



HAL
open science

Delays in Biological Regulatory Networks

Jamil Ahmad, Adrien Richard, Gilles Bernot, Jean-Paul Comet, Olivier Roux

► **To cite this version:**

Jamil Ahmad, Adrien Richard, Gilles Bernot, Jean-Paul Comet, Olivier Roux. Delays in Biological Regulatory Networks. Proceedings of the 2nd International Workshop on Bioinformatics Research and Applications, 2006, United Kingdom. pp.887–894, 10.1007/11758525_118 . hal-00341972

HAL Id: hal-00341972

<https://hal.science/hal-00341972>

Submitted on 18 Jul 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Delays in Biological Regulatory Networks (BRN)

Jamil Ahmad¹, Adrien Richard², Gilles Bernot²,
Jean-Paul Comet² and Olivier Roux¹

¹IRCCyN UMR CNRS 6597

BP 92101, 1 rue de la Noë, 44321 Nantes Cedex 3, France

²Programme épigénomique and IBISC, Université d'Evry

Tour Evry 2, Boulevard François Mitterrand, 91025 Evry cedex, France

Abstract. In this article, we propose a refinement of the modeling of genetic regulatory networks based on the approach of René Thomas. The notion of delays of activation/inhibition are added in order to specify which variable is faster affected by a change of its regulators. The formalism of linear hybrid automata is well suited to allow such refinement. We then use HyTech for two purposes: (1) to find automatically all paths from a specified initial state to another one and (2) to synthesize constraints on the delay parameters in order to follow any specific path.

1 Introduction to Biological Regulatory Networks

Biologists often represent their knowledge on a biological system in terms of graphs[dJ02]. Biological regulatory networks (BRN) represent interactions between biological entities which can be genes or their products, proteins. For example, genetic regulatory networks are graphs where vertices represent genes or regulatory products (e.g. RNA, proteins) and edges represent interactions between them. These interactions are further directed (regulators are distinct from targets) and signed (+ for activation, – for inhibition).

It is now clear for researchers that the semantics of a biological regulatory system and more generally an interaction system, is encoded in the dynamics of the system and not only in the entities of this system. Biologists often use the previously described regulatory graphs as a basis for generating dynamical models using either continuous representation or discrete ones.

- In differential models the activity of each gene is represented by a concentration of the associated RNA or proteins x_i , and the evolutions of all concentrations $x = (x_i)_{i \in [1, n]}$ obey a differential equation system $dx/dt = f(x)$. Observation leads biologists to consider only highly non-linear models with some strong threshold effects. The derivation of the dynamics from the interaction graph is not trivial even if the type of each interaction is known, because a lot of parameters have to be inferred, and a tiny modification of a parameter can lead to a strong change in the dynamics.
- In discrete models, the threshold effects are highlighted and allow modellers to discretize the concentrations. The first approach has been based on drastic discretization since all genes can be either *on* (present) or *off*

(absent) [Tho78]. This boolean model has been generalized into a multi-valued model [Sno89,Tho91], in which logical identification of all steady states [ST93,DHL03] becomes possible. The dynamics of these networks are based on abstraction of continuous-time switching networks which are a special type of hybrid systems as studied in, for example, control theory. Such continuous-time switching networks have been used to model dynamics in, for example, the sporulation network of *Bacillus subtilis* [dJGB⁺04]. The derivation of the dynamics from the interaction graph remains difficult even if the number of possible models is now finite. Since the formalism consists essentially in the discretization of the continuous differential equation system, the state space is divided into set of domains representing the symbolic qualitative states of the network. The transitions between the different states depend on logical parameters that play the role of limits of the solutions of the differential equation system of each domain in the continuous space. These limits are sometimes called attractors or targets [BCRG04].

