# Semantics of Biological Regulatory Networks

Gilles Bernot[2], Franck Cassez[1], Jean-Paul Comet[2],
Franck Delaplace[2], Céline Muller[2]
Olivier Roux[1] and Olivier Roux[1]

| IRCCyN[1] | Bio-Info Team[2] |
|---|---|
| UMR CNRS 6597 | LaMI - Genopôle - UEVE |
| BP 92101 | UMR CNRS 8042 |
| 1 rue de la Noë | 523 Place des Terrasses |
| 44321 Nantes Cedex 3 | 91025 Evry Cedex |
| France | France |

July 17, 2003

**Abstract**

The aim of the paper is to revisit the model of Biological Regulatory Networks (BRN) which was proposed by René Thomas to model the interactions between a set of genes. We give a formal semantics for BRN in terms of transition systems which formalizes the evolution rules given by René Thomas. Then we show how to use this model to find interesting properties of a BRN like the set of stable states, cycles etc using tools for analyzing transition systems.

## 1 Introduction

**Modeling of Biological Systems.** The arrival of massive amount of expression data puts the emphasis on computational methods to overcome the difficulties of interpretation of experimental data. Instead of providing a clear explanation of biological systems, data reveals the difficulty for analyzing them. The variety of components and their interacting capabilities lead to cope with their complexity. This opens a field of *modeling* to investigate computational *biological systems*.

Computational *systems biology* [Wol01] tries to establish methods and techniques that enable us to understand the structure of the system, such as gene/metabolic/signal transduction networks. The modeling of the dynamics of such systems is a first step towards the control, the design and the modification of the systems in order to ensure some desired properties[Kit02].

**Formal Methods.** Formal methods have been used for a decade or more in the area of verification of *safety critical systems*. The techniques and tools that have emerged from this field to analyze the behaviors of such systems, makes it possible to *model* and *verify* complex concurrent systems (huge number of states) even with continuous information (dense time) or parameters. It is then natural to try and use such techniques to model and analyze biological systems especially when one wants to find properties about their behaviors.

**Biological Regulatory Networks.** *Biological Regulatory Networks* (in the sequel BRN) modelize interactions between biological entities (RNA or Proteins). Their regulations involve a lot of complex processes, but it is common to simplify the complexity of the regulations by taking into account only two actions: *activation* and *inhibition*. BRNs are statically represented by graphs: *vertices* abstract genes and *edges* represent their interactions (activation or inhibition). Moreover at a given time, a numerical value is associated to each vertex to describe the concentration level of the corresponding entity.

The René Thomas' boolean approach has been justified as a discretization of the continuous differential equation system[Sno89], it has been confronted to the more classical analysis in terms of differential equations[KT87]. Then Thomas and Snoussi showed that all steady states can be found *via* the discrete approach[ST93]. More recently Thomas and Kaufman have shown that the discrete description provides a qualitative fit of the differential equations with a small number of possible combinations of values for the parameters[TK01].

Works of René Thomas and co-workers provide the basis to develop a formal computational framework for gene regulation.

**Our Contribution.** In this paper we propose a semantics for an extended gene regulatory model of R. Thomas' theory. In our extended model a gene can be activator at a certain level and inhibitor at another. This is to our knowledge the first time a formal semantics is proposed for BRN. This enables us to derive automatically a behavioral model of a BRN and use existing tools for analyzing finite state models (e.g. model-checking tools).

**Outline of the Paper.** The paper is organized as follows: section 2 gives the basics of BRN. The core of the paper is in section 3 where we give a formal semantics for BRN. In section 4 we show on a small example how to use the tool HyTech [HHWT97] to analyze a BRN.

# 2 Biological Regulatory Networks

**Notations** Given a finite set $E$, $|E|$ denotes the cardinality of $E$. We denote $2^E$ the set of subsets of $E$. If $\phi$ is a formula of propositional logic over a set $X$, $[\![\phi]\!]$ denotes the set of values of the variables satisfying $\phi$. By convention, if $U = \emptyset$, $\phi(x)$ are propositional formulas, $\bigwedge_{x \in U} \phi(x) = true$. Biological examples often rely on intervals: an integer interval $[a, b]$ stands for the set of values $\{x \in \mathbb{N}, a \leq x \leq b\}$, and we denote $[]$ the empty interval.

The original model of Biological Regulatory Networks [KT87] makes the assumption that the actual concentration of the products of the genes can be approximated by integer levels: the continuous concentration function is approximated by a piecewise constant function. Those constant levels give *the expression levels of the genes.* In our formal description of a Biological Regulatory Network, a set $V$ of "variables" stands for the genes of the network. An oriented edge from a variable $x$ to $y$ indicates that $x$ is a regulator (activator, inhibitor) of $y$.

