A description of dynamical graphs associated to elementary regulatory circuits

E. Remy1,∗, B. Mossé1, C. Chaouiya2 and D. Thieffry2

1Institut de Mathématiques de Luminy, Campus de Luminy, Case 907, 13288 Marseille cedex 9, France and 2LGPD, Campus de Luminy, 13288 Marseille cedex 9, France

Received on March 17, 2003; accepted on June 9, 2003

ABSTRACT
The biological and dynamical importance of feedback circuits in regulatory graphs has often been emphasized. The work presented here aims at completely describing the dynamics of isolated elementary regulatory circuits. Our analytical approach is based on a discrete formal framework, built upon the logical approach of R. Thomas.

Given a regulatory circuit, we show that the structure of synchronous and asynchronous dynamical graphs depends only on the length of the circuit (number of genes) and on its sign (which depends on the parity of the number of negative interactions). This work constitutes a first step towards the analytical characterisation of discrete dynamical graphs for more complex regulatory networks in terms of contributions corresponding to their embedded elementary circuits.

Contact: remy@iml.univ-mrs.fr

INTRODUCTION
Biologists are long aware about the biological and dynamical importance of regulatory circuits, i.e. closed chains of regulatory interactions (see e.g. Monod and Jacob (1961)). In particular, positive circuits (i.e. circuits involving an even number of negative interactions) have been associated with the generation of alternative regimes of gene expressions (Lewis et al., 1977; Meinhardt, 1978; Thomas, 1978). Since then, much progress has been made in the molecular analysis of gene regulation and development. In the process, dozens of examples of direct or indirect positive auto-regulation of key regulatory genes involved in differentiation have been found (e.g. genes MyoD and Wg).

In parallel, the requirement of positive regulatory circuits for multistationarity, as well as that of negative circuits for sustained oscillations, have been formally proved by several authors (Plathe et al., 1995; Gouzé, 1998; Snoussi, 1998). Consequently, the biological roles of positive and negative feedback circuits have been further clarified. Whereas negative circuits allow the buffering of gene dosage effects, as well as tight control of the expression of key regulatory genes, positive regulatory circuits may constitute developmental switches generating alternative developmental pathways, and/or encode positional information.

Despite recent progress in genomics and functional genomics, reproducible and quantitative information about gene networks and their temporal behaviour remains scarce. For this reason, we rely on a discrete formalisation of regulatory networks (regulatory graphs), as well as of their dynamics (dynamical graphs). See also Glass and Kauffman (1973); Kauffman (1993); Edwards et al. (2001); and, for recent reviews, De Jong (2002); Smolen et al. (2000).

Described in Chaouiya et al. (2003), our formal framework combines discrete mathematical and graph-theoretic notions. On the first hand, in a regulatory graph, genes (or other biological molecular components) are associated with nodes (vertices), and regulatory interactions correspond to activating or inhibitory arrows (arcs) linking the corresponding pairs of nodes. On the other hand, in a dynamical graph, the current expression status of a gene network is represented by the value of a vector identifying a specific node (expression state), whereas expression changes (state transitions) correspond to arcs linking relevant node pairs.

This formal approach has already served as a basis to develop a computational tool enabling the simulation of genetic regulatory network (called GINsim, see Chaouiya et al. (2003)), as well as a series of biological applications (see e.g. Sánchez and Thieffry, 2001, and references therein), which clearly emphasise the crucial role of specific regulatory circuits in developmental decisions. The study of such model systems largely confirms the theoretical considerations on the dynamical and biological roles of feedback circuits, while extending these considerations to cases where such circuits are embedded in more complex networks.
A description of dynamical graphs associated to elementary regulatory circuits

However, we still lack a proper theory to specify the structure of dynamical graphs on the sole basis of the knowledge of the corresponding (parametrised) regulatory graph. In this respect, the present manuscript encompasses the results of a systematic analytical investigation of the relationships between isolated elementary regulatory circuits and the structural features of the corresponding dynamical graphs. Even in the case of such relatively simple feedback structures, a variety of dynamical graphs can be generated. Nevertheless, we show below how this variety can be ultimately subsumed by combinations of a limited set of fundamental dynamical motifs. As we shall see, the whole dynamical behaviour of arbitrarily simple feedback structures, a variety of dynamical graphs can be generated.

