Temporal Logic Constraints in the Biochemical Abstract Machine BIOCHAM

François Fages

INRIA Rocquencourt, France
Francois.Fages@inria.fr

Abstract. Recent progress in Biology and data-production technologies push research toward a new interdisciplinary field, named Systems Biology, where the challenge is to break the complexity walls for reasoning about large biomolecular interaction systems. Pioneered by Regev, Silverman and Shapiro, the application of process calculi to the description of biological processes has been a source of inspiration for many researchers coming from the programming language community.
In this presentation, we give an overview of the Biochemical Abstract Machine (BIOCHAM), in which biochemical systems are modeled using a simple language of reaction rules, and the biological properties of the system, known from experiments, are formalized in temporal logic. In this setting, the biological validation of a model can be done by model-checking, both qualitatively and quantitatively. Moreover, the temporal properties can be turned into specifications for learning modifications or refinements of the model, when incorporating new biological knowledge.

1 Introduction

Systems biology is a cross-disciplinary domain involving biology, computer science, mathematics, and physics, aiming at elucidating the high-level functions of the cell from their biochemical bases at the molecular level. At the end of the Nineties, research in Bioinformatics evolved, passing from the analysis of the genomic sequence to the analysis of post-genomic data and interaction networks (expression of RNA and proteins, protein-protein interactions, etc). The complexity of these networks requires a large research effort to develop symbolic notations and analysis tools applicable to biological processes and data.

Our objective with the design of the Biochemical Abstract Machine BIOCHAM [1, 2] is to offer a software environment for modeling complex cell processes, making simulations (i.e. "*In silico* experiments"), formalizing the biological properties of the system known from real experiments, checking them and using them as specification when refining a model. The most original aspect of our approach can be summarized by the following identifications:

biological model = transition system,
biological property = temporal logic formula,
biological validation = model-checking.

2 Syntax of Biomolecular Interaction Rules

The objects manipulated in BIOCHAM represent molecular compounds, ranging from small molecules to proteins and genes. The syntax of objects and reaction rules is given by the following grammar:

object = molecule | molecule `::` location
molecule = name | molecule`-`molecule |molecule`~`{name,...,name}
reaction = solution `=>` solution | kinetics `for` solution `=>` solution
solution = `_` | object | number`*`object | solution`+`solution

The objects can be localized in space with the operator "::" followed by a location name, such as the membrane, the cytoplasm, the nucleus, etc. The binding operator `-` is used to represent the binding of a molecule on a gene, the complexation of two proteins, and any form of intermolecular bindings. The alteration operator "$\sim$" is used to attach a set of modifications to a protein, like for instance the set of its phosphorylated sites (as long as they impact its activity).

Reaction rules express elementary biochemical interactions. There are essentially seven main rule schemas :

- `G => G + A` for the synthesis of `A` by gene `G`,
- `A => _` for the degradation of `A`,
- `A + B => A-B` for the complexation of two proteins `A` and `B`,
- `A-B => A + B` for the reversed decomplexation,
- `A + B => A~{p} + B` for the phosphorylation of protein `A` at site p catalyzed by `B`,
- `A~{p} + B => A + B` for the reversed dephosphorylation,
- `A::L => A::L'` for the transport of `A` from location L to L'.

The reaction rules can also be given with a kinetic expression, like for instance `0.1*[A][B] for A + B => A-B` where a mass action law kinetics with constant rate `0.1` is specified for the formation of the complex.

This rule-based language is used to model biochemical systems at three abstraction levels which correspond to three formal semantics: boolean, concentration (continuous dynamics) and population (stochastic dynamics).

A second language based on Temporal Logic [3] is used in BIOCHAM to formalize the biological properties of the system, and validate a model by model-checking [4, 5]. More precisely, symbolic and numerical model-checking tools are used respectively for CTL in the boolean semantics, for LTL with constraints over real numbers in the concentration semantics, and for PCTL with constraints over integers in the stochastic semantics.