**Definition 1 (Biological Regulatory Networks.)** *A Biological Regulatory Network (BRN) is a 3-uple $\mathcal{R} = (V, E, \pi)$ where:*

- *$V$ is a finite set of* vertices*,*

- *$E \subseteq V \times V$ is a finite set of* edges*,*

- *$\pi = (\pi_+, \pi_-)$ with $\pi : E \mapsto 2^{\mathbb{N}} \times 2^{\mathbb{N}}$ are respectively the* activation *and* inhibition *functions associated to an edge $e \in E$. Moreover, we assume:*

  - *$\pi_+(e) \cup \pi_-(e) \neq []$*
  - *$\forall e \in E, \pi_+(e) \cap \pi_-(e) = []$: for a given level, a gene cannot be both activator and inhibitor.* □

**Remark 1** *For $v \in V$ we will use $v$ as well to denote the expression level of the gene $v$. $\pi_+(x, y)$ (resp. $\pi_-(x, y)$) gives the interval inside which $x$ activates (resp. inhibits) $y$. Note that $\pi_+(x, y) = []$ (resp. $\pi_-(x, y) = []$), means that $x$ never activates (resp. inhibits) $y$. Note also that Def. 1 rules out edges $(x, y)$ for which $\pi_+(x, y) = \pi_-(x, y) = []$ which would have no observable effect in the network.*

([1,1], [])

([1,1], [2,2])

$x$     $y$

([], [1,1])   ([1,1], [])

$x$     $y$
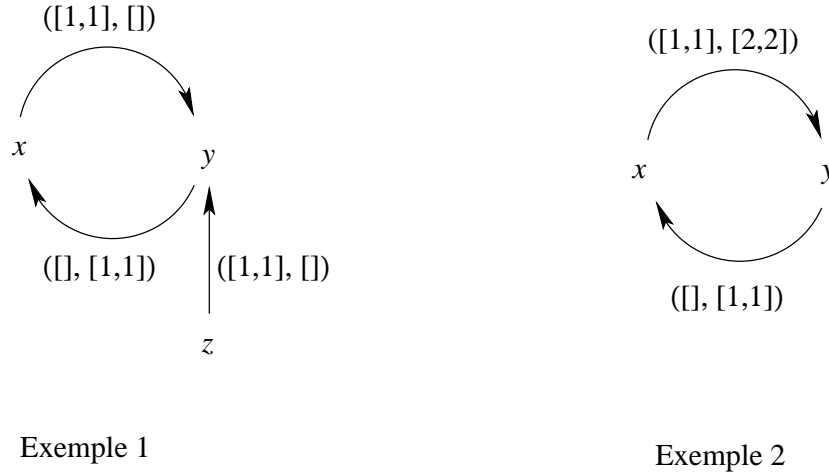
([], [1,1])

$z$

Exemple 1

Exemple 2

Figure 1: Examples of BRNs

**Example 1** *Figure 1 gives two examples of BRN. In the first example, $V = \{x, y, z\}$ and $\pi(x, y) = ([1, 1], [])$: $x$ activates $y$ when $x$ has the level $1$; $\pi(y, x) = ([], [1, 1])$: $y$ inhibits $x$ when $y$ has the level $1$; $\pi(z, y) = ([1, 1], [])$: $z$ activates $y$ when $z$ has the level $1$. The meaning of the second example is defined accordingly.*

**Definition 2 (Activators and Inhibitors.)** *Let $\mathcal{R} = (V, E, \pi)$ be a Biological Regulatory Network, we define the following sets:*

*1. $\forall x \in V, R_+(x) = \{y \in V, \pi_+(y, x) \neq []\}$ is the set of activators of $x$,*

*2. $\forall x \in V, R_-(x) = \{y \in V, \pi_-(y, x) \neq []\}$ is the set of inhibitors of $x$.* $\square$

**Example 2** *For example 1 of Figure 1, $R_+(x) = \emptyset$, $R_-(x) = \{y\}$, $R_+(y) = \{z, x\}$, $R_-(y) = \emptyset$ and $R_+(z) = \emptyset$, $R_-(z) = \emptyset$. In example 2 $R_+(x) = \emptyset$, $R_-(x) = \{y\}$ and $R_+(y) = \{x\}$, $R_-(y) = \{x\}$.*

# 3   Formal Semantics of Biological Regulatory Networks

In René Thomas' theory, the evolution of the expression levels of the genes is described by an original notion of "attractor". Informally, it represents an upper or a lower bound which is attained if no change occurs in the rest of the BRN. Hence, the computation of the evolution of concentrations is based

4

on the attractors. They are defined by a set of parameters. The evolution of the BRN highly depends on the choice of those parameters.

