A SAT-based approach to decipher Gene Regulatory Networks

Fabien Corblin[1] & Lucas Bordeaux[2] & Youssef Hamadi[2] & Eric Fanchon[1] & Laurent Trilling[1]

[1]Laboratory TIMC-IMAG, UMR UJF/CNRS, Faculté de Médecine, 38706 La Tronche, France
[2]Microsoft Research Cambridge, UK

abstract

Computer tools are needed in systems biology to analyse qualitatively the dynamics of Gene Regulatory Networks (GRNs). Particularly, biologists are interested in infering these networks from observed behaviours. In this paper we present a Boolean satisfiability (SAT) approach applied on a widely used asynchronous logical description of such networks. After a brief presentation of the asynchronous logical formalism, we explain how we express into constraints the evolution rule of GRNs. Then, we show how to translate efficiently these constraints into Boolean formulae. We finally report results about infering parameters of a biological model of the λ-phage immunity control. Our study shows that SAT solving is a powerful tool for analysing GRNs and related transition systems found in biological applications.

1 Introduction

The comprehension of a cellular system can pass by the modeling of a Gene Regulatory Network (GRN). For this purpose, it is necessary to look for the main players (genes) and the existence of interactions between them. Most often the knowledge of these interactions in terms of kinetic data, composition functions of several interactions or activation threshold values of an interaction remains incomplete.

In order to solve this problem, we propose a formal tool to help the reasoning which lean on a constraint technology, the Boolean satisfiability (SAT). This technology permits to describe a problem in terms of mathematical relations (the constraints) on a set of variables. It induces a declarative approach which permits to model a network and the rules which govern its dynamic without fixing the variables of the problem and finally to perform a mix of simulation and inference according to some biological observations or assumptions. It allows to gain knowledge on a GRN by solving the related combinatorial problem.

To be specific, we use a well accepted formalism in the domain of Gene Regulatory Network (GRN) analysis: the asynchronous logical description by R. Thomas [9]. We have described how to formalize and analyse these networks with constraint programming (CP) [5, 3], we propose a SAT representation for this problem to improve significantly the performances by using a SAT solver [4].

This paper is organized as follows. First, the problem is expressed and the asynchronous logical description is presented. Then, the design and the main ideas of the SAT model are presented. Results and some comparative evaluation follows. Finally, conclusions and plans for future work are laid out.

2 The problem

The problem can be expressed as follows: Is there a model composed of identified genes and interactions which is coherent with some observed qualitative behaviours and hypotheses ? If yes, what are the properties (about behaviours, kinetics of reactions) of the models which are compatible with the behaviours and hyptohesese?

To be more precise we define in the following the GRNs and the asynchronous logical description.

2.1 Gene Regulatory Network

To introduce the notion of Gene Regulatory Network, and we use for illustration purposes a particular biological process, the λ-*phage immunity control* [8], whose GRN we now call λGRN for short.

A GRN abstracts the interactions between several *genes* of a cell. An interaction can be an *activation* or an *inhibition*. For instance, in the λGRN, the gene *cI* inhibits gene *cII*: it means that it has been observed at least one case where the quantity of proteins *cI* being low (under a certain threshold), the concentration of proteins *cII* has tended to increase, while it has not been observed if *cI* is high (above this threshold).

In many biological interactions the influence of one gene upon another can be thought of as a sigmoid function: there is a threshold in the concentration of the protein at which the effect on the production of the target protein changes steeply from efficient to inefficient. So, sigmoids are often approximated by step functions.

The interactions between the genes are traditionally represented by an *interaction graph*: see Fig. 1 for the interaction graph of our λGRN example. In this graph the nodes represent the genes and the arrows from a node a to a node b model the fact that the concentration of a affects the concentration of b. Each arrow carries two annotations: a sign denoting the nature of the interaction: $+$ for activation, $-$ for inhibition; an index for the threshold at which the interaction changes its activity status.

