

Verification of biological models with Timed Hybrid Petri Nets

S. Troncale*, J.-P. Comet[†] and G. Bernot[†]

**IBISC, Evry France*

[†]*Epigenomic project, Evry France*

Abstract. The formalism of Hybrid Functional Petri Nets (HFPN) has proved its convenience for simulating biological systems. The drawback of the noticeable expressiveness of HFPN is the difficulty to perform formal verifications of dynamical properties. In this article, we propose a model-checking procedure for Timed Hybrid Petri Nets (THPN), a sub-class of HFPN. This procedure is based on the translation of the THPN model and of the studied property into real-time automata. It is applied to a sub-network involved in amphibian metamorphosis.

Keywords: THPN, model-checking, verification, amphibian metamorphosis

PACS: 01.30.Cc

INTRODUCTION

System biology aims to the understanding of biological systems taking into account molecular processes, signal transduction, interactions between sub-cellular compartments... For this goal researchers are interested in modelling of biological systems. A lot of modelling frameworks focus on the simulation of biological systems: local evolution laws are given in a strong expressive way and simulations are used to make emerge some complex behavioral properties of the global system. For example, Hybrid Functional Petri Nets (HFPN) [1] allow the representation in a unified way of discrete and continuous entities (*e.g.* number of cells or molecular concentration) as well as discrete and continuous processes (*e.g.* mitosis is an example of discrete event although evolution of concentrations due to enzymatic reactions exemplifies a continuous one). The functional aspects of HFPN allow the definition of biological functions in terms of mathematical functions associated with processes, leading to a more strong expressiveness of this formalism. At the opposite, other approaches focus on the capability of checking whether the *in silico* model satisfies properties of the biological system (model checking): only a model which is coherent with the known biological behavioral properties is recognized as a relevant model.

In this paper, we propose to make a compromise between expressiveness and verification. We choose to restrict the expressiveness of HFPN by removing its functional aspects because they induce some implicit use of system states. For this obtained sub-class of HFPN, called Timed Hybrid Petri Nets (THPN) [2], we developed a model checking procedure based on the translation of the THPN model and of the studied property into real time automata. Depending on whether there exist common executions between both automata, we conclude whether the THPN model satisfies or not the studied property. We apply our model checking procedure on a sub-network involved in the amphibian metamorphosis. This biological application, performed in collaboration with a team of developmental biologists, leads to new crucial biological questions.

The paper is organized as follows: Section 2 presents the studied biological system. Section 3 recalls major definitions and properties of THPN. Sections 4 and 5 present the conversion into real-time automata of the THPN model and the property respectively. Finally, intuitions of the

procedure of final decisions are given in Section 6 before we discuss our results in Section 7.

BIOLOGICAL CONTEXT

Most amphibians undergo numerous morphological changes at the tadpole stage, a biological process called metamorphosis. Amphibian metamorphosis can be divided into three periods. During premetamorphosis the feeding tadpole grows. During prometamorphosis hind limbs grow and differentiate. Finally, tail resorption characterizes metamorphic climax. All metamorphosis-associated morphological changes are under the control of thyroid hormone, denoted TH. It is relevant to distinguish two molecular forms of TH [3]. Thyroxine (tetraiodothyronine, TH4) corresponds to the major form secreted by the thyroid gland, it is an “inactive” form of TH and is considered a pro-hormone. Triiodothyronine (TH3) is the biologically active form but it is secreted in a smaller quantity [4].

The TH-dependant regulation at the origin of tail resorption requires a strong intra-cellular TH concentration [3]. This regulation is triggered at the climax of metamorphosis when plasmatic TH concentration is maximal. Intra-cellular TH bind their nuclear receptors, noted TR [4]. Complexes TH3/TR are transcription factors inducing in a first time, a set of early genes and in a second time, a set of late genes responsible for apoptosis.