In this section, we formalize the evolution of the states of a BRN by a transition system. This semantics also involves some *evolution parameters* as defined in [KT87].

## 3.1 State Space of a BRN

**Definition 3 (State Space of a BRN.)** *Let $\mathcal{R} = (V, E, \pi)$ be a BRN. The state space $S_x$ of a variable $x \in V$ is defined by $S_x = [0, \max_{y \in V} \pi_+(x, y) \cup \pi_-(x, y)\,]$. The state space of $\mathcal{R}$ is defined by $S(\mathcal{R}) = \times_{x \in V} S_x$. A state of the network $\mathcal{R}$ is a mapping $\nu : V \to \mathbb{N}$ such that $\forall x \in V, \nu(x) \in S_x$.* □

The previous definition of the set $S_x$ requires that 0 belong to the state space.

**Example 3** *For the example 2 of Figure 1, $S_x = [0, 2], S_y = [0, 1]$.*

## 3.2 Parameters of a BRN

As already mentioned in the beginning of this section, the behavior of a BRN depends on some *parameters*. Those parameters play the role of attractors and give the expression levels towards which a gene is attracted, depending on which genes activate or inhibit it.

**Definition 4 (Parameters of a BRN.)** *Let $\mathcal{R} = (V, E, \pi)$ be a BRN. The set $Para(\mathcal{R})$ of parameters of $\mathcal{R}$ is defined by*

$$Para(\mathcal{R}) = \{K_{x,A,B} | A \subseteq R_+(x), B \subseteq R_-(x)\}$$

*A valuation of the parameters $Para(\mathcal{R})$ is a mapping $\kappa : Para(\mathcal{R}) \to \mathbb{N}$ such that $\forall x \in V, A \subseteq R_+(x), B \subseteq R_-(x), \kappa(K_{x,A,B}) \in S_x$. In the sequel we use $K_{x,U,V}$ instead of $\kappa(K_{x,U,V})$ when the meaning is clear from the context.* □

**Definition 5 (Activity Assumption.)** *Let $\mathcal{R} = (V, E, \pi)$ be a BRN and $\kappa$ a valuation for $Para(\mathcal{R})$. $\kappa$ satisfies the* activity assumption *iff $\forall x \in V$:*

$\forall y \in R_+(x), \exists X_+ \subseteq R_+(x), \exists X_- \subseteq R_-(x), \kappa(K_{x,X_+ \cup \{y\}, X_-}) > \kappa(K_{x,X_+,X_-})$
$\forall y \in R_-(x), \exists X_+ \subseteq R_+(x), \exists X_- \subseteq R_-(x), \kappa(K_{x,X_+, X_- \cup \{y\}}) < \kappa(K_{x,X_+,X_-})$

□

**Definition 6 (Monotonicity Assumption.)** *Let $\mathcal{R} = (V, E, \pi)$ be a BRN and $\kappa$ be a valuation for Para$(\mathcal{R})$. $\kappa$ satisfies the* monotonicity assumption *iff: $\forall X_+ \subseteq R_+(x)$, $\forall X_- \subseteq R_-(x)$, $\forall X'_+ \subseteq R_+(x)$, $\forall X'_- \subseteq R_-(x)$, $X_+ \subseteq X'_+, X'_- \subseteq X_- \Rightarrow \kappa(K_{x,X_+,X_-}) \leq \kappa(K_{x,X'_+,X'_-})$.* $\square$

The activity assumption stands for the observability of the action of a gene on another. Without this assumption it is possible that any combination of activators of a gene does not have any observable effect on the target gene because its level of expression would remain the same. It seems then quite obvious that any valuation of the parameters should satisfy this property. The monotonicity property is is a biological experimental fact, already pointed out by René Thomas. Anyway our framework does not rely on these assumptions motivated by biology.