3 Boolean Semantics

The most abstract semantics is the boolean semantics which ignores kinetic expressions. In that semantics, a boolean variable is associated to each BIOCHAM object, representing simply its presence or absence in the system. Reaction rules are then interpreted as an *asynchronous transition system* over states defined by

the vector of boolean variables (similarly to the term rewriting formalism used in [6]). A rule such as `A + B => C + D` defines four possible state transitions corresponding to the possible consumption of the reactants: $A \wedge B \rightarrow A \wedge B \wedge C \wedge D$, $A \wedge B \rightarrow \neg A \wedge B \wedge C \wedge D$, $A \wedge B \rightarrow A \wedge \neg B \wedge C \wedge D$, $A \wedge B \rightarrow \neg A \wedge \neg B \wedge C \wedge D$. In that semantics, the choice of asynchrony and non-determinism is important to represent basic biological phenomena such as competitive inhibition, where a reaction "hides" another one because it consumes the reactants before the other reaction can occur. Formally, the boolean semantics of a set of BIOCHAM rules is defined by a *Kripke structure* $K = (S, R)$ where S is the set of states defined by the vector of boolean variables, and $R \subseteq S \times S$ is the transition relation between states.

In that boolean semantics, Computation Tree Logic (CTL) formulae are used to formalize the known biological properties of the system, and to query such properties in a model. Given an initial state specifying the biological conditions of the property, typical CTL formulae used in this context are :

- $EF(P)$, abbreviated as `reachable(P)`, stating that the organism is able to produce molecule P;
- $\neg E(\neg Q\ U\ P)$, abbreviated as `checkpoint(Q,P)`, stating that Q is a checkpoint for producing P;
- $EG(P)$, abbreviated as `steady(P)`, stating that the system can remain infinitely in a set of states described by formula P;
- $AG(P)$, abbreviated as `stable(P)`, stating that the system remains infinitely in P and cannot escape;
- $AG((P \Rightarrow EF\ \neg P) \wedge (\neg P \Rightarrow EF\ P))$, abbreviated as `oscil(P)`, a necessary (yet not sufficient without strong fairness assumption) consition for oscillations w.r.t. the presence of molecule P;
- $AG((P \Rightarrow EF\ Q) \wedge (Q \Rightarrow EF\ P))$, abbreviated as `loop(P,Q)`, a necessary condition for the alternance between states P and Q.

BIOCHAM evaluates CTL properties through an interface to the OBDD-based symbolic model checker NuSMV [7]. This technology makes it possible to check or query large models, like the model of the cell cycle control involving 165 proteins and genes, 500 variables and 800 reaction rules reported in [5].

4 Concentration Semantics

Basically the same scheme is applied to quantitative models, where each rule is given with a kinetic expression. The concentration semantics associates to each BIOCHAM object a real number representing its concentration. A set of BIOCHAM reaction rules $E = \{e_i \text{ for } S_i \Rightarrow S'_i\}_{i=1,...,n}$ with variables $\{x_1, ..., x_m\}$, is then interpreted by the following set of (non-linear) ordinary differential equations (ODE) :

$$dx_k/dt = \sum_{i=1}^{n} r_i(x_k) * e_i - \sum_{j=1}^{n} l_j(x_k) * e_j$$

where $r_i(x_k)$ (resp. l_i) is the stoichiometric coefficient of x_k in the right (resp. left) member of rule i. Given an initial state, i.e. initial concentrations for each of the objects, the evolution of the system is deterministic and numerical integration methods compute discrete time series (i.e. linear Kripke structures) describing the evolution of the concentrations over time.

The concentration semantics being deterministic, Linear Time Logic (LTL) is used here to formalize the temporal properties. A first-order fragment of LTL is used to express numerical constraints on the concentrations of the molecules, or on their derivatives. For instance, `F([A]>10)` expresses that the concentration of `A` eventually gets above the threshold value 10. Oscillation properties, abbreviated as `oscil(M,K)`, are defined here as a change of sign of the derivative of M at least K times. These LTL formulae with constraints are checked with an ad-hoc model-checker implemented in Prolog, using the trace of the numerical integration of the ODEs associated to the rules.

5 Population Semantics

The population semantics is the most realistic semantics. It associates to each BIOCHAM object an integer representing the number of molecules in the system, and interprets reaction rules as a continuous time Markov chain. The kinetic expression e_i for the reaction i is converted into a transition rate τ_i (giving a transition probability after normalization) as follows [8]:

$$\tau_i = e_i \times (V_i \times K)^{(1-\sum_{k=1}^{m} l_i(x_k))} \times \prod_{k=1}^{m} (!l_i(x_k))$$

where l_i is the stoichiometric coefficient of the reactant x_k in the reaction rule i. Stochastic simulation techniques [9] compute realizations of the process. They are generally noisy versions of those obtained with the concentration semantics, however qualitatively different behaviors may also appear when small number of molecules are considered, which justifies the use of a stochastic dynamics.