The thresholds, noted t_c^j (where j is the index of the threshold and c is the component), at which these changes occur may have no precise numerical value, but we can sometimes determine experimentally the order between these thresholds. For instance, for the species *CI*, t_{cI}^1 is the threshold for the interaction on *N*, t_{cI}^2 is the threshold for the three interactions on *cI* (itself), *cro* and *cII* (cf. Fig. 1) and the order between these two threshold is the following: $t_{cI}^1 < t_{cI}^2$.

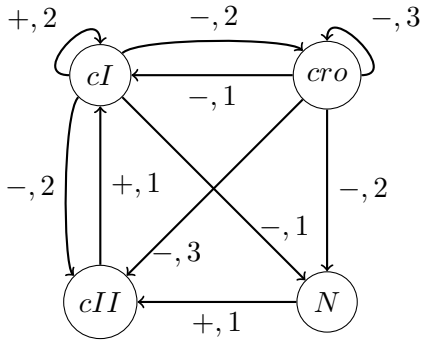

Figure 1: Interaction graph for the λ GRN.

2.2 Asynchronous logical description

We present the asynchronous logical description, created by the biologist R. Thomas [9], which relates the interaction graph of a GRN and its dynamic behaviours. The main goal of this formalism is to obtain a qualitative understanding of the network dynamics by reasoning on discrete entities.

It can be described as follows:

1. The model is purely discrete:

    - The concentration of each protein a is modeled by a discrete variable, noted X_a. If there are i thresholds of interest for the concentration of a, then X_a will range over

$[0, i]$[1]. For the λGRN, we obtain 3, 4, 2 and 2 discrete values for the concentrations of *cI, cro, cII* and *N*, respectively.

- A *discrete concentration state* X, or just *state*, of the system is represented by a vector of discrete values for each of its genes. For instance a possible state X of the λGRN is $X = \langle X_{cI} = 0, X_{cro} = 1, X_{cII} = 0, X_N = 0\rangle$, or $X = \langle 0, 1, 0, 0\rangle$ for short. This state is interpreted as follows: the concentration of genes *cI*, *cII* and *N* are all below their lowest threshold, while the concentration of *cro* is between its first and second thresholds. In the λGRN we have 48 different states.

2. The *transition rule* defines for a particular state which are its successors. The transition rule is based upon the following notions:

   (a) *Tendency*: in a given state, the system can be thought of as tending to evolve towards a new state, called *focal state*. The complex mix of influences between the genes of the system is therefore reduced to abstractions of the form "in state $\langle 0, 1, 0, 0\rangle$, the system tends to evolve towards the focal state $\langle 2, 0, 0, 0\rangle$". The focal state of a state X is noted F_X and the value of each of its components c is denoted $F_{c,X}$.

   (b) *Asynchronicity*: in a given transition, the system is not supposed to cross two or more thresholds simultaneously.

   Then, the transition rule can be expressed as follows:

   - If the system's current state X is different from its focal state F_X, then the concentration of one of its components c will change, by one unit, and in the direction indicated by $F_{c,X}$. This is done independently for each component so that a state X can have several successors We thus obtain a non-deterministic transition system.
   - If the system current state is equal to its focal state, the system is said to be in a *steady state* and no concentration changes.

   For instance if the state $\langle 0, 1, 0, 0\rangle$ has $\langle 2, 0, 0, 0\rangle$ as focal state, then its successors are either $\langle 1, 1, 0, 0\rangle$ (move along the first dimension towards value 2) or $\langle 0, 0, 0, 0\rangle$ (second dimension, towards value 0).