**GRAPH-BASED MODELLING OF REGULATORY NETWORKS AND THEIR DYNAMICS**

In this section, we briefly describe the formalism, focusing on isolated elementary circuits.

We first recall our definition of a *regulatory graph*. It involves the following four constituent parts:

(i) A set of nodes $G = \{g_1, g_2, \ldots, g_n\}$, called genes, which interact. Note that we will mainly refer to interactions between genes, though these interactions may involve various types of other molecular mechanisms.

(ii) For each node $g_i$, a positive integer $\max_i$ represents its *maximum expression level*. Therefore, the possible levels of expression of $g_i$ are 0, 1, ..., $\max_i$. In the simplest case, called the *Boolean case*, the maximum level $\max_i$ is equal to 1 for each $i$.

(iii) A labelled oriented graph $\mathcal{R}$, where the set of vertices is $G$ and where arcs represent interactions between regulatory products. A number $\varepsilon \in \{-1, 0, 1\}$, called the *sign of the interaction*, is associated to each arc, specifying the nature of this interaction: it may be an *activation* ($\varepsilon = +1$), an *inhibition* ($\varepsilon = -1$) or undetermined ($\varepsilon = 0$). An interval is also attached to each interaction, defining the range of discrete levels for which the interaction is operating; out of this interval, the interaction will have no influence on its target (note that for an interaction with source $g_i$, this interval is included in $\{0, 1, \ldots, \max_i\}$). Multi-arcs between pairs of nodes are allowed.

(iv) For each node $g_j$, an application called *logical function* and denoted by $K_j$. This application associates to any subset $X$ of incoming interactions the value $K_j(X)$, called *parameter*, to which the level of $g_j$ tends when $X$ is the actual set of operating interactions exerted upon it. These logical functions allow the qualitative specification of the effects of combinations of interactions controlling a given gene.

We say that a circuit of $\mathcal{R}$ is *positive* when none of its interactions are undetermined and the number of inhibitions in $\mathcal{R}$ is even (the product of the interactions signs is thus equal to +1). Similarly, it is said to be *negative* when none of its interactions are undetermined and the number of inhibitions in $\mathcal{R}$ is odd (the product of the interactions signs is thus equal to −1).

Let us now restrict ourselves to the case where $\mathcal{R}$ is itself an isolated elementary circuit. Then, the general model is simplified (here and in the following, the indices are considered *modulo* $n$, i.e. $i + n = i$):

Each gene $g_i$ is the target of a unique interaction going out $g_{i-1}$, and is the source of a unique interaction towards $g_{i+1}$. We denote by $T'$ the interaction from $g_i$ to $g_{i+1}$.

The Boolean case is then sufficient: the interval attached to any interaction is reduced to the singleton {1}; therefore $T'$ is determined by $g_i$, $g_{i+1}$ and $\epsilon_i$.

Without loss of generality, we define the logical function $K_j$ associated to $g_j$ as follows:

- if $\epsilon_{j-1} = +1$, $K_j(\emptyset) = 0$ and $K_j((T'j-1)) = 1$. This means that when the activation $T'\_j-1$ is operating, the expression level of the gene $g_j$ is set to 1; otherwise it is set to 0.
- if $\epsilon_{j-1} = -1$, $K_j(\emptyset) = 1$ and $K_j((T'j-1)) = 0$. Here the expression level of the gene $g_j$ is set to 0 when the inhibition $T'\_j-1$ is operating, and to 1 otherwise.
- if $\epsilon_{j-1} = 0$, then either $K_j(\emptyset) = K_j((T'j-1)) = 0$ or $K_j(\emptyset) = K_j((T'j-1)) = 1$.