**Example 4** *In the example 2 of Figure 1, a possible valuation of the parameters is: $K_{x,\emptyset,\emptyset} = 0, K_{x,\emptyset,y} = 1$ and $K_{y,\emptyset,\emptyset} = 0, K_{y,x,\emptyset} = 0, K_{y,z,\emptyset} = 0, K_{y,xz,\emptyset} = 1$ and $K_{z,\emptyset,\emptyset} = 0$. Notice that this valuation does not satisfy the activity property as $K_{x,\emptyset,\emptyset} \geq K_{x,\emptyset,y}$. If we take $K_{x,\emptyset,\emptyset} = 1, K_{x,\emptyset,y} = 0$ this property is satisfied.*

## 3.3 Transition System of a BRN

Let us consider a BRN $\mathcal{R} = (V, E, \pi)$. Following [TGL76, Tho78] the evolution of the state of the network depends (i) on the expression level of the genes (ii) on a set of parameters (see Def. 4). The expression level of a gene $x$ may either decrease or increase according to which other genes of the network activate or inhibit it. If $X_+$ and $X_-$ are respectively the set of genes that currently activate $x$ and the set that currently inhibit $x$, then the value of $x$ evolves towards the value defined by the parameter $K_{x,X_+,X_-}$. Which genes are currently activating or inhibiting $x$ is defined according to the levels given in the network (e.g. for example 2 of Figure 1, $x$ activates $y$ when its expression level is 1, inhibits $y$ when its expression level is 2, and has no effect on $y$ when it is 0.) We formally define the different configurations of a network according to the activators and inhibitors of a gene $x$ in Def. 7.

**Definition 7 (State constraints of a BRN.)** *For $x \in V$, $X_+ \subseteq R_+(x)$, $X_- \subseteq R_-(x)$, we define $A_{x,X_+}$, $I_{x,X_-}$ and $C_{x,X_+,X_-}$ by:*

- $A_{x,X_+} = \left( \bigwedge_{y \in X_+} (y \in \pi_+(y, x)) \right) \wedge \left( \bigwedge_{y \in R_+(x) \setminus X_+} (y \notin \pi_+(y, x)) \right)$

- $I_{x,X_-} = \left( \bigwedge_{y \in X_-} (y \in \pi_-(y, x)) \right) \wedge \left( \bigwedge_{y \in R_-(x) \setminus X_-} (y \notin \pi_-(y, x)) \right)$

6

- $C_{x,X_+,X_-} = A_{x,X_+} \wedge I_{x,X_-}$      □

$C_{x,X_+,X_-}$ is true iff the values of the genes in $X_+$ are in the intervals in which they activate $x$ and the values of the genes in $X_-$ are in the intervals in which they inhibit $x$.

**Example 5** *For example 2 of Fig. 1 the activation and inhibition functions are: $\pi_+(x,y) = [1,1], \pi_-(x,y) = [2,2]$ and $\pi_+(y,x) = \emptyset, \pi_-(y,x) = [1,1]$. The activators and inhibitors sets are given by $R_+(y) = x, R_-(y) = x$ and $R_+(y) = \emptyset, R_-(x) = y$. The set of constraints are given by: $C_{x,\emptyset,\emptyset} = y \notin [1,1]$, $C_{y,x,\emptyset} = x \in [1,1] \wedge x \notin [2,2]$ and $C_{y,\emptyset,x} = x \notin [1,1] \wedge x \in [2,2]$.*

Another feature of the evolution of the state of a network is that the expression level of a gene $x$ evolves step-by-step i.e. it cannot go from 1 to 3 in a single step, it must evolve by one unit from 1 to 2 and if some conditions are met[1] will go from 2 to 3. This is captured in the definition of an *evolution operator*:

**Definition 8 (Evolution Operator ↗)** *Let $x \in \mathbb{N}$ and $k \in \mathbb{N}$. The evolution operator ↗ is defined by:*

$$x \nearrow k = \begin{cases} x - 1 & \textit{iff } x > k \\ x + 1 & \textit{iff } x < k \\ x & \textit{otherwise} \end{cases}$$

□

Notice that in the case $x = k$ the next value of $x$ will remain equal to $k$.

We can now define a transition system giving the semantics of a BRN.

**Definition 9 (Transition System of a BRN.)** *Let $\mathcal{R} = (V, E, \pi)$ be a BRN and $\kappa$ a valuation of the parameters in $Para(\mathcal{R})$. The semantics of $\mathcal{R}$ with valuation $\kappa$ is the labeled transition system $S_{\mathcal{R}}(\kappa) = (S(\mathcal{R}), V, \rightarrow)$ with $\rightarrow \subseteq S(\mathcal{R}) \times V \times S(\mathcal{R})$ such that:*

$$\nu \xrightarrow{x} \nu' \iff \begin{cases} & \exists A \subseteq R_+(x), \exists B \subseteq R_-(x), \nu \in [\![ C_{x,A,B} ]\!] \\ \wedge & \nu(x) \neq K_{x,A,B} \\ \wedge & \nu'(x) = x \nearrow K_{x,A,B} \\ \wedge & \forall y \neq x, \nu'(y) = \nu(y) \end{cases}$$

□

---

[1]it could be that from level 2 it is impossible to reach level 3.

