution. More sophisticated tools can be proposed such as proving

that an execution trace correspond to a given state automata. The ultimate goal is indeed to try to answer a reverse engineering question : given a set of traces, what can we infer about its SRG?



## Conclusion and perspectives

We believe that the large corpus of tools offered by formal methods in computer science is not yet used at its full potential by bioinformatics. Formal methods could provide noticeable advances to bioinformatics, particularly in the field of modelling.

We have defined a computer science formal framework of the R. Thomas regulatory networks. This definition gives us a strong basis for computer tool developments and reuse validated by this approach.

We represent interactions by transitions having a positive (or negative) action on the variables. Thus, transitions can be seen as the actual resources of the network. It leads us naturally to use a marking with set of transitions.

In our definitions, we have tried to separate as much as possible the topology of the interaction graph, from the expression level of the variables and from the system dynamics (synchronous or asynchronous).

As future developments of our work, let us mention

- the formal study of feedback loop with temporal logic tools.
- the modelling of populations (set of models where each individual does not have necessarily the same marking than the others).
- some extensions of the notion of transition in order to describe more complex variable interactions.
- the validation of our model on real biological cases. We have already treated epigenetic regulation of mucus production in *Pseudomonas aeruginosa* [5], Moreover we start modelling of the network regulating hematopoiesis.

## 4 Acknowledgments

This work has been partially supported by the Genopole® working group on epigenomics.

Special thanks are due to Janine Guespin-Michel and Denis Thieffry who gave us precious advices.

It is a pleasure to thank Jean-Paul Comet and Franck Delaplace for their valuable comments.

## References

- [1] V. Bassano, G. Bernot, and D. Thieffry. Marked Regulatory Graphs : A formal framework for biological regulatory networks. Interdisciplinary school on Imaging, modelling and manipulating regulatory networks. France, Octobre 2002.
- [2] V. Bassano, G. Bernot, and D. Thieffry. Marked Regulatory Graphs. Graphes de régulation avec marquages, vers un modèle formel informatique des réseaux de régulation biologiques. JOBIM2002, St-Malo, France, Jul 2002.
- [3] H. de Jong. Modeling and simulation of genetic regulatory systems: A literature review. *J Comp Biol*, 9:69–105, 2001.
- [4] P. Goss and J. Peccoud. Quantitative modeling of stochastic systems in molecular biology by using stochastic petri nets. In *Proceedings of the National Academy of Sciences USA*, volume 95, pages 6750– 6754, 1998.
- [5] J. Guespin-Michel. Epigenesis, multistationarity and positive feedback circuits. In *Modelling and simulation of biological process in the context of genomics*, page 193. Proceeding of the Autrans Seminar, 2002.
- [6] N. Kam, I. Cohen, and D. Harel. The immune system as a reactive system: Modeling t cell activation with statecharts. In *Proc. Visual Languages and Formal Methods(VLFM01)*, pages 15–21.
- [7] R. Kuffner, Rand Zimmer and T. Lengauer. Pathway analysis in metabolic databases via differential metabolic display (dmd). *Bioinformatics*, 16:825–836, 2000.
- [8] H. Matsuno, A. Doi, and S. Miyano. Hybrid petri net representation of gene regulatory network. In *Proceedings of the Pacific Symposium on Biocomputing'2000*, pages 341–352, 2000.
- [9] S. Pérès and J. Comet. Contribution of computational tree logic to biological regulatory networks: Example from *pseudomonas aeruginosa*. In *International workshop on Computational Methods in System Biology*, February 2003.
- [10] V. Reddy, M. Liebman, and M. Mavrouniotis. Qualitative analysis of biochemical reaction systems. *Comput Biol Med.*, 26:9–24, 1996.
- [11] S. Shen-Orr, R. Milo, S. Mangan, and U. Alon. Network motif in the transcriptional regulation network of *escheria coli*. *Nat. Genet.*, 31(1), 2002.
- [12] P. Smolen, D. Baxter, and J. Byrne. Modeling transcriptional control in gene networks - methods, recent results, and future directions. *Bull Math Biol*, 62:247–292, 2000.
- [13] E. Snoussy. Qualitative dynamics of a piecewise-linear differential equations : a discrete mapping approach. 1988.
- [14] R. Thomas. Regulatory networks seen as asynchronous automata : A logical description. *J. theor. Biol.*, 153, 1991.
- [15] R. Thomas and R. d'Ari. *Biological Feedback*. CRC Press, 1990.
- [16] R. Thomas, D. Thieffry, and M. Kaufman. Dynamical behaviours of regulatory networks - I . biological role of feedback loops and practical use of the concept of feedback loop. *Bulletin of Mathematical Biology*, 57(2):247–276, 1995.