The modeling activities then focus on the determination of parameters of the model which lead to a dynamic coherent with the specification (formal translation of experimental facts). Formal verification is not possible in the general framework of differential equation systems. In [dJGHP03] authors focus on a particular discrete model and use model checking in order to verify if the temporal properties are satisfied. Bernot et al. [BCRG04] proposed to consider all possible parameterizations, to generate all possible dynamics, to call for each of them a model checker for verification and to select only models which lead to a dynamic coherent with the specification. The enormous number of models limits this brute force approach.

One can also notice that the transition systems obtained in the formalism of R. Thomas [Tho78] or of H. de Jong [dJGHP03] are not deterministic: they abstract all possible continuous trajectories but they introduce some traces which do not correspond to continuous ones. This is due to a complete and total abstraction of time. To overcome this point Adélaïde and Sutre [AS04] showed that under some conditions of equality of degradation constants, this abstraction can lead to a dynamic which does not present the same drawback.

In this article we propose to take into account activation and inhibition delays in the formalism of R. Thomas following [TK01] where delays have been introduced to study traces closer to the experiment facts. After having briefly presented R. Thomas modelling in section 2, we introduce in section 3 the refinement based on delays. The example presented in section 4 allows us to present an algorithm for searching paths between two specified states (section 5). Finally we show how this algorithm can be helpful for parameter synthesis (section 6). The section 7 is devoted to conclusion.

2 Modeling of R. Thomas

In a directed graph $G = (V, A)$, we note $G^-(v)$ and $G^+(v)$ the set of predecessors and successors of a node $v \in V$ respectively.

Definition 1. A biological regulatory network, or BRN for short, is a tuple $\mathcal{G} = (V, A, l, s, t, K)$ where

- (V, A) is a directed graph denoted by G ,
- l is a function from V to \mathbb{N} with $l(v) > 0$ if $G^+(v) \neq \{\}$,
- s is a function from A to $\{+, -\}$,
- t is a function from A to \mathbb{N} such that $\{t(u, v) | v \in G^+(u)\} = \{1, \dots, l(u)\}$,
- $K = \{K_v | v \in V\}$ is a set of maps: for each $v \in V$, K_v is a function from $2^{G^-(v)}$ to $\{0, \dots, l(v)\}$ such that $K_v(\omega) \leq K_v(\omega')$ for all $\omega \subseteq \omega' \subseteq G^-(v)$.

The map l describes the domain of each variable v : if $l(v) = k$, the abstract concentration on v holds its value in $[0, 1..k]$. Similarly, the map s represents the sign of the regulation (+ for an activation, - for an inhibition).

$t(u, v)$ is the threshold of the regulation: the regulation takes place iff the abstract concentration of u is above $t(u, v)$, in such a case the regulation is said active. The condition on these thresholds states that each abstract level of u plays a role in the set of regulations of v . For all $x \in [0..l(u) - 1[$, the set of active regulations of u , when the expression level of u is x , differs from the set when the expression level is $x + 1$.

Finally, the map K_v allows us to define what is the effect of a set of regulators on the specific target v . If this set is $\omega \subseteq G^-(v)$, then, the target v is subject to a set of regulations which makes it to evolve towards a particular level $K_v(\omega)$.

Definition 2 (States). A state μ of a BRN $\mathcal{G} = (V, A, l, s, t, K)$ is a function from V to \mathbb{N} such that $\mu(v) \in \{0, \dots, l(v)\}$ for all variable $v \in V$. We denote $E^{\mathcal{G}}$ the set of states of \mathcal{G} .

When $\mu(u) > t(u, v)$ and $s(u, v) = +$, we say that u is a resource of v since the activation takes place. Similarly when $\mu(u) < t(u, v)$ and $s(u, v) = -$, u is also a resource of v since the inhibition does not take place (the absence of the inhibition is treated as an activation).