**Remark 1.** This definition of the parameters is the most natural with respect to the signs $\epsilon_i$ and corresponds to the notion of functional circuit according to R. Thomas (Thomas, 1991; Thomas et al., 1995).

A state of the system is a $n$-uple $x = (x_1, \ldots, x_n)$, where the component $x_i$ represents the level of expression of gene $g_i$ (with $x_i = 0$ or 1).

Let us denote by $E$ the set of all possible states. From this regulatory graph we deduce two different dynamics on this set, respectively called synchronous and asynchronous (see the precise definitions below). The central problem is to understand and to describe these dynamics.

Let us immediately point out that when there exists at least one undetermined interaction in $\mathcal{R}$, then both dynamics are trivial: a unique stationary state is reached in at most $n$ time steps from any initial state. Therefore,
we may and do suppose in the following that all the signs $\varepsilon_i$ are equal to $\pm 1$

We associate to each state $x \in \mathcal{E}$ a n-uple $\mathcal{I}(x) = (\mathcal{I}_1(x) \ldots \mathcal{I}_n(x))$, called instruction and defined by:

$$\mathcal{I}_j(x) = \begin{cases} \emptyset & \text{if } x_{j-1} = 0, \\ \{T_j^{-1}\} & \text{if } x_{j-1} = 1. \end{cases}$$

Therefore, $\mathcal{I}(x)$ represents the set of those interactions which are operating at the state $x$. We also associate a set $\mathit{Upd}(x)$ to each state, gathering the indices for which there exists a call for updating (call for change) of expression level:

$$\mathit{Upd}(x) = \{ j \in [1, \ldots, n], K_j(\mathcal{I}_j(x)) \neq x_j \}.$$

Below, depending on the chosen dynamics, we distinguish between two types of dynamical graphs.

The synchronous dynamical graph. Under the synchronous assumption, at each time step, all updating orders are executed simultaneously. As a result, each state has exactly one successor. From a biological point of view, this assumption implies that all macromolecular processes are realised in identical amounts of times (or "delays"), which is clearly unrealistic and often at the origin of simulation artefacts. Therefore, what follows has only a theoretical interest and, as a matter of fact, will be useful in the sequel.

The synchronous dynamics is generated by the iteration of a transformation $S$, that we define on the set of states $\mathcal{E}$. This application maps each state $x$ to a unique state $y = S(x)$ obtained by simultaneously updating all coordinates of $x$, following the instruction $\mathcal{I}(x)$. This means that, writing $y = (y_1, \ldots, y_n)$, one has

$$y_j = K_j(\mathcal{I}_j(x)). \quad (1)$$

We are then able to define $\xi_s = (\mathcal{E}, \mathcal{F}_s)$, the synchronous dynamical graph, where $\mathcal{E}$ is the set of vertices, and $\mathcal{F}_s$ the set of arcs: each state $x$ is the origin of one single arc, whose end is the state $y = S(x)$.

The asynchronous dynamical graph. Under the asynchronous assumption, when multiple updating orders occur at a given logical state, additional information is needed to select a specific transition (i.e. the values of relevant time delays or at least some ordering relationships). Here, time-delays are associated to each reaction (synthesis, degradation, activation, inhibition), and are all a priori considered different from each other. As we have no information about these time delays, all possible transitions are generated. As a consequence, each state $x$ has a number of successors equals to the number of updating calls at this state.

Under this assumption, the dynamics is less simple to describe. In particular it cannot be expressed in terms of iteration of a transformation defined on $\mathcal{E}$. Hence, we explicitly describe the associated graph $\xi_a = (\mathcal{E}, \mathcal{F}_a)$, called asynchronous dynamical graph: the set of vertices is still the set $\mathcal{E}$, and $\mathcal{F}_a$ denotes the set of arcs. Let $x$ be a given state; for each $j$ belonging to $\mathit{Upd}(x)$ i.e. such that $K_j(\mathcal{I}_j(x)) \neq x_j$, there exists an arc which links $x$ to

$$y = (x_1, \ldots, x_{j-1}, x_j, x_{j+1}, \ldots, x_n). \quad (2)$$

Therefore, two linked states $x$ and $y$ differ by exactly one coordinate. Moreover, in the asynchronous graph, one arc represents one updating order.