**Remark 2** *Note that according to Def. 7, there is a unique $\nu'$ such that $\nu \xrightarrow{x} \nu'$. The transition system $S_\mathcal{R}(\kappa)$ is (partially) deterministic in the sense it is deterministic for each $x$-transition.*

*Nevertheless, there may be another $y$-transition from the state $\nu$ and thus $S_\mathcal{R}(\kappa)$ is not deterministic. The nondeterminism models the fact that the expression levels of the genes evolve asynchronously. Note also that there is an $x$-transition only when $x$ has not reached the value it tends to get closer to (i.e. $K_{x,A,B}$ for the right $A$ and $B$). This will enable us to define the stable states of a network as those states that have no outgoing transitions (the deadlock states).*

**Definition 10 (Stable State of a BRN.)** *Let $\mathcal{R} = (V, E, \pi)$ be a BRN and $\kappa$ a valuation of the parameters in $Para(\mathcal{R})$ and $S_\mathcal{R}(\kappa) = (S(\mathcal{R}), V, \rightarrow)$ its semantics. A state $\nu \in S(\mathcal{R})$ is non stable iff $\exists \nu' \in S(\mathcal{R}), x \in V$ such that $\nu \xrightarrow{x} \nu'$. A state $\nu$ is a stable state if it is not a non stable state (i.e. a stable state is a deadlock state).* $\qquad\square$

# 4 Simple Case-Study

We consider in this section the example 1 of Fig. 1. We use the verification tool Hytech [HHWT97] to automatically compute the results. Of course we could have chosen any model-checker to analyze our models but Hytech enables us to compute some constraints on the parameters such that certain properties are satisfied (we will not cover this in this paper.) The Hytech input files and results are given in appendix A.3.

The set of activators and inhibitors are given in Example 2, page 4. The state space is $S_x = [0, 1]$, $S_y = [0, 1]$ and $S_z = [0, 1]$. The parameters are $K_{x,\emptyset,\emptyset}$, $K_{x,\emptyset,y}$ (for $x$); $K_{y,\emptyset,\emptyset}$, $K_{y,x,\emptyset}$, $K_{y,z,\emptyset}$, $K_{y,\{x,z\},\emptyset}$ (for $y$) and $K_{z,\emptyset,\emptyset}$ (for $z$).

## 4.1 Example with Regular Stabilization

Let us fix the following values for the parameters:

$$
\begin{array}{ccccccc}
& & & K_{y,\emptyset,\emptyset} & = & 0 & \\
K_{x,\emptyset,\emptyset} & = & 1 & K_{y,x,\emptyset} & = & 0 & \\
K_{x,\emptyset,y} & = & 0 & K_{y,z,\emptyset} & = & 0 & K_{z,\emptyset,\emptyset} & = & 0 \\
& & & K_{y,\{x,z\},\emptyset} & = & 1 &
\end{array}
$$

The monotonicity and activity assumptions are satisfied by these parameters.

For this example we obtain the transition system given in Fig. 2 in the appendix A.1. The Hytech input file is given in appendix A.3. We can

easily compute the set of stable states and non stable states as given in the output file Figure 5, appendix A.3, 15. Note that the cycle reveals indeed an equilibrium state which is not stable.

## 4.2   Example without Regular Stabilization

Let us now fix the parameters to:

$$
\begin{array}{ccc}
& K_{y,\emptyset,\emptyset} &=& 0 \\
K_{x,\emptyset,\emptyset} = 1 & K_{y,x,\emptyset} &=& 1 \\
K_{x,\emptyset,y} = 0 & K_{y,z,\emptyset} &=& 1 \\
& K_{y,xz,\emptyset} &=& 1
\end{array}
\qquad K_{z,\emptyset,\emptyset} = 0
$$

The transition system obtained in this case is given in Fig. 3, appendix A.2. Again the results (Figure 6, appendix A.3, page 16) obtained with Hytech show that there is no regular stable state in this case. Note that the cycle is indeed a stable state, which is called *singular* in the R.Thomas'approach.

# 5   Conclusion and Future Work

In this paper we have given a formal semantics for Biological Regulatory Network. The main advantages of this work are (i) the formal semantics enables us to build automatically a (behavioral) model of a network (ii) this model can then be analyzed by verification tools exactly as safety critical programs can be (e.g. the formal semantics characterizes the stable and non stable states).