3. This leaves the question of how each component $F_{c,X}$ of the focal state of a state X is defined. This definition is by case: component c is influenced by a number of other components. For instance the gene *cro* is influenced by *cI* and *cro* (itself); depending on the current discrete values of X_{cI} and X_{cro}, the tendency $F_{cro,X}$ can take different values. As in the λGRN $t^1_{cI} < t^2_{cI}$ and $t^1_{cro} < t^2_{cro} < t^3_{cro}$ (that gives the value 2 and 3 for the discrete thresholds t^2_{cI} and t^3_{cro}, respectively). We obtain 4 possible cases for $F_{cro,X}$:

$$F_{cro,X} = \begin{cases} P^1_{cro} & \text{if } X_{cI} \geq 2 \wedge X_{cro} \geq 3 \\ P^2_{cro} & \text{if } X_{cI} \geq 2 \wedge X_{cro} < 3 \\ P^3_{cro} & \text{if } X_{cI} < 2 \wedge X_{cro} \geq 3 \\ P^4_{cro} & \text{if } X_{cI} < 2 \wedge X_{cro} < 3 \end{cases} \quad (1)$$

   The parameters P^j_c (where j is the term number and c is the component) are called *discrete kinetic parameters* (they characterize the kinetics of the system). These parameters are in general not known.There are, however, constraints on these parameters that are imposed by the interaction graph. For instance, because the observation of the negative interaction of *cI* on *cro*, we add the following *observability* constraint: $P^1_{cro} < P^3_{cro} \vee P^2_{cro} < P^4_{cro}$. We add

[1]Note that this model implies that two thresholds with different labels also have different numerical values.

also *additivity* constraints which define the composition function of several interactions on a same gene like a sum. For instance, for the two interactions influencing cro, we add the following constraint: $P^1_{cro} \leq P^2_{cro} \wedge P^1_{cro} \leq P^3_{cro} \wedge P^2_{cro} \leq P^4_{cro} \wedge P^3_{cro} \leq P^4_{cro}$. Note that we could left undefined this composition function or define it in another manner.

Note that this transition rule (tendency and asynchronicity) permits to represent in intension a transition system by the use of the concept of focal state.

3 Encoding the problem of analysing GRN in Conjunctive Normal Form (CNF)

First, we give a general explanation of the SAT problem and the interest of this representation in our context. Then, we present a formal definition of a model in terms of few formulae using multivalued variables and numerical/Boolean constraints. Finally, we describe the translation of this definition into CNF.

3.1 SAT problem and SAT solvers

The SAT problem consists in finding a valuation of a set of Boolean variables (a binding of the Boolean variables to values) which renders true a CNF formula. A CNF formula is a conjunction of *clauses*, where each clause is a disjunction of *literals* which represent a Boolean variable or its negation. A CNF formula is said to be *satisfiable* if it accepts an assignment (valuation of the variables) which renders it true, otherwise it is said to be *unsatisfiable*.

The SAT solvers are optimized to find quickly a solution to a formula in the very specific format of conjunction of clauses. These solvers offer a very efficient manner to solve a lot of combinatorial problems, even with hundred of thousands of variables. The efficiency of SAT solvers is the result of a well adapted representation of CNF formula, a subtle equilibrium between deductions and "enumerations" (attribution of values to well-chosen variables), and the use of a backtrack search with some adaptive heuristics [1].

The key issue to apply SAT is to find an efficient CNF encoding of the problem to solve. In our case, the first difficulty is the need to use only Boolean variables to obtain a CNF. That mean that we will have to translate each multi-valued variable into a collection of Boolean variables. A second difficulty arises because of the number and the size of clauses to be generated (cf. section 3.3).

3.2 Description of a model

The description of the model is lead by the modeling of the transition rule. We must express the formal relation between a state and its successor states, by using intermediate variables relative to the focal state and the *term* conditions appearing in the focal equation (e.g. $X_{cI} \geq 2 \wedge X_{cro} \geq 3$ in equation (1)). For the sake of clarity, we first express this relation using a high-level description in terms of multi-valued variables and arbitrary numerical/Boolean constraints (section 3.3 will explain how this is mapped into a Boolean encoding).