The intra-cellular TH3 concentration depends on the plasmatic contribution, but it is also regulated by two enzymes, type 2 deiodinase, D2 [5] and type 3 deiodinase D3 [6]. Type 2 deiodinase synthesizes the active form of the hormone (TH3) from its inactive form (TH4) : $D2 + TH4 \rightarrow TH3$. On the contrary, type 3 deiodinase inactivates the two hormone types TH3 and TH4 : $D3 + TH4 \rightarrow rTH3$ and $D3 + TH3 \rightarrow T2$, where rTH3 and T2 are inactive. The gene expressing D3 is a direct-response gene of TH [6]. Expression of gene encoding D2 is also regulated by thyroid hormone, but in an indirect manner. D2 concentration then increases with the one of TH3, but response time is much longer (few days) [5].

In a previous work [7], a HFPN model representing kinetics of different factors which interfere in tail resorption was constructed. The main goal of such a modelling work was to lead to a better understanding of the biological system. In our case, the model was constructed in the goal of going into detail our knowledge about specific functionalities of thyroid hormone in the tail. The initial question refers to the TH nature: can we talk about morphogen for thyroid hormone? Recall that a morphogen is a secreted molecule which has the property to induce different cellular types with different concentrations.

Nevertheless, model construction leads to expose new pertinent questions. Indeed, it is easy to remark that TH3 is undergone to two retroaction loops: a positive retroaction loop with D2 and a negative retroaction loop with D3. It is commonly thought that the intra-cellular concentration of TH3 becomes sufficient to induce tail resorption thanks to D2 action. Nevertheless, the kinetics data (Km and Vmax) given by Germain *et al.* in [8] show that D2 is 10 times less active than D3. We then ask us on matter of D2 role. Is D2 sufficient to induce an enough strong concentration of TH3? To answer this question, we model in this paper competition existing between the two loops. Thus, the asked question can be informally exposed as follows: “At the climax of metamorphosis, when the concentration of TH is maximal, does always the negative loop overcome the positive loop ?” Our model-checking procedure allows to verify whether the model satisfies the previous biological property.

TIMED HYBRID PETRI NETS

The competitive regulation between D2 and D3 is modelled by a Timed Hybrid Petri Net. Evolution of a THPN can then be studied by constructing its evolution graph. Let us first recall the definition of THPN models and of evolution graphs [2].

Definition 1 A *Timed Hybrid Petri Net* is a 8-tuple $(\mathcal{P}, \mathcal{T}, \zeta, Pre, Post, m_0, delay, V)$ where:

- \mathcal{P} and \mathcal{T} are disjoint sets of places and transitions,
- $\zeta : \mathcal{P} \cup \mathcal{T} \rightarrow \{D, C\}$ called “hybrid function,” indicates for every node whether it is a discrete node or a continuous one.
Let T^D (resp. P^D) and T^C (resp. P^C) be the sets of discrete and continuous transitions (resp. places),
- $Pre : \mathcal{P} \times \mathcal{T} \rightarrow \mathbb{R}^+ \cup \mathbb{N}$ is the input incidence application. If $T \in T^D$ then $Pre(P, T) \in \mathbb{N}$ else $Pre(P, T) \in \mathbb{R}^+$,
- $Post : \mathcal{T} \times \mathcal{P} \rightarrow \mathbb{R}^+ \cup \mathbb{N}$ is the output incidence application. If $T \in T^D$ then $Post(T, P) \in \mathbb{N}$ else $Post(T, P) \in \mathbb{R}^+$,
- $m_0 : \mathcal{P} \rightarrow \mathbb{R}^+ \cup \mathbb{N}$ is the initial marking. If $P \in P^D$ then $m_0(P) \in \mathbb{N}$ else $m_0(P) \in \mathbb{R}^+$,
- $delay : T^D \rightarrow \mathbb{R}^+$ is a timing associated with each discrete transition,
- $V : T^C \rightarrow \mathbb{R}^+$ represents the maximal firing speed associated with each continuous transition.

