Analysing Gene Regulatory Networks by both Constraint Programming and Model-Checking

J. Fromentin[1], J.-P. Comet[2], P. Le Gall[2] and O. Roux[1]

[1] IRCCyN UMR 6597, CNRS & École Centrale de Nantes
[2] IBISC UMR 8042, CNRS & Université d'Évry

Abstract

We propose a formal method to analyse Gene Regulatory Networks (GRN). The dynamics of such a system is often described by an ordinary differential equation system, but has also been abstracted into a discrete modeling due to R. Thomas. This modeling depends on parameters and different values for these parameters are possible. Each instantiation of these parameters defines a possible dynamics and verification tools can be used to select the tuples of values which lead to dynamics coherent with a given temporal property. GRN are so complex that their discrete modeling gives a number of possible dynamics exponential in function of the GRN's size (number of genes and interactions). We propose to use constraint programming and CTL formal language to determine the set of all possible dynamics coherent with the given properties without enumerate all of them.

Gene Regulatory Networks

Definition 1 (GRN) *A GRN is a labelled directed graph* $G = (V_G, E_G)$ *where each vertex* $i \in V_G$ *is provided with a boundary* $\beta_i = \max(1, d_i)$ *where* d_i *is the number of out-going edges of* i*. Each edge* $(i \rightarrow_G j) \in E_G$ *is labelled with a pair* $(\tau_{ij}, \varepsilon_{ij})$ *where* $\varepsilon_{ij} \in \{+, -\}$ *is the sign of the interaction and* τ_{ij}*, called abstract threshold, is an integer value between* 1 *and* β_i*.*

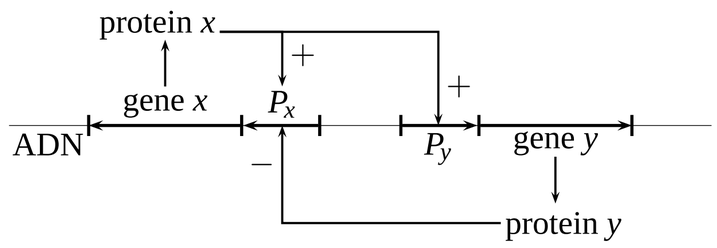
Definition 2 (state) *A state of a GRN is a tuple* $n = (n_1, n_2, \ldots, n_p)$*, where* p *is the number of vertices and* n_i*, the concentration level of* i*, is an integer value between* 0 *and* β_i*.*

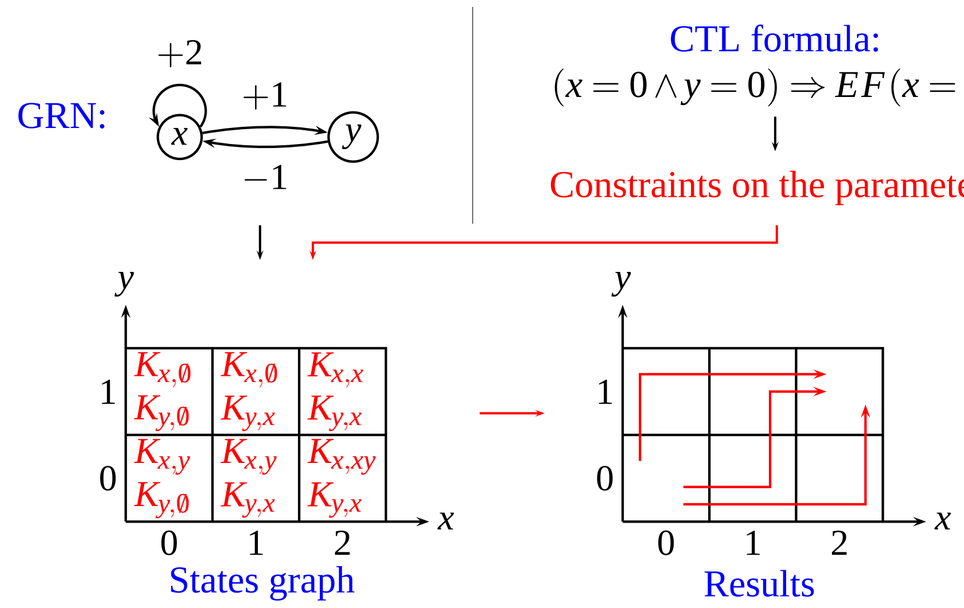
Definition 3 (Resources) *Given a GRN* $G = (V_G, E_G)$ *and a state* $n = (n_1, n_2, \ldots, n_p)$*, the set of resources of* i *is* $\omega_i(n) = \left\{ j \in V_G \;\middle|\; \begin{array}{l} \text{either } (j \xrightarrow{(\tau_{ji},+)}_G i) \in E_G \text{ and } (n_j \geq \tau_{ji}) \\ \text{or } \quad (j \xrightarrow{(\tau_{ji},-)}_G i) \in E_G \text{ and } (n_j \leq \tau_{ji}) \end{array} \right\}$

Definition 4 (Model of a GRN) *Let* $G = (V_G, E_G)$ *be a GRN. The model of* G *is* $M(G) = (Var, Const)$ *with:*

- $K \subseteq Var$ *is a set of parameters such that* $K_{i,\omega_i(n)} \in K$ *denotes the level toward which the concentration of* $i \in V_G$ *evolves when the resources are* $\omega_i(n)$*.*
- *Const is a set of constraints on the variables of the set Var.*

Example: Mucus production in *Pseudomonas aeruginosa*





complexity

The number (#) of constraints and the number of additional variables for the resolution of a problem have the same complexity of $O(\#\text{ states} \times \#\text{ CTL operator})$. The additional variables are strongly constrained and their domain is either $[0, 1]$ or $[1, \#\text{ states}]$.

Implementation

Our software "SeMoCo-GRN" (Selecting of Model by Constraint for GRN) is developed in *Java* using the library *JaCoP* for the constraint programming.

efficiency

The efficiency of the approach is related to the number of solutions. Thus, if the specifications of a model allow a limited number of solutions then, the GRN's size has a limited impact on the efficiency. Therefore, it is possible with this approach to analyse large GRN.