Our future work will consist in adding *timing constraints* in the network to build a more accurate model. Our semantics is ready to be extended with timing constraints: in this case we will derive a *timed* or *hybrid automata* model and use tools for analyzing this types of models to prove properties of the network.

## Acknowledgements

# References

[HHWT97]  Thomas A. Henzinger, Pei-Hsin Ho, and Howard Wong-Toi. HYTECH: A model checker for hybrid systems. *International Journal on Software Tools for Technology Transfer*, 1(1-2):110–122, 1997.

[Kit02]  H. Kitano. Looking beyond the details: a rise in system-oriented approaches in genetics and molecular biology. *Curr. Genet.*, 41(1):1–10, 2002.

[KT87]  M. Kaufman and R. Thomas. Model analysis of the bases of multistationarity in the humoral immune response. *J. Theor. Biol.*, 129(2):141–62, 1987.

[Sno89]  E.H Snoussi. Qualitative dynamics of a piecewise-linear differential equations : a discrete mapping approach. *Dynamics and stability of Systems*, 4:189–207, 1989.

[ST93]  E.H. Snoussi and R. Thomas. Logical identification of all steady states : the concept of feedback loop caracteristic states. *Bull. Math. Biol.*, 55(5):973–991, 1993.

[TGL76]  R. Thomas, A.M. Gathoye, and L. Lambert. A complex control circuit. regulation of immunity in temperate bacteriophages. *Eur. J. Biochem.*, 71(1):211–27, 1976.

[Tho78]  R. Thomas. Logical analysis of systems comprising feedback loops. *J. Theor. Biol.*, 73(4):631–56, 1978.

[TK01]  R. Thomas and M. Kaufman. Multistationarity, the basis of cell differentiation and memory. I. & II. *Chaos*, 11:170–195, 2001.

[Wol01]  O. Wolkenhauer. Systems biology: the reincarnation of systems theory applied in biology? *Brief Bioinform.*, 2(3):258–70, 2001.

# A  Appendix

## A.1  Transition System of the Example with Regular Stabilization



Figure 2: Transition system of the BRN of the example 1 of Fig. 1

We can notice (Fig. 2) that state (1,0,0) is stable. Moreover, we can observe a cycle (0,0,1) (1,0,1) (1,1,1) (0,1,1).

## A.2  Transition system of the Example without Regular Stabilization

We can notice (Fig. 3) that there is no regular stable state, but an attractile cycle: $(0,0,0) \xrightarrow{x} (1,0,0) \xrightarrow{y} (1,1,0) \xrightarrow{x} (0,1,0) \xrightarrow{y} (0,0,0)$ which can be considered as a singular stable state.
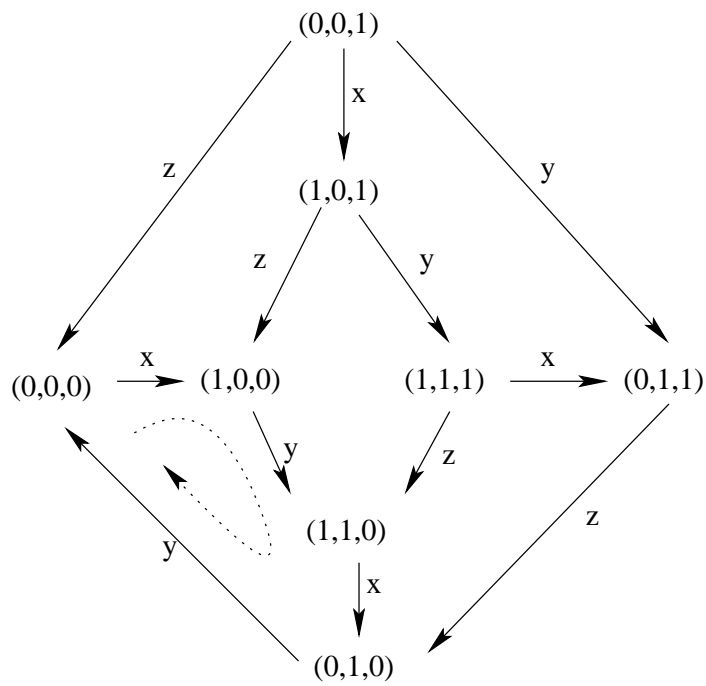
Figure 3: State graph of BRN of example 1

## A.3   Hytech Files

For the example of section 4, we use the input file of Fig. 4 to model our network.