Definition 3 (Resource function). Let $\mathcal{G} = (V, A, l, s, t, K)$ be a BRN. For each $v \in V$ we define the resource function $\omega_v : E^{\mathcal{G}} \rightarrow 2^{G^-(v)}$ by:

$$\omega_v(\mu) = \{u \in G^-(v) \mid (\mu(u) \geq t(u, v) \text{ and } s(u, v) = +) \text{ or } (\mu(u) < t(u, v) \text{ and } s(u, v) = -)\}.$$

As said before, at the state μ , $K_v(\omega_v(\mu))$ gives the level towards which the variable v tends to evolve. We consider three cases, (1) if $\mu(v) < K_v(\omega_v(\mu))$ then v can increase by one unit, (2) if $\mu(v) > K_v(\omega_v(\mu))$ then v can decrease by one unit and (3) if $\mu(v) = K_v(\omega_v(\mu))$ then v can not evolve. The state graph of BRN represents the set of the states that a BRN can adopt with transitions between them deduced from these rules.

Definition 4 (State graph). Let $\mathcal{G} = (V, A, b, s, t, K)$ be a BRN. The state graph of \mathcal{G} is a directed graph $\mathbf{G} = (E^{\mathcal{G}}, T)$ with $(\mu, \mu') \in T$ if there exists $v \in V$ such that :

$K_v(\omega_v(\mu)) \neq \mu(v)$ and $\mu'(v) = \mu(v) + \alpha_v(\mu)$ and $\mu(u) = \mu'(u), \forall u \in V \setminus \{v\}$ where $\alpha_v(\mu) = +1$ if $K_v(\omega_v(\mu)) > \mu(v)$ and -1 otherwise.

3 Refinement based on delays

In the semantics that we use, a state can have several successors, each of them corresponding to the evolution of the expression level of a unique gene (the dynamic is asynchronous). One way to overcome this indeterminism is to use timing constraints in terms of parameters with a path algorithm (section 5). These parameters represent time delays for the discrete changes in expression levels of a gene. To be more precise, when an order of activation/inhibition arrives, the biological machinery starts to increase/decrease the corresponding protein concentration, but this action takes time. These times are abstracted by these time delays. We use two types of time delays $d_v^+(x)$ and $d_v^-(x)$ to represent the time required to change the expression level of a gene v from an abstract level x to $x + 1$ and from the level x to $x - 1$ respectively as shown in figure 1.

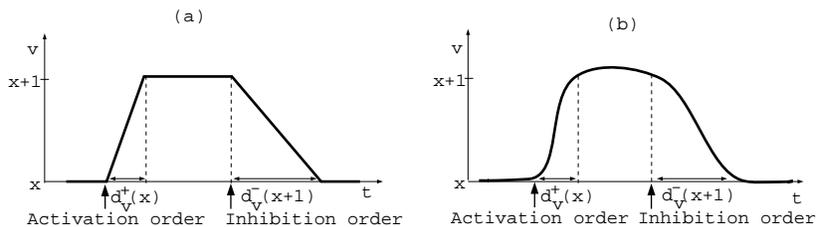


Fig. 1. Approximate evolution (a) of the actual evolution (b) of a gene's expression

4 Example of a BRN

Consider three genes a, b and c which interact according to the graph of figure 2-(a). From this BRN, we obtain the table of resources (figure 2-(c)) and state graph as shown in figure 2-(b).

In the table of figure 2, ω_v is the set of resources of a gene $v \in \{a, b, c\}$, that is the set of biological entities which helps variable v to increase. The evolution of the expression level of a gene v depends on its resources ω_v . Table of figure 2-(c) helps the reader to reconstruct the state graph representing the dynamics of the BRN. The edges in the graph give the possible transitions between states which can occur with time. The circuit $(010, 011, 001, 101, 100, 110, 010)$ represents an unstable circuit in the dynamics and the two states not involved in the circuit are the only two stable states.

We associate a clock h_v , to each variable $v \in \{a, b, c\}$. All the clocks of the system increase continuously and simultaneously¹. The guard $h_v == d_v^\alpha$, where

¹ It is possible to go from x to x' via the increasing (resp. decreasing) of variable v if the time delay d_v^+ (resp. d_v^-) is achieved by the clock associated to v .