THPN places model biological entities such as enzymes or hormones, and transitions model biological processes such as complex formation or enzymatic reactions. A discrete place is used if the modelled biological entity is numbered (number of a cell population, for example). At the opposite, a continuous place measures a real quantity such as a molecular concentration. Discrete processes occur at some given times (use of delays on discrete transitions), although continuous processes continuously occur at some speed (use of speed on continuous transitions). Thus, mitosis is an example of discrete event although evolution of concentrations due to enzymatic reactions exemplifies a continuous one.

Biological example: THPN. Since molecular concentrations (real values) are used for TH3, TH4, D2 and D3, these entities are modelled by continuous places (Figure 1). Initially, we consider no D2 and no D3 and we consider a factor 5 between the concentrations of TH4 and TH3 [5]. D3 is a direct-response gene of TH3. Its induction by the hormone is modelled by a continuous transition (T_2). At the opposite, D2 is induced in an indirect manner, what leads to a delay between TH3 induction and D2 synthesis. This indirect induction is modelled by the discrete transition T_1 . The continuous transition T_3 models D3 activity by removing TH3 and TH4 and the continuous transition T_4 models D2 action by synthesizing TH3 from TH4. Note that TH3, D2 and D3 are not consumed during respectively, enzyme induction and enzyme action, consequently a bidirectional arc is used.

Semantics intuition. Definition 1 gives a syntactic definition of THPN. So as to understand dynamics of THPN, let us describe their semantics.

- A discrete transition T is *enabled* if each place P_i preceding T satisfies $m(P_i) \geq Pre(P_i, T)$, where $m(P_i)$ is the current marking of the place P_i .
(Figure 1: T_1 is enabled if the place TH3 reaches the threshold 1)
- If the transition T stays enabled during the time $delay(T)$, it will be fired at the end of this delay. $Pre(P_i, T)$ tokens are then removed from each place P_i and $Post(T, P_j)$ tokens are added to each place P_j succeeding to T .

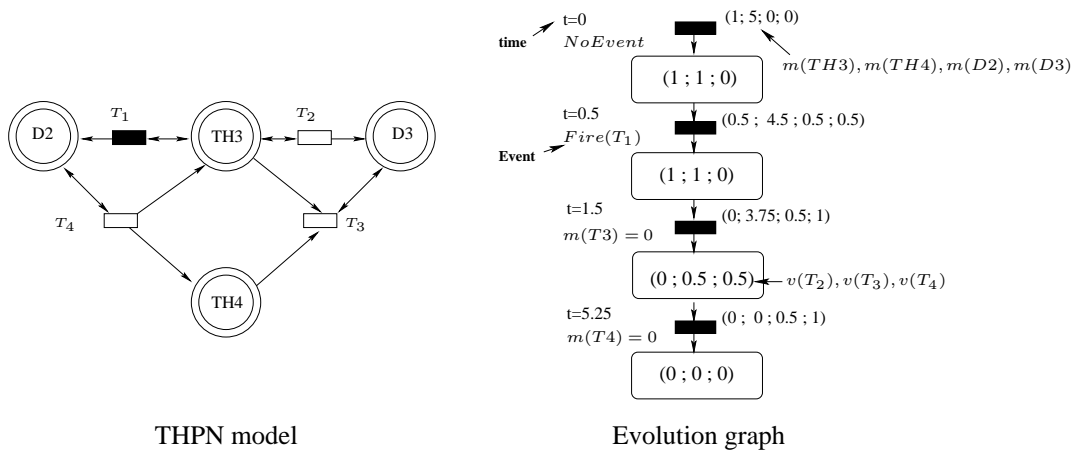


FIGURE 1. THPN of competitive regulation between D2 and D3 and its evolution graph.