In this setting, LTL formulae can be evaluated with their probability using a Monte Carlo method, which has proved to be more efficient than existing model-checkers for the probabilistic temporal logic PCTL. However, both the stochastic simulation and the model-checking are computationally more expensive than in the concentration semantics.

6 Learning Reaction Rules from Temporal Properties

Beyond making simulations, and checking properties of the models, the temporal properties can also be turned into specifications and temporal logic constraints for automatically searching and learning modifications or refinements of the model, when incorporating new biological knowledge. This is implemented in BIOCHAM by a combination of model-checking and search in the three abstraction levels.

This methodology is currently investigated with models of the cell cycle control (which regulates cell division) for the learning of kinetic parameter values from LTL properties in the concentration semantics [10], and for the learning of reaction rules from CTL properties in the boolean semantics [11]. A coupled model of the cell cycle and the circadian cycle is under development along these lines in BIOCHAM with applications to cancer chronotherapies.

Acknowledgements. This is a joint work with Nathalie Chabrier-Rivier, Sylvain Soliman and Laurence Calzone, with contributions from Sakina Ayata, Loïc Fosse, Lucie Gentils, Shrivaths Rajagopalan and Nathalie Sznajder. Support and fruitful discussions with our partners of the EU STREP project April-II are warmly acknowledged.

References

1. Fages, F., Soliman, S., Chabrier-Rivier, N.: Modelling and querying interaction networks in the biochemical abstract machine BIOCHAM. Journal of Biological Physics and Chemistry **4** (2004) 64–73
2. Chabrier, N., Fages, F., Soliman, S.: BIOCHAM's user manual. INRIA. (2003–2005)
3. Clarke, E.M., Grumberg, O., Peled, D.A.: Model Checking. MIT Press (1999)
4. Chabrier, N., Fages, F.: Symbolic model cheking of biochemical networks. In Priami, C., ed.: CMSB'03: Proceedings of the first Workshop on Computational Methods in Systems Biology. Volume 2602 of Lecture Notes in Computer Science., Rovereto, Italy, Springer-Verlag (2003) 149–162
5. Chabrier-Rivier, N., Chiaverini, M., Danos, V., Fages, F., Schächter, V.: Modeling and querying biochemical interaction networks. Theoretical Computer Science **325** (2004) 25–44
6. Eker, S., Knapp, M., Laderoute, K., Lincoln, P., Meseguer, J., Sönmez, M.K.: Pathway logic: Symbolic analysis of biological signaling. In: Proceedings of the seventh Pacific Symposium on Biocomputing. (2002) 400–412
7. Cimatti, A., Clarke, E., Enrico Giunchiglia, F.G., Pistore, M., Roveri, M., Sebastiani, R., Tacchella, A.: Nusmv 2: An opensource tool for symbolic model checking. In: Proceedings of the International Conference on Computer-Aided Verification, CAV'02, Copenhagen, Danmark (2002)
8. Gibson, M.A., Bruck, J.: A probabilistic model of a prokaryotic gene and its regulation. In Bolouri, H., Bower, J., eds.: Computational Methods in Molecular Biology: From Genotype to Phenotype. MIT press (2000)
9. Gillespie, D.T.: General method for numerically simulating stochastic time evolution of coupled chemical-reactions. Journal of Computational Physics **22** (1976) 403–434
10. Calzone, L., Chabrier-Rivier, N., Fages, F., Soliman, S.: A machine learning approach to biochemical reaction rules discovery. In III, F.J.D., ed.: Proceedings of Foundations of Systems Biology and Engineering FOSBE'05, Santa Barbara (2005) 375–379
11. Calzone, L., Chabrier-Rivier, N., Fages, F., Gentils, L., Soliman, S.: Machine learning bio-molecular interactions from temporal logic properties. In Plotkin, G., ed.: CMSB'05: Proceedings of the third Workshop on Computational Methods in Systems Biology. (2005)