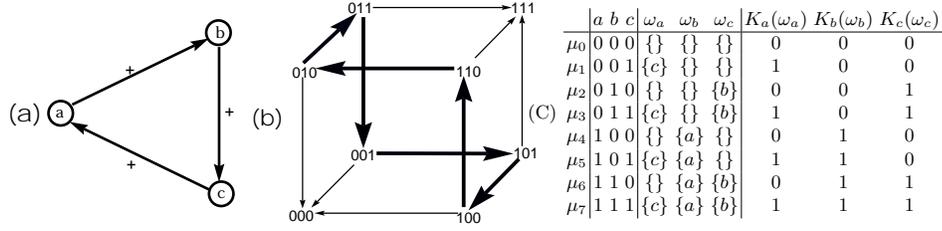


Fig. 2. BRN (a) and its state graph (b) according to the attractor parameters: $K_a(\{\}) = K_b(\{\}) = K_c(\{\}) = 0$ and $K_a(\{c\}) = K_b(\{a\}) = K_c(\{b\}) = 1$. (c) Associated table of resources

$\alpha \in \{+, -\}$ gives the condition for the transition. The clock of a variable is set to zero when this variable changes its abstract level. In figure 3 all the transitions are labeled with guards and clock initializations.

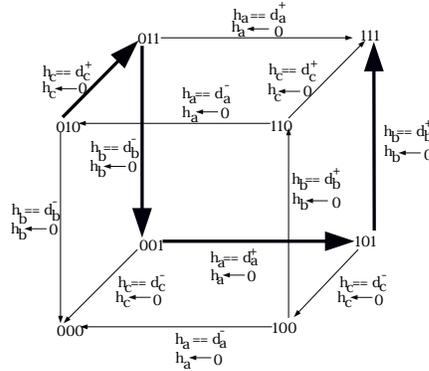


Fig. 3. Transition system for the BRN of figure 2 along with guard conditions

5 Searching paths between two states

The analysis of the hybrid refinement of the BRN is performed by using a linear hybrid model checker HyTech [HHWT97]. The delays are defined as parameters whose values are unknown. We associate timing constraints with qualitative states of a BRN in terms of time delay parameters. Our path algorithm finds all the possible paths between two qualitative states of a regulatory network. We have implemented this algorithm in HyTech and obtained the exact path between any two qualitative states of a BRN by applying timing constraints in terms of delay parameters.

Algorithm 1 is the pseudocode of the HyTech implementation. `pre` and `post` operators returns respectively the predecessors and successors of a state including the state itself. The difficulty of the algorithm lies in the fact that it converts the breadth-first search (induced by the `post` operator) into the depth-first search of a path. The algorithm consists of two main loops. In the outer loop the algorithm exhaustively searches the `final_state` from the `initial_state` and accumulates the accessed states in a set named `states_accumulated`. When the algorithm finds the `final_state` then it starts the nested loop and begins backward search from `final_state` and takes the intersection of each accessed states with the set `states_accumulated`. If the intersection is not empty then the algorithm gives the intersection as a state of the path which is accumulated in a set `path_states`. Finally the algorithm invokes the procedure `print_path(path_states)` to print the states of a path in proper order.

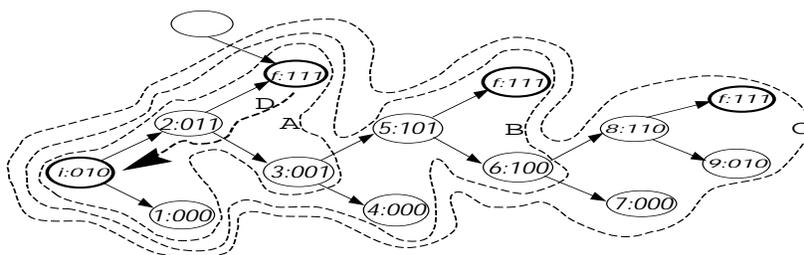


Fig. 4. How the algorithm finds the final states from initial state and vice versa. The empty state shows the accessibility of final state through other path

The dashed lines (A), (B) and (C) of figure 4 represent the successive sets of accumulated states when the algorithm finds the final state `f` during the outer loop. The inner loop is used for backward search and the dashed arrow (D) shows this search for the first path. Algorithm 1 finds the three paths between states (0,1,0) and (1,1,1) in the example of figure 3.