(Figure 1: firing of T_1 induces the production of one token in D2 and TH3, as well as the consumption of one token in TH3)

- A continuous transition T is *enabled* if each place P_i preceding T satisfies $m(P_i) > 0$.
(Figure 1: T_3 is enabled if places TH3 and TH4 are not empty)
- A continuous transition is fired to its *instantaneous firing speed* $v(T)$ such that $0 \leq v(T) \leq V(T)$. $v(T)$ corresponds to the maximal speed a transition can fire according to the current marking. A flow of $Pre(P_i, T) \times v(T)$ tokens are removed from each place P_i preceding T and a flow of $Post(T, P_j) \times v(T)$ tokens are added to each transition P_j succeeding to T .

Evolution graph. The behavior of a THPN can be represented by an *evolution graph* (EG), represented by a Petri net [2]. Each place corresponds to an IB-state (invariant behavior state) and each transition is associated with an event (change of marking) whose occurrence produces a change of IB-state. Such a transition can only occur if an event belonging to one of the following types takes place: the marking of a continuous place becomes zero, a discrete transition fires or a discrete transition becomes enabled because of a change of continuous place marking.

Intuitively, a transition of the evolution graph (denoted T^{EG}) is labelled with the set of occurred events, with time of the event occurrence and with marking of all continuous places. IB-states are annotated by marking of all discrete transitions and by the vector of instantaneous speeds (see Figure 1).

For constructing such an evolution graph, a restriction is imposed to THPN [2]. Since the evolution graph represents a deterministic behavior, one has to solve conflicts which occur when the marking of a place is not sufficient to allow the different transitions to fire simultaneously. Generally, conflicts are solved by *priority*, by ranking involved transitions.

Biological example: EG. The place TH3 (Figure 1) must fire three different transitions: T_1 , T_2 and T_3 . If the marking of TH3 is not sufficient to allow these three firings, a conflict exists. Different resolution rules can be given: T_1 has priority then T_2 and T_3 , T_2 has priority,... Figure 1 gives the evolution graph of the THPN model when the transition T_3 has priority. For pedagogical reasons, only one conflict resolution is handled. Nevertheless, our method handles all conflict resolutions. An evolution graph with several branches is then obtained, each branch corresponds to a conflict resolution.

EVENT CLOCK AUTOMATON OF THE THPN

Each branch of evolution graph gives an *infinite* timed execution, called a timed trace of THPN. The set of timed traces of a THPN is represented by a real-time automaton. Thus, for studying a THPN, we have first to convert timed traces of evolution graph into this real-time automaton, called Event Clock automaton [9]. After definition of an Event Clock automaton (ECA), our algorithm allowing the translation of an evolution graph into an ECA is illustrated on the evolution graph of the Figure 1.

A timed trace of a THPN inform us about the value of each variable (marking, speed,...) at some given times. Observations of THPN variables are called *atoms*. For example, the following atoms $Fire(T_1)$, $m(TH4) \geq 2.0$ and $NoEvent$ respectively mean that the discrete transition T_1 is fired, the marking of the place $TH4$ is greater than 2.0 and no specific event occurred. The Event Clock automaton associated with the evolution graph models succession of atoms in the course of THPN execution. This ECA is timed thanks to the use of two clocks, the history clock x_α and the prophecy clock y_α [9], where α is an atom. The value of a history clock x_α is the time elapsed since the last occurrence of α . The value of a prophecy clock y_α is the time to wait for the next occurrence of α . For example, the *clock constraint* $y_p \leq 3$ means that p will be true in at most 3 time units and $x_q = 4$ means that q was observed 4 time units ago.

Definition 2 *An Event Clock automaton is a 6-tuple $A = (L, L_0, At, C, E, \mathcal{F})$ where:*

- L is a finite set of locations and $L_0 \subseteq L$ is the subset of start locations,
- At is a set of atoms,
- C is a set of history or prophecy clocks,
- E is a finite set of edges. An edge is a 4-tuple (l_1, ψ, l_2, w) where $l_1 \in L$ is the source location, $l_2 \in L$ is the target location, ψ is a set of atoms describing the location l_1 , and w is a clock constraint,
- $\mathcal{F} = \{F_1, \dots, F_n\}$ where $F_i \subseteq L$, is a set of sets of accepting locations

Algorithm. Each transition of the evolution graph gives a *location* of the ECA, called A_M . Edges outgoing from these locations are *labelled by atoms describing the evolution graph transitions*: marking of each place, speed value of each continuous transition or the set of occurred events. Moreover, time is associated with transitions of the EG. Two successive transitions (T_1^{EG} and T_2^{EG}) respectively labelled by times t_1 and t_2 are then separated by $t_2 - t_1 = \Delta t$. This delay labels ECA edges thanks to history clocks.