Finally, the set of stationary states is $\{ x \in \mathcal{E} ; \mathit{Upd}(x) = \emptyset \}$. They are the same in both synchronous and asynchronous cases: these states are those which have no successor distinct from themselves.

Remark 2. Note that we are considering the whole dynamical graph, but a particular pathway (given a set of initial states) can be extracted as a subgraph of $\xi_s$ or $\xi_a$, depending on the updating hypothesis.

ANALYTICAL DYNAMICAL STUDY OF ISOLATED CIRCUITS

We will now describe the structure of the graphs $\xi_s$ and $\xi_a$ in the case of isolated circuits. Our main result can be summarised as follows.

Main Result. The graphs $\xi_s$ and $\xi_a$ can be completely described, for arbitrary large circuits. These graphs are, up to the numbering of the states, entirely determined by the number $n$ of genes and the sign of the regulatory circuit.

In other words, for a given number of genes and up to the numbering of the states, each type of circuit, positive or negative, leads to exactly one synchronous and one asynchronous dynamical graphs. In the sequel we show how these graphs are constructed (proofs not shown for sake of space).

Let us first introduce some notation.

Notation 3. 1. For $a \in \{0, 1\}$, we set $\bar{a} = 1 - a$.

More generally, if $a = (a_1, \ldots, a_n)$ is a n-uple of elements in $\{0, 1\}$, we set $\tilde{a} = (a_1, \ldots, \bar{a}_n)$ and call it the mirror of $a$.

2. For $a \in \{0, 1\}$, we define:

$$a^\varepsilon = \begin{cases} a & \text{if } \varepsilon = 1, \\ \bar{a} & \text{if } \varepsilon = -1. \end{cases} \quad (3)$$

This function has the following property: $(a^{\varepsilon_1})^{\varepsilon_2} = a^{\varepsilon_1 \cdot \varepsilon_2}.$
A useful geometrical representation

It will be convenient to geometrize the representation of the set $\{1, \ldots, n\}$ and its subsets.

The set $\{1, \ldots, n\}$ is simply represented by $n$ points regularly spread on a circle and labelled from 1 to $n$. The application $\sigma$ is then identified to a rotation of angle $\frac{2\pi}{n}$ around the center of the circle. Note that $\sigma^n$ is the identity function.

Any subset $P$ of $\{1, \ldots, n\}$ with $k$ elements is called a $k$-motif; these $k$ elements are marked with a star on the circle (see Fig. 1).

We call configuration of $P$ the disposition of the elements in $P$ around the circle, up to rotations: all $k$-motifs obtained from $P$ by the iterated action of $\sigma$ have same configuration (see Fig. 1). Let consider $P$ a $k$-motif, and $m$ the smallest strictly positive integer such that $\sigma^m(P) = P$. It can be proved that $m|n$ and, with $d = \frac{n}{m}$, that $d|k$; the numbers $m$ and $d$ depend only on the configuration of $P$.

Denoting by $A(k, n)$ the number of configurations of $k$-motifs, one has:

$$A(k, n) = \frac{1}{n} \sum_{d|\text{GCD}(n, k)} \phi(d) C_{\frac{n}{d}}^{\frac{k}{d}},$$

where GCD stands for ‘Greatest Common Divisor’, $C_p^q = \frac{q!}{p!(q-p)!}$ is a binomial coefficient and $\phi$ the Euler indicator (i.e. $\phi(d)$ is the number of natural integers $i$ such that $1 \leq i \leq d$ and $i$ and $d$ are relatively prime). Remark that $A(k, n) = A(n - k, n)$.

The numbers $A(k, n)$ will be used in the following description of the synchronous graph $\xi_\epsilon$.