The results obtained with the input file of Figure 4 is given in Figure 5.

```
-- hytech input file
var
      k_x_0_0,k_x_0_y,k_y_0_0,k_y_x_0,
      k_y_z_0,k_y_xz_0,k_z_0_0: parameter ; -- parameters
      x,y,z: discrete ;

      k: discrete ; -- k changes on every discrete transition
      k1,k2,k3: parameter ; -- used for detecting cycles


automaton rrb
synclabs: ;
initially Start ;

loc Start:
      while x>=0 & y>=0 & z>=0 & x<=1 & y<=1 & z<=1 wait {}
      -- C_x,0,0 -> K_x_0_0
      when  y < 1 & x > k_x_0_0 do {x'=x-1,k'=1-k} goto Start;
      when  y < 1 & x < k_x_0_0 do {x'=x+1,k'=1-k} goto Start;
      -- C_x,0,y
      when y >= 1 & x < k_x_0_y do {x'=x+1,k'=1-k} goto Start;
      when y >= 1 & x > k_x_0_y do {x'=x-1,k'=1-k} goto Start;
      -- C_y,0,0 -> k_y_0_0
      when x<1 & z<1 & y < k_y_0_0 do {y'=y+1,k'=1-k} goto Start;
      when x<1 & z<1 & y > k_y_0_0 do {y'=y-1,k'=1-k} goto Start;
      -- C_y,x,0 -> k_y_x_0
      when x>=1 & z<1 & y < k_y_x_0 do {y'=y+1,k'=1-k} goto Start;
      when x>=1 & z<1 & y > k_y_x_0 do {y'=y-1,k'=1-k} goto Start;
      -- C_y,z,0
      when x<1 & z>=1 & y < k_y_z_0 do {y'=y+1,k'=1-k} goto Start;
      when x<1 & z>=1 & y > k_y_z_0 do {y'=y-1,k'=1-k} goto Start;
      -- C_y,xz,0
      when x>=1 & z>=1 & y < k_y_xz_0 do {y'=y+1,k'=1-k} goto Start;
      when x>=1 & z>=1 & y > k_y_xz_0 do {y'=y-1,k'=1-k} goto Start;
      -- C_z,0_0
      when z < k_z_0_0 do {z'=z+1,k'=1-k} goto Start;
      when z > k_z_0_0 do {z'=z-1,k'=1-k} goto Start;

end

var init_reg, f_reachable, stable_states, non_stable_states,
xy_f_reachable, cycle_states : region;


init_reg := loc[rrb]=Start & x>=0 & y>=0 & z>=0 & x<=1 & y<=1 & z<=1 &
      k_x_0_0=1 & k_x_0_y=0 & k_y_0_0=0 & k_y_x_0=1 & k_y_z_0=1 &
      k_y_xz_0=1 & k_z_0_0=0;

prints "initial values for the K_ parameters and x,y,z:" ;
print omit rrb locations hide k,k1,k2,k3 in init_reg endhide ;

-- compute the reachable set of states ... must be finite
-- even if there is a cycle
f_reachable := reach forward from init_reg endreach;
if empty(f_reachable)
      then prints "No reachable states ...";
      else
```

```
            prints "The reachable states are:";
            print  hide k_x_0_0,k_x_0_y,k_y_0_0,k_y_x_0,
                  k_y_z_0,k_y_xz_0,k_z_0_0,k in f_reachable endhide;
endif ;

-- compute the projection on x,y of f_reachable
xy_f_reachable := hide k_x_0_0,k_x_0_y,k_y_0_0,k_y_x_0,
      k_y_z_0,k_y_xz_0,k_z_0_0,k in f_reachable endhide;


-- compute the set of non stable states i.e. reachable states
-- with a successor
-- define the strict predecessor operator
-- here is a trick to do this with Hytech (otherwise Hytech computes
-- the set of predecessor of a set including the set itself)
-- hide k in pre(A & k=0) & k=1 endhide gives the strict predecessor
-- of A
non_stable_states :=  f_reachable &
            hide k in (pre(f_reachable & k=0) & k=1) endhide;

-- print the result
if empty(non_stable_states) then
            prints "No non stable states";
            else
            prints "the reachable non stable states are:";
            print hide k_x_0_0,k_x_0_y,k_y_0_0,k_y_x_0,
               k_y_z_0,k_y_xz_0,k_z_0_0,k in non_stable_states endhide;
endif ;

stable_states := f_reachable  & ~non_stable_states ;
if empty(stable_states)
            then prints "No stable states ... !!!";
            else
            prints "The reachable stable states are:";
            print hide k_x_0_0,k_x_0_y,k_y_0_0,k_y_x_0,
               k_y_z_0,k_y_xz_0,k_z_0_0,k in stable_states endhide;
endif ;

-- now look for cycles ... easy in hytech with hide
-- (existential quantification)
-- first we define the strict sucessor function
-- it is a post where k changes followed by a reach

cycle_states  := x=k1 & y=k2 & z=k3 & f_reachable  &
            reach forward from hide k_x_0_0,k_x_0_y,
               k_y_0_0,k_y_x_0,k_y_z_0,k_y_xz_0,k_z_0_0,k in
               (post(x=k1 & y=k2 & z=k3 & f_reachable & k=0) & k=1)
                endhide endreach;

-- print  the result
if empty(cycle_states)
            then prints "No infinite path in the system";
            else prints "There is a cycle in the system ! ...
                from any of these states:";
            print hide k_x_0_0,k_x_0_y,k_y_0_0,k_y_x_0,
            k_y_z_0,k_y_xz_0,k_z_0_0,k,k1,k2,k3 in cycle_states endhide;
endif ;
```