Thus, the first transition of the EG in Figure 1 gives the location l_0 in A_M (Figure 2). The edge outgoing from this location is labelled by the following atoms: $NoEvent$, $m(TH3) = 1, v(T_3) = 1, \dots$

Each IB-state gives also a *location* in A_M . Atoms associated with IB-state stay true during all the time Δt . Execution then loops on these locations. Edges outgoing from these locations are *labelled by these atoms*.

The first IB-state (Figure 2) gives the location l_1 . One of the two edges outgoing from l_1 loops. They are labelled by the following atoms: $v(T_3) = 1, v(T_4) = 0, \dots$

The *start location* is the location corresponding to the first transition of evolution graph (l_0). The *set of sets of accepting locations* is reduced to singletons containing only the locations corresponding to the last IB-states (l_8).

Biological example. The Event Clock automaton A_M resulting from the evolution graph of the Figure 1 is partially shown in the Figure 2.

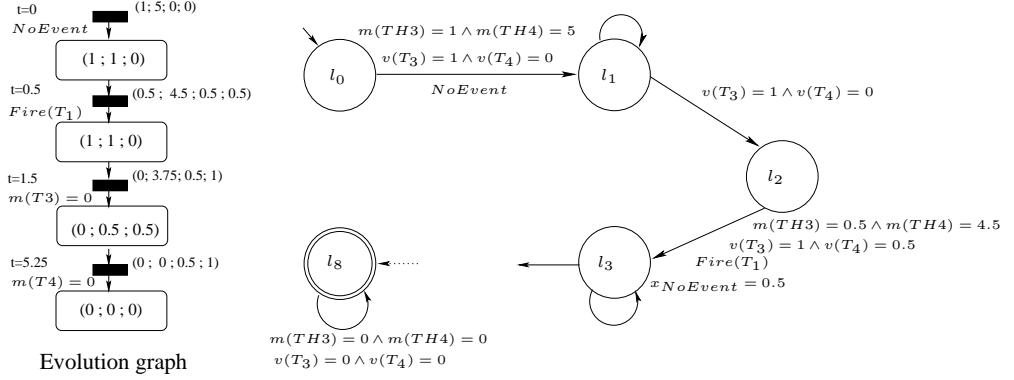


FIGURE 2. Event Clock automaton A_M . To simplify, atoms labelling edges only consider marking of places TH3 and TH4 and speed values of the continuous transitions T_3 and T_4 .

STUDY OF A PROPERTY

Timed traces of a THPN integrate notions of real time. The studied property is then expressed into a real-time logic, we called Continuous Time Evolution Logic (CTEL). Definition 3 presents syntactic rules of a CTEL formula.

Definition 3 Following [10], a CTEL well formed formula is composed of atoms, connectives $\neg, \vee, \wedge, \Rightarrow$, temporal operators *Next* (\circ), *Previous* (\ominus), *Until* (U) and *Since* (S) and of real-time operators: predicting and history operators ($\triangleright, \triangleleft$):

$$\varphi ::= a \parallel \neg\varphi \parallel \circ\varphi \parallel \ominus\varphi \parallel \varphi_1 \wedge \varphi_2 \parallel \varphi_1 \vee \varphi_2 \parallel \varphi_1 \Rightarrow \varphi_2 \parallel \varphi_1 U \varphi_2 \parallel \varphi_1 S \varphi_2 \parallel \triangleleft_{\sim n} \alpha \parallel \triangleright_{\sim n} \alpha,$$

where α is an atom, $\sim \in \{=, <, >, \leq, \geq\}$, $\varphi, \varphi_1, \varphi_2$ are formulas and $n \in \mathbb{N}$.