The synchronous dynamical graph

The synchronous application $S$ is equivalently defined by $(S x)_n = x_1^\epsilon$ (according to (3)); where $x = (x_1, \ldots, x_n)$ is a state and $\epsilon$ the sign of $T^\dagger$. It is a one-to-one transformation (permutation) of $E$. Moreover, $Upd(Sx)$ is obtained from $Upd(x)$ by the action of $\sigma$:

$$Upd(Sx) = \sigma(Upd(x)).$$

The permutation $S$ is composed of disconnected cycles; for any vertex $x$ in any cycle, $S(x)$ defines its successor.

Let $P$ be a $k$-motif. Remind that if $x$ is a state and $Upd(x) = P$, the number $k$ is therefore the number of calls for updating, and the $k$-motif $P$ defines positions of theses calls upon the components of the state $x$.

- In the case of positive circuits, if $k$ is even, there exists exactly two states $x$ and $y$ in $E$ such that $Upd(x) = Upd(y) = P$, which are mirroring each other ($y = x$). In particular, there exists two stationary states (case $k = 0$). If $k$ is odd, there is no state associated to $P$.

- In the case of negative circuits, if $k$ is odd, there exists exactly two states $x$ and $y$ in $E$ such that $Upd(x) = Upd(y) = P$, which are mirroring each other ($y = x$). In this case, there is no stationary state (as $k \geq 1$). If $k$ is even, there is no state associated to $P$.

Let us describe the structure of $\xi_\epsilon$ for a given number of genes $n$. It can be decomposed in different disconnected stages, each of them composed of disconnected cycles. The vertices of the $k$th stage are the states having $k$ updating orders, i.e. $\{x \in E : \#Upd(x) = k\}$. Moreover, given $k$, all states $x$ in the $k$th stage are distributed according to the $A(k, n)$ possible configurations for $Upd(x)$. To each of the $A(k, n)$ configurations ($k$ having the relevant parity) correspond $2m$ states. These $2m$ states are connected into either two cycles of length $m$ when $\frac{k}{2}$ is even, or into a single cycle of length $2m$ when $\frac{k}{2}$ is odd ($m$ and $d$ defined as previously for a given configuration).

These results completely describe the topology of $\xi_\epsilon$.

From now on, we will denote by $C_{ik}^q$ the $i$th cycle in the $k$th stage (in an arbitrary fixed numbering).

The asynchronous dynamical graph

We can now characterise the structure of the asynchronous graph $\xi_\alpha$, with the help of $\xi_\epsilon$ and its ‘staged’ structure.

Contrary to $\xi_\epsilon$, the graph $\xi_\alpha$ constitutes a single connected component (compare second and third column of Fig. 2). For $k = 0$ and $k = n$, the vertices of the $k$th-stage of $\xi_\alpha$ are the strongly connected components of $\xi_\alpha$, reduced to a single vertex.

When $k \neq 0$ and $k \neq n$, the vertices of the $k$th-stage of $\xi_\alpha$ forms a strongly connected component of $\xi_\alpha$ denoted...
by $C_k$. Each state $x$ in $C_k$ has exactly $k$ successors (remind that $x \in C_k$ means that $x$ receives $k$ updating orders). We will see that these successors are either in $C_k$, or in $C_{k-2}$. Thus, there is no way to reach upper stages from the component $C_k$ (i.e. the number of updating calls never increases).

More precisely, let $x$ be in $C_k$ (where $k \geq 1$).

- For any $i$ in $Upd(x)$ such that $i + 1 \not\in Upd(x)$, there is an arc linking $x$ to a state $y$ such that $Upd(y) = (Upd(x) \setminus \{i\}) \cup \{i + 1\}$ (and so $\#Upd(y) = \#Upd(x) + 2 = k + 2$ and $y$ is a node of $C_{k-2}$). Moreover, $S(x)$ is linked to $S(y)$ in $x_a$.