6 Parameters synthesis

The delay parameters used for the increasing and decreasing of the expression level of a gene v can be synthesized in HyTech to form timing constraints for each transition that takes place in paths between two states. The conjunction of constraints along any sequence of transitions gives the synthesized parameters constraint for the given path.

- For the transition (0,1,0) \rightarrow (0,1,1): $d_c^+ \leq d_b^-$
- For the transition (0,1,1) \rightarrow (0,0,1): $d_b^- \leq d_a^+$
- For the transition (0,0,1) \rightarrow (1,0,1): $d_c^+ + d_a^+ \leq d_a^- + d_b^- + d_c^-$
- For the transition (1,0,1) \rightarrow (1,1,1): $d_a^+ + d_b^+ \leq d_c^- \wedge d_a^+ + d_c^+ \leq d_a^- + d_b^-$

Algorithm 1 Finds paths between two states

```
1: Path(initial_state, final_state)
2: states_accessed := initial_state; // The first accessed states is the initial state
3: reached := initial_state; // The first visited state is the initial state
4: path := initial_state; // Path is set to initial state
5: states_accumulated := initial_state;
6: // while exist accessible states of a BRN from initial state
7: while not_empty(states_accessed) do
8:   // The set of accessed states is now the successors of the previously accessed states
9:   states_accessed := post(states_accessed) - states_accessed;
10:  path := states_accessed - path; // Find new states of a path
11:  states_accumulated := states_accumulated  $\cup$  path; // Set that accumulates the states
12:  states_accumulated := states_accumulated - initial_state; // Remove the initial state from
  the set
13:  states_accessed := states_accessed - reached; // Remove all previously visited states from set
  of accessed states
14:  reached := reached  $\cup$  states_accessed; // The previously visited states will now be the accessed
  states
15:  // Check if final state is accessed
16:  if not_empty(path  $\cap$  final_state) then
17:    states_accessed1 := final_state;
18:    reached1 := final_state;
19:    path1 := final_state;
20:    //The nested loop starts here
21:    while not_empty(states_accessed1) do
22:      // The set of accessed states is now the predecessors of the previously
23:      // accessed states in a set of accumulated states
24:      states_accessed1 := (pre(states_accessed1)-states_accessed1)  $\cap$  states_accumulated;
25:      path1 := states_accessed1 - path1;
26:      path_states := path_states  $\cup$  path1; // Set that accumulates the states of one discovered
      path
27:      states_accessed1 := states_accessed1 - reached1;
28:      reached1 := reached1  $\cup$  states_accessed1;
29:    end while
30:    print_path(path_states); // To print the states of a path
31:  end if
32: end while
```

Now if we desire to find only the path shown by the bold line in figure 3, then we use the timing constraints which are synthesized² by HyTech for the transitions in one path starting from a state where $h_a = h_b = h_c = 0$ (see below). Thus, we draw an equivalence between the path $(0, 1, 0) \rightarrow (0, 1, 1) \rightarrow (0, 0, 1) \rightarrow (1, 0, 1) \rightarrow (1, 1, 1)$ and the region described by the conjunction of constraints: $(d_c^+ \leq d_b^-) \wedge (d_b^- \leq d_a^+) \wedge (d_c^+ + d_a^+ \leq d_a^- + d_b^- + d_c^-) \wedge (d_a^+ + d_b^+ \leq d_c^-) \wedge (d_a^+ + d_c^+ \leq d_a^- + d_b^-)$.