Property ECA. The studied property translates a set of admissible behaviors of the system. Consequently, a property expressed in CTEL translates a set of timed traces which can also be translated into an Event Clock automaton. If we study the property f , the Event Clock automaton $A_{\neg f}$ of the negation of f is constructed. $A_{\neg f}$ represents the set of *all* timed traces of $\neg f$. We can then search whether there exists a trace common to A_M and $A_{\neg f}$. If such a trace exists, we conclude that our THPN model does not satisfy f since there exists a trace satisfying $\neg f$.

Raskin and Schobbens proposed an algorithm to construct an Event Clock automaton from a property [10]. This algorithm (not detailed here) is *complete*, a timed trace satisfying $\neg f$ is then necessarily represented by the automaton.

Biological example. Our THPN model simulates competition existing between the two retroaction loops: the positive loop with D2 and the negative loop with D3. We ask us whether in all cases of conflict resolutions, the negative loop overcomes the positive loop. This property is translated into a CTEL formula f : $f \equiv \square \diamond (m(TH3) = 0)$ where \diamond means eventually and \square means globally. Both operators \diamond and \square correspond to the following abbreviations: let ϕ be a CTEL formula, then $\diamond\phi \equiv True U \phi$ and $\square\phi \equiv \neg \diamond \neg\phi$.

The formula means that the concentration of TH3 finally reaches zero, what translates the predominance of the negative loop. Figure 3 shows the Event Clock automaton of $\neg f \equiv \neg \square \diamond (m(TH3) = 0)$.

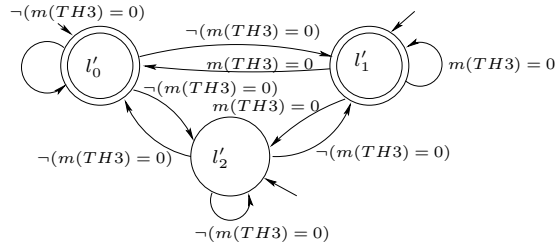


FIGURE 3. Event Clock automaton A_{-f} .

Model-checking versus simulation. Satisfiability of the studied property could be tested by simulation, seeing whether TH3 goes to 0 over time. Nevertheless, two major reasons argue in favour of model-checking procedure. On one hand, ECA model *infinite* timed traces, although simulation shows THPN behavior during a finite time. Consequently, we cannot be sure that the model reaches a stable state.

On the other hand, simulation shows a unique timed trace (conflicts are not explicitly treated), although our model-checking procedure handles timed traces corresponding to all conflict resolutions. Consequently, a conflict resolution could disagree with the property, whereas the simulation trace could agree with it.

FINAL DECISION

As usual, to extract common traces between the two automata A_M and A_{-f} , we need to perform the *product* of these automata and to decide whether the set of timed traces of the product automaton, noted A_p , is empty (*emptiness decision*). In our case, a specific product integrating quantitative notions is required. We then define a product, called the *semantic product* and we extended the usual emptiness decision procedure [10] deciding whether the set of traces is empty or not (see [11] for more details).

The semantic product consists in making the conjunction of atoms and of clock constraints. If there is an edge in A_M such that $(l_1, m(TH4) = 0, l_2, True)$ and if there is an edge in A_{-f} such that $(l'_1, m(TH4) \geq 4, l'_2, x_{NoEvent} = 3)$ then $((l_1, l'_1), m(TH4) = 0 \wedge m(TH4) \geq 4, (l_2, l'_2), x_{NoEvent} = 3)$ is an edge of the product automaton A_p . Our emptiness decision procedure has then been extended so as to remove all inconsistent edges. Then, the edge given below is removed since there is no possible value of $m(TH4)$ which can simultaneously satisfy $m(TH4) \geq 4$ and $m(TH4) = 0$. Finally, if no trace is read on the product automaton, we say that the language of the product automaton is empty.