- When both $i$ and $i + 1$ belong to $Upd(x)$, there is an arc linking $x$ to a state $y$ such that $Upd(y) = Upd(x) \setminus \{i, i + 1\}$ (and so $\#Upd(y) = \#Upd(x) - 2 = k - 2$ and $y$ is a node of $C_{k-2}$). Moreover, $S(x)$ is linked to $S(y)$ in $x_a$.

These results completely describe the topology of $x_a$.

Figure 2 illustrates three examples of regulatory graphs (first column), with the corresponding synchronous (second column) and asynchronous (third column) dynamical graphs: (A) logical scheme of the ‘toggle switch’, involving two cross-repressing genes, as described in Gardner et al. (2000); (B) more detailed logical scheme for the same toggle switch; (C) logical scheme for a three elements negative circuit, involving three negative interactions, as in the repressilator described in Elowitz and Leibler (2000); the nodes G1, G2, and G3 each stands for a repressor gene; the nodes P1 and P2 stand for regulatory products (proteins). The regulatory graph of the detailed toggle-switch means that the first gene (G1) is at the origin of the expression of the first protein (P1), which inhibits the expression of the second gene (G2), etc. In (A) and (C), only the global interactions between genes are described.

In second and third columns, the nodes represent states of gene expression, each labelled by a Boolean vector, in which each bit corresponds to a specific gene or gene product (‘0’ stands for the absence of a product or for the lack of expression, ‘1’ for the opposite case); arcs (arrows) stand for allowed state transitions. Note that a state can be the source of at most one single arc in synchronous graphs (second column), loops being omitted on terminal nodes, which is not further true for asynchronous graphs (third column).

Finally, the fourth column of Figure 2 gives the different values of $k$ (number of updating orders) corresponding to the different stages of the dynamical graphs.

A compact notation for the asynchronous dynamical graph

The asynchronous graph is quite complex because it contains many arcs, particularly for large $n$. There is a much more simple graph which is, in some sense, a simplification of the asynchronous graph, and which allows to reveal its main structure. We call it the simplified asynchronous dynamical graph, and denote it by $S_a$.

By definition, the vertices of $S_a$ are the different cycles $C_k$ which compose $x_a$. If there exists in $x_a$ at least one arc from some element in $C_k$ to some element in $C_h$, we define an arc in $S_a$ from $C_k^i$ to $C_h^j$.

In other words, while vertices in graphs $x_a$ and $x_a$ correspond to states, the simplified graph provides a description of the asynchronous dynamics at the level of configurations.

The fact that every path in $S_a$ admits at least one lifting as a path in $x_a$ makes relevant the consideration of $S_a$.

Figure 3 represents the compact graph corresponding to the detailed toggle switch of Figure 2.

CONCLUSIONS AND PROSPECTS

In this manuscript, we are proposing a graph-based representation of the discrete dynamics of genetic regulatory networks. Focusing on the simple case of isolated regulatory circuits, we have shown how the structure of the asynchronous dynamical graphs can be analytically computed in terms of elementary cycles. Building on this analytical formulation, we could then derive the structure of the corresponding asynchronous dynamical graphs. In agreement with previous work on the role of positive versus negative circuits, our analysis of asynchronous graphs points to fundamentally different features for these two classes of circuits, i.e. the occurrence of two stable states in the case of functional positive circuits, versus the occurrence of interconnecting dynamical cycles in the case of functional negative circuits (Thomas et al., 1995). In this respect, we are further proposing a new notation for asyn-

**Fig. 3.** The simplified asynchronous graph for the more detailed toggle switch model (cf. Fig. 2 middle-right).
chronologial dynamical graphs, which ease their visualisation and interpretation.

With these results on the precise structure of the whole asynchronous dynamical graph, we can naturally address now the question of the asymptotic behaviour of the modelled system. For example, is the system susceptible to follow a trajectory in a single stage of the asynchronous graph, or is it necessarily attracted in the lower stage (corresponding to the minimum number of updating calls)?