3.2.1 Relation between a state and term conditions in focal equations.

The first step is to create intermediate Booleans variables, noted $B_{i,X}$, defining the truth value of each *elementary* condition in focal equations for a state X and an interaction i. An elementary condition is associated to a single interaction (arrow in the interaction graph, like in figure 1) and reflects the "activity" or the "inactivity" of this interaction. Formally a single interaction i is described by a tuple of four entities: an effector given by *effector(i)*, a target gene given by *target(i)*, a sign given by *sign(i)*, a discrete threshold given by *threshold(i)*. For example, the interaction in

λGRN from *cI* to *cro* (cf. figure 1) is identified by $(cI, cro, -, 2)$. Then, the notion of "activity" $B_{i,X}$ is defined by:

$$B_{i,X} = \begin{cases} (X_{\textit{effector(i)}} \geq \textit{threshold(i)}) \text{ if } sign(i) = + \\ (X_{\textit{effector(i)}} < \textit{threshold(i)}) \text{ if } sign(i) = - \end{cases}$$

Now, it becomes easy to define formally the *term* condition noted $C^j_{c,X}$, for each term j of each component c for the state X. The variables $C^j_{c,X}$ depend only of the variables $B_{i,X}$ such that $target(i) = c$. More precisely, $C^j_{c,X}$ is a conjunction of such $B_{i,X}$ or negation of such $B_{i,X}$. For example, for the equation (1) in 2.2, given a state X, we have $C^3_{cro,X} = B_{i1,X} \wedge \neg B_{i2,X}$ with $i1 = (cI, cro, -, 2)$ and $i2 = (cro, cro, -, 3)$. Note that, by construction of the focal equations, for a given state X and a component c, we have exactly one $C^j_{c,X}$ true.

3.2.2 Relation between term conditions, parameters and focal components.

The second step of the modeling is to define the relation between term condition $C^j_{c,X}$, parameters P^j_c and focal components $F_{c,X}$ for a current state X, as follows:

$$\forall c, \forall j,\ C^j_{c,X} \rightarrow F_{c,X} = P^j_c$$

Example: For a state X with $X_{cI} = 2$ and $X_{cro} = 1$, we obtain $C^2_{cro,X} = true$ and so by implication that $F_{cro,X} = P^2_{cro}$ (cf. 2.2 formula (1) to see the definition of $F_{cro,X}$). For another state X, we can obtain also deduction from the fact that the domains of $F_{cro,X}$ and P^2_{cro} have no shared values, because this property imply that $X_{cI} < 2$ or $X_{cro} = 3$.

3.2.3 Relation between a current state, its focal state and its successor states.

The third step is to define the possible successor states of a current state. We introduce the (multivalued) variable S_c which is the value of the component c of the possible successor state S of a state X. We introduce several Boolean variables: Bst true if X is steady, Bst_c corresponding to the steadiness of X_c, and $Bdown_c$ and Bup_c corresponding to a decrease and an increase of the value of X_c, respectively. With these variables S_c is defined by:

$\forall c, X_c - 1 \leq S_c \leq X_c + 1$	S_c adjacent to X_c
$\forall c, Bdown_c \leftrightarrow S_c = X_c - 1$	defines $Bdown_c$
$\forall c, Bup_c \leftrightarrow S_c = X_c + 1$	defines Bup_c
$\forall c, Bdown_c \rightarrow X_c > F_c$	tendency to go down
$\forall c, Bup_c \rightarrow X_c < F_c$	tendency to go up
$\forall c, Bst_c \leftrightarrow F_c = X_c$	stationary tendency
$Bst \leftrightarrow \bigwedge_c Bst_c$	stationary tendency
$exactly\_one($<list of $Bdown_c, Bup_c$ and $Bst >)$	asynchronicity

The last constraint is the link between all the previous constraints; it enforces that exactly one Boolean variable is true among the $Bdown_c$ variables, the Bup_c variables and Bst.

A rapid examination shows that the number of the Booleans necessary for expressing a transition between two states, namely $B_{i,X}$, $C^j_{c,X}$, $Bdown_c$, Bup_c, Bst_c stays linear according the number of species. In the case of the $C^j_{c,X}$ one can note that their number grows exponentially according to the inside branching factor of the interaction graph. Fortunately, this factor is rarely above 4. It should be remarked also that a path composed of successive states requires for its definition a number of constraints which is proportional to the size of the path.