7 Conclusion

We propose in this paper a refinement based on delays for BRN modelling. The introduction of delays allows one to distinguish paths from one state to another one. This refinement reintroduces time in the abstraction of R. Thomas, and this way is different from the refinement of [BRdJ⁺05] and [AS04] which split the state space by partitioning the domains of the state space. The present work describes how the introduction of time can be helpful for modeling such

² The interested readers can contact the authors for the HyTech codes and results.

networks, allowing the modeller to verify temporal properties. It is now crucial to confront this modelling with real experiments. Our experience in modeling in a multidisciplinary context will help to initiate biological modeling with delays.

Our future work is threefold: (1) formal representation of parameters constraints, (2) parameter synthesis: we have to check when a path is equivalent to an empty region and what this really means, and (3) cycles: we have to check if there are cycles in the state graph of a BRN.

References

- [AS04] M. Adélaïde and G. Sutre. Parametric analysis and abstraction of genetic regulatory networks. In *Proc. 2nd Workshop on Concurrent Models in Molecular Biology (BioCONCUR'04), London, UK, Aug. 2004*, Electronic Notes in Theor. Comp. Sci. Elsevier, 2004.
- [BCRG04] G. Bernot, J.-P. Comet, A. Richard, and J. Guespin. Application of formal methods to biological regulatory networks: Extending Thomas' asynchronous logical approach with temporal logic. *Journal of Theoretical Biology*, 229(3):339–347, 2004.
- [BRdJ⁺05] G. Batt, D. Ropers, H. de Jong, J. Geiselmann, M. Page, and D. Schneider. Qualitative analysis and verification of hybrid models of genetic regulatory networks: Nutritional stress response in *Escherichia coli*. In M. Morari and L. Thiele, editors, *Eighth International Workshop on Hybrid Systems: Computation and Control, HSCC 2005*, volume 3414 of *Lecture Notes in Computer Science*, pages 134–150. Springer, 2005.
- [DHL03] V. Devloo, P. Hansen, and M. Labbé. Identification of all steady states in large networks by logical analysis. *Bull. Math. Biol.*, 65(6):1025–51, 2003.
- [dJ02] H. de Jong. Modeling and simulation of genetic regulatory systems: a literature review. *J. Comput. Biol.*, 9(1):67–103., 2002.
- [dJGB⁺04] H. de Jong, J. Geiselmann, G. Batt, C. Hernandez, and M. Page. Qualitative simulation of the initiation of sporulation in *Bacillus subtilis*. *Bulletin of Mathematical Biology*, 66(2):261–299, 2004.
- [dJGHP03] H. de Jong, J. Geiselmann, C. Hernandez, and M. Page. Genetic network analyzer: qualitative simulation of genetic regulatory networks. *Bioinformatics*, 19(3):336–44., 2003.
- [HHWT97] T-A. Henzinger, P-H. Ho, and H. Wong-Toi. HYTECH: A model checker for hybrid systems. *International Journal on Software Tools for Technology Transfer*, 1(1–2):110–122, 1997.
- [Sno89] E.H. Snoussi. Qualitative dynamics of a piecewise-linear differential equations : a discrete mapping approach. *DSS*, 4:189–207, 1989.
- [ST93] E.H. Snoussi and R. Thomas. Logical identification of all steady states : the concept of feedback loop characteristic states. *Bull. Math. Biol.*, 55(5):973–991, 1993.
- [Tho78] R. Thomas. Logical analysis of systems comprising feedback loops. *J. Theor. Biol.*, 73(4):631–56, 1978.
- [Tho91] R. Thomas. Regulatory networks seen as asynchronous automata : A logical description. *J. theor. Biol.*, 153:1–23, 1991.
- [TK01] R. Thomas and M. Kaufman. Multistationarity, the basis of cell differentiation and memory. *Chaos*, 11:180–195, 2001.