The results presented constitute the first outcome of a more general, systematic analysis of the relationships between logical regulatory graphs and the corresponding (a)synchronous dynamical graphs. Our approach has now to be progressively generalised to encompass:

- multiple inputs and/or outputs branching on a circuit;
- the consideration of multi-level variables (allowing the representation of several qualitatively different levels of expression for some genes);
- the consideration of intertwined circuits.

A crude approach would consist in enumerating the different cases, combining inputs levels. However, we believe that it should possible to develop a more analytical classification of the different configurations.

In the face of the complexity of real cross-regulatory networks, we are further considering the use of the graph-theoretic notions of circuit and of strongly connected component to decompose complex regulatory graphs into well-defined sets of intertwined circuits or cross-regulatory modules (for a discussion of this notion of cross-regulatory modules and a specific application, see Thieffry and Sánchez (2003)). We plan to study the dynamical contribution of each circuit separately, leading to the generation of the corresponding dynamical graph. The challenge then consists in deriving analytical tools to connect these dynamical graphs in order to provide a consistent and global characterisation of the discrete asynchronous dynamical graph for the complete system.

At present, the combinatorial explosion of logical models impedes exhaustive enumerative exploration of the dynamical properties of large logical models, forcing modeller to focus on specific numerical simulations. In this respect, the prospect of a thorough and general analytical treatment of the relationships between logical regulatory graphs and discrete asynchronous dynamical graphs shed a new light on the way com-

<table>
<thead>
<tr>
<th>Regulatory graph</th>
<th>Dynamical synchronous graph</th>
<th>Dynamical asynchronous graph</th>
<th>Stages</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A) Toggle-switch</td>
<td><img src="A" alt="Diagram" /></td>
<td><img src="B" alt="Diagram" /></td>
<td>k=0</td>
</tr>
<tr>
<td>G1 ←_→ G2</td>
<td><img src="C" alt="Diagram" /></td>
<td><img src="D" alt="Diagram" /></td>
<td>k=2</td>
</tr>
<tr>
<td>(B) More detailed toggle-switch</td>
<td><img src="E" alt="Diagram" /></td>
<td><img src="F" alt="Diagram" /></td>
<td>k=4</td>
</tr>
<tr>
<td>G1 ←_→ P1</td>
<td><img src="G" alt="Diagram" /></td>
<td><img src="H" alt="Diagram" /></td>
<td>k=2</td>
</tr>
<tr>
<td>P2 ←_→ G2</td>
<td><img src="I" alt="Diagram" /></td>
<td><img src="J" alt="Diagram" /></td>
<td>k=4</td>
</tr>
<tr>
<td>(C) Repressilator</td>
<td><img src="K" alt="Diagram" /></td>
<td><img src="L" alt="Diagram" /></td>
<td>k=3</td>
</tr>
</tbody>
</table>

Fig. 2.
plex models of genetic regulatory systems are being modelled.

In parallel with our analytical efforts, we are applying our logical formalism and the corresponding software implementation (GIN-sim Chaouiya et al. 2003) to specific regulatory graphs found to be involved in the control of the cell cycle, cell differentiation, and pattern formation during the development of the fly *D. melanogaster* (see e.g. Sánchez and Thieffry 2001, Accepted; Thieffry and Sánchez 2003). These applications demonstrate the power of the approach to qualitatively reproduce wild-type differentiative pathways as well as to simulate the dynamical behaviour of the corresponding gene networks, in the presence of various types of mutations or perturbations. Beyond such simulations, the analytical approach initiated here provide means to directly derive crucial dynamical properties from the analysis of the structure of regulatory networks.

**ACKNOWLEDGEMENTS**

We thank V. Chepoi, Y. Vaxes (LIF Marseille) and R. Lima (CPT Marseille) for discussions in the context of interdisciplinary seminars, as well as P. Tchamitchian (UIII Marseille) for his critical reading of previous versions of this manuscript. Finally, we acknowledge financial support through the French project *inter-EPST Bioinformatique*.

**REFERENCES**