3.3 CNF encoding

In the following we give the procedure to replace a multivalued variable by a set of Boolean variables and clauses, and the ideas behind the procedures encoding integer relations.

3.3.1 Boolean representation of a multivalued variable.

We choose a very easy manner to encode in Boolean variables the multivalued variables of the model. For every variable X_c, $F_{c,X}$, S_c and P_c^j we introduce as many Boolean variables as the size of their domains, and a set of clauses specifying that exactly one of these new Boolean variables is true.
Example: For the variable X_{cI} having the domain $\{0, 1, 2\}$, we introduce three Boolean variables $x0$, $x1$, $x2$ and the following links to X_{cI} and its values: $x0$ is equivalent to $X_{cI} = 0$, $x1$ is equivalent to $X_{cI} = 1$, etc. Then, we add the clauses: $x0 \vee x1 \vee x2$, $\neg x0 \vee \neg x1$, $\neg x0 \vee \neg x2$ and $\neg x1 \vee \neg x2$.

For a variable with a domain of l values we obtain 1 clause of l literals, and $l(l-1)/2$ clauses of 2 literals (so with immediate propagation on these two literals in current SAT solvers). As in our context the variable domains are small, we obtain finally a quite small set of clauses (of a non penalyzing size) to introduce multivalued variables. For every other variables (which are necessarily also Boolean) we add just one Boolean variable (and not two Boolean variables).

3.3.2 Introduction of relations between (multivalued) variables.

We have to encode only relations with at most two multivalued variables with small domains. In spite of the lack of heavy encoding problems, we cannot use a trivial way to obtain a clausal form by developing the formula deduced from the truth table of the initial relation, because in this case we have still to face an exponential explosion of the number of generated clauses according to the size of the variable domains. With our method we take into account the fact that exactly one of the Boolean variables for each possible value of a multivalued variable is true.
Example: We consider a relation of the type $B \leftrightarrow X = Y$, with the domains $\{0, 1\}$, $\{0, 1, 2\}$ $\{1, 2, 3\}$ for B, X and Y respectively (like $Bup_{cI} \leftrightarrow S_{cI} = X_{cI} + 1$, see section 3.2.3). Let b, $x0$, $x1$, $x2$, $y1$, $y2$ and $y3$ the Boolean variables which are linked to the multivalued variables B, X and Y. To encode this relation, we introduce six clauses relative to the conjunction of the two following formulae:

- $B \rightarrow X = Y$: $(\neg b \vee \neg x1 \vee y1) \wedge (\neg b \vee x1 \vee \neg y1) \wedge (\neg b \vee \neg x2 \vee y2) \wedge (\neg b \vee x2 \vee \neg y2)$
- $X = Y \rightarrow B$: $(b \vee \neg x1 \vee \neg y1) \wedge (b \vee \neg x2 \vee \neg y2)$

From the previous example we have an idea of the encoding algorithm of a relation $B \leftrightarrow X = Y$. The algorithm loops on the values of B, then these of X and finally these of Y and produces the implication (clauses) on the values of these different variables.

To do the encoding of our model we need a procedure for every type of relation: $B1 \vee B2$, $B1 \wedge B2$, $B1 \leftrightarrow B2$, $X = Y$, $X \neq Y$, $X \leq Y$, $X < Y$, $B \rightarrow X = Y$, $X = Y \rightarrow B$, $B \rightarrow X < Y$, $X < Y \rightarrow B$, $B \leftrightarrow \bigwedge_i B_i$ and $exactly\_one(<$ list of $Bi >)$, where the B variables are Booleans and the X and Y variables are potentially multivalued. The size of the CNF representation for all these relations is linear according to the size of the multivalued variable domains or to the number of species of the system, except for $exactly\_one(<$ list of $B_i >)$ which produces a quadratic number of clauses according of the number of species but with two literals only in each clause.