Figure 4: Hytech Specification of the BRN of Example 1 Figure 1.

```
initial values for the K_ parameters and x,y,z:
     k_x_0_0 = 1   & k_x_0_y = 0   & k_y_0_0 = 0
     & k_y_x_0 = 0   & k_y_z_0 = 0   & k_y_xz_0 = 1
     & k_z_0_0 = 0   & x <= 1   & y <= 1   & z >= 0
     & x >= 0   & y >= 0   & z <= 1
.Number of iterations required for reachability: 1

The reachable states are:
Location: Start
     x <= 1   & y >= 0   & z <= 1
   & x >= 0   & y <= 1   & z >= 0
the reachable non stable states are:
Location: Start
     x = 0   & y < 1   & z >= 0   & y >= 0   & z <= 1
|
     y = 1   & x = 1   & z <= 1   & z >= 0
|
     y = 1   & z < 1   & x < 1   & z >= 0   & x >= 0
|
     z = 1   & y >= 0   & x <= 1   & x >= 0   & y <= 1
The reachable stable states are:
Location: Start
     y < 1   & 0 < x   & z < 1   & y >= 0
     & x <= 1   & z >= 0
.......Number of iterations required for reachability: 7

There is a cycle in the system ! ... from any of these states:
Location: Start
     z = 1   & y = 0   & x = 1
|
     z = 1   & y = 1   & x = 0
|
     z = 1   & y = 1   & x = 1
|
     z = 1   & y = 0   & x = 0


==================================================================
Max memory used =     0 pages =        0 bytes =   0.00 MB
Time spent      =    57.24u +     6.24s =     63.48 sec total
==================================================================
```

Figure 5: Hytech results for example of Fig. 4 (with Stabilization)

```
initial values for the K_ parameters and x,y,z:
     k_x_0_0 = 1    & k_x_0_y = 0    & k_y_0_0 = 0    &
     k_y_x_0 = 1    & k_y_z_0 = 1    & k_y_xz_0 = 1    &
     k_z_0_0 = 0    & x <= 1    & y <= 1    & z >= 0    &
     x >= 0    & y >= 0    & z <= 1
.Number of iterations required for reachability: 1

The reachable states are:
Location: Start
     x <= 1    & y >= 0    & z <= 1    & x >= 0
     & y <= 1    & z >= 0
the reachable non stable states are:
Location: Start
     x = 0    & y < 1    & z >= 0    & y >= 0    & z <= 1
|
     y = 1    & x = 1    & z <= 1    & z >= 0
|
     y = 1    & z < 1    & x < 1    & z >= 0    & x >= 0
|
     y = 0    & x = 1    & z < 1    & z >= 0
|
     z = 1    & y >= 0    & x <= 1    & x >= 0    & y <= 1
The reachable stable states are:
Location: Start
     0 < y    & z < 1    & x <= 1    & 0 < x    & z >= 0    & y < 1
|
     y >= 0    & z < 1    & 0 < x    & x < 1    & z >= 0    & y < 1
.......Number of iterations required for reachability: 7

There is a cycle in the system ! ... from any of these states:
Location: Start
     x = 1    & y = 0    & z >= 0    & z < 1
|
     x = 0    & y = 1    & z >= 0    & z < 1
|
     x = 1    & y = 1    & z >= 0    & z < 1
|
     x = 0    & y = 0    & z >= 0    & z < 1
======================================================================
Max memory used =     0 pages =        0 bytes =   0.00 MB
Time spent      =      73.04u +     8.10s =     81.14 sec total
======================================================================
```

Figure 6: Hytech results for example of Fig. 4 (with no stabilization)