Biological example. Considering the timed traces of A_M (Figure 2), it is easy to see that at a moment, the concentration of TH3 reaches zero. When the procedure is extended to all timed traces obtained from all conflict resolutions, the emptiness decision procedure shows that the concentration of TH3 eventually reaches zero.

DISCUSSION

From the point of view of our biological example, we show in this paper that when the concurrent actions of D2 and D3 on TH are modelled, the negative loop always overcomes the positive loop. These conclusions are in contradiction with the usual assumption that the intra-

cellular concentration of TH3 becomes sufficient to induce apoptosis thanks to D2 [5]. Then new questions arise: “Are, in a same cell, D2 and D3 co-expressed ? If the two enzymes are expressed in the same cells, what is really the action of D2 ? How would evolve a cell whose D2 expression is blocked ?” Since it is difficult to answer these new questions from an *in vivo* biological model, an *in vitro* cellular model is nowadays developed by the developmental biologists we work with. Hypotheses formulated from the *in silico* model will then be tested on the *in vitro* cellular model before experimentation on the *in vivo* model. The *in vitro* cellular model then corresponds to an intermediate model, indispensable for the *in silico* model exploitation.

From a theoretical point of view, we develop in this paper a model-checking procedure for THPN. The global procedure has been proved correct and complete: we proved that the THPN model satisfies the studied property f if and only if the language of the product automaton is empty. This work opens two main research directions. Firstly, THPN correspond to a restricted class of HFPN in which the functional aspect has been removed. We then plan to re-introduce functions by restricting their expressiveness. Secondly, we would like to model check parametric models in which some dynamical parameters are treated symbolically. It would then be possible to answer this kind of questions: “What is the speed value of a transition allowing the model to satisfy a given property ?”

The presented model-checking procedure seems to be an enough powerful basis to develop these points on the long range.

Acknowledgments We sincerely thank the Laboratory of Evolution and Development of the Paris-South University and more particularly N. Pollet and R. Thuret.

REFERENCES

1. H. Matsuno, Y. Tanaka, H. Aoshima, A. Doi, M. Matsui, and S. Miyano, *In Silico Biology* **3**, 389–404 (2003). pages 1
2. R. David, and H. Alla, *Discrete, Continuous, and Hybrid Petri Nets.*, vol. 3-540-22480-7, Springer, 2005. pages 1, 3, 4
3. C. Rose, *TRENDS in Ecology and Evolution* **20**, 129–135 (2005). pages 2
4. Z. Wang, and D. Brown, *Journal of Biological Chemistry* **268**, 16270–16278 (1993). pages 2
5. L. Cai, and D. Brown, *Developmental Biology* **266**, 87–95 (2003). pages 2, 3, 8
6. H. Huang, N. Marsh-Armstrong, and D. Brown, *Developmental Biology* **96**, 962–967 (1999). pages 2
7. S. Troncale, R. Thuret, B. C., N. Pollet, J.-P. Comet, and G. Bernot, “Modelling of the TH-dependent regulation of tadpole tail resorption.” in *RIAMS*, 2007. pages 2
8. D. Germain, R. Schwartzman, W. Croteau, A. Kanamori, Z. Wang, D. Brown, and V. Galton, *Developmental Biology* **91**, 7767–7771 (1994). pages 2
9. R. Alur, L. Fix, and T. Henzinger, *Theor. Computer Science* **211**, 253–273 (1999). pages 5
10. J.-F. Raskin, and P.-Y. Schobbens, *Journal of Automata, Languages and Combinatorics* **4**, 247–282 (1999). pages 6, 7
11. S. Troncale, J.-P. Comet, and G. Bernot, Verification of Timed Hybrid Petri Nets with temporal logic, Tech. rep., IBISC (2007). pages 7