4 Deciphering Gene Regulatory Networks

In this section we give technical informations about the developped tool. Then we present some results obtained by queries on the biological model of λGRN. We give also some comparisons between our tool based on a SAT technology, and our other tool based on a Constraint Programming (CP) technology, implementing exactly the same model with straightforward heuristics.

4.1 The developped tool

The developped tool is constituted of:

- an input interface to express easily the biological model (interaction graph) and the query,
- a module which implements at a multivalued level the asynchronous logical description and which calls the procedures of the CNF encoding module,
- a CNF encoding module,
- an extended version of the SAT solver minisat 2.0 [4] to take into account the queries which need the set of solutions,
- an output interface to show the solution(s) at a multivalued level.

4.2 Case study: the immunity control of λ-phage

The first example of query concerns the identification of all possible steady states of the λGRN for any parameter values. Expressing that state X is a steady state means imposing the existence of a path of two states beginning in the state $\langle X_{cI}, X_{cro}, X_{cII}, X_N\rangle$ and reaching the same state $\langle X_{cI}, X_{cro}, X_{cII}, X_N\rangle$. Note that this enforcement adds some new clauses related to the variables associated to the discrete kinetic parameter to the original ones due to the interaction graph. So all the discrete states of the systems are not possibly or necessarily steady. The query asks for all the possible assignments of the Boolean variables associated to the integer variables X_{cI}, X_{cro}, X_{cII}, and X_N which accept at least one possible assignment of all the discrete kinetic parameter P variables. It is processed by adding a so-called blocking clause to the CNF formula each time a solution is found (to avoid to find again the same solution), and to relaunch the solver with this augmented CNF formula. The translated problem in CNF contains 234 variables and 838 clauses. We obtain the 16 potential steady states in less than 0.01s (on a Pentium M 1.8GHz with 1Go of RAM).

The main query, in the case of λGRN, concerns the imposition of the two possible observed behaviours in response to a infection of a bacterial cell by a λ-phage virus. These behaviours are due to the reachability of the *lytic attractor* corresponding to the cycle between the two states $\langle 0, 2, 0, 0\rangle$ and $\langle 0, 3, 0, 0\rangle$ from the *initial state* $\langle 0, 0, 0, 0\rangle$, and to the reachability of the *lysogenic state* $\langle 2, 0, 0, 0\rangle$ from the initial state.

The query to impose these possible behaviors is more general than the previous one in the sense that it involves paths of states whose length is not known. This length is necessary to generate the system of constraints. An easy way to overcome this problem is to consider paths with a length equal to the number of discrete states in space. In the case of λGRN we have a discrete concentration space of 48 states. But to face this important issue we must find the maximal length of a path without cycles (redundancy) and corresponding to an instanciated λGRN model accepting also the both following paths with a known length (the attractors):

- the path of length 2 beginning in $\langle 0, 0, 0, 0\rangle$ and ending in $\langle 0, 0, 0, 0\rangle$ (steadiness of the lysogenic state),

- the path of three states beginning in $\langle 0, 2, 0, 0\rangle$, passing in $\langle 0, 3, 0, 0\rangle$ and returning in $\langle 0, 2, 0, 0\rangle$ (lytic cycle).

This maximal length is equal to 43 (satisfiability in 1 min. and 13 sec. for a formula with 8304 variables and 52765 clauses). The query to prove the unsatisfiability for a path of 44 states all different takes about 17 min (with a formula with 8577 variables and 54708 clauses).

With this maximal length we can now search for the coherence of the general model and find the possible instanciated models. The query imposes the existence of the two previous paths (attractors) and of the two following paths:

- the path of 43 states beginning in $\langle 0, 0, 0, 0\rangle$ and reaching (in a future step of the path) the state $\langle 2, 0, 0, 0\rangle$ (reachability of the lysogenic state),
- the path of 43 states beginning in $\langle 0, 0, 0, 0\rangle$ and reaching (in a future step of the path) either $\langle 0, 2, 0, 0\rangle$ or $\langle 0, 3, 0, 0\rangle$ (reachability of the lytic cycle),

The query asks for all the assignements of the Boolean variables associated to the discrete kinetic parameter P which accept at least one possible assignment of all the variable associated to the four paths. The translated problem in CNF contains 9809 variables and 46041 clauses. We obtain 175 coherent models in 6s (7s for a path length equal to 48 if we suppose not known the maximal length of a path without cycle). The same query with a length of 15 instead of 43 with our tool based on a CP solver takes about 150s.

5 Conclusions

Analysing the properties of Gene Regulatory Networks is a key problem in systems biology. Our work aimed at proposing an automated tool for this problem.

GRNs are a type of transition systems which, compared with the more classical application domains of SAT [2, 6, 7], exhibit a number of interesting specificities:

- The transition relation of the program is itself not completely known and is defined as a set of constraints that depend upon so-called kinetic parameters. For that reason, the tool needs to be flexible and to allow queries that go beyond classical reachability or temporal logics (LTL, CTL). Typical queries involve the parameters. For instance a biologist may ask "having observed that the system moves from state S1 to state S2, which parameter values are possible?".
- The system is a discrete abstraction allowing to reason on a biological system which, in reality, evolves in continuous time and involves continuous quantities (concentrations of proteins and kinetic parameters). We think that the application is therefore exciting in that we are not modeling a traditional, discrete, digital system, but proteins interacting by activation/inhibition mechanisms.

An important conclusion of our work is that a translation to Boolean constraints is a good approach allowing to solve a large range of queries on this new kind of transition systems. This, we argued, can be explained by the fact that it naturally involves complex Boolean combinations of simple numerical constraints, and that the discrete variables have small domains that are easily converted into Booleans.

On the technical side, a key contribution of the paper is to present the whole abstraction process used for GRNs, from the description of the biological problem to its modeling using discrete constraints. It details, in particular, the encoding into Boolean constraints which is one key to reach good performance.

References

[1] L. Bordeaux, Y. Hamadi, and L. Zhang. Propositional satisfiability and constraint programming: A comparative survey. *ACM Comput. Surv.*, 38(4), 2006.

[2] E. Clarke, A. Biere, R. Raimi, and Y. Zhu. Bounded model checking using satisfiability solving. *Formal Methods in System Design*, 19(1):7–34, 2001.

[3] F. Corblin, E. Fanchon, and L. Trilling. Modélisation de réseaux biologiques discrets en programmation logique par contraintes. *Technique et science informatiques*, 26:73–98, 2007.

[4] N. Eén and N. Sörensson. An extensible SAT-solver. In *SAT*, pages 502–518, 2003.

[5] E. Fanchon, F. Corblin, L. Trilling, B. Hermant, and D. Gulino. Modeling the molecular network controlling adhesion between human endothelial cells: Inference and simulation using constraint logic programming. In V. Danos and V. Schachter, editors, *Computational Methods in Systems Biology*, volume 3082, pages 104–118, 2004.

[6] M. K. Ganai, A. Gupta, and P. Ashar. Efficient SAT-based unbounded symbolic model checking using circuit cofactoring. In *International Conference on Computer-Aided Design (ICCAD)*, 2004.

[7] M. R. Prasad, A. Biere, and A. Gupta. A survey of recent advances in SAT-based formal verification. *STTT*, 7(2):156–173, 2005.

[8] D. Thieffry and R. Thomas. Dynamical behaviour of biological regulatory networks – ii. immunity control in bacteriophage lambda. *Bulletin of Mathematical Biology*, 57:277–297, 1995.

[9] R. Thomas and M. Kaufman. Multistationarity, the basis of cell differentiation and memory. ii. logical analysis of regulatory networks in term of feedback circuits. *Chaos*, 11:180–195, 2001.