Systems Biology, Automata, and Languages

Bhubaneswar Mishra[(1)] and Alberto Policriti [(2)]

(1) Courant Institute of Mathematical Sciences, New York University. 251 Mercer Street, New York, NY 10012.
and
Watson School of Biological Sciences. Cold Spring Harbor Laboratory.
mishra@cs.nyu.edu
(2) Dipartimento di Matematica e Informatica, Università di Udine. Via delle Scienze 206, 33100 Udine. Italy
policriti@dimi.uniud.it

Keywords. System Biology, Modeling, Temporal Query Languages.

The central theme of our work (see [2,3,4]) is related to problem of formulating a "unitary step" that defines how a complex biological system makes a transition from one "state" or one "control mode" to another, as well as the conditions under which such transitions are enabled. This is because we recognize that automata (either discrete or *hybrid*, that is capable of modeling a mixed discrete/continuous behaviour), based on the formulation of these unitary steps, can elegantly model biological control mechanisms, allow us to reason about such mechanisms in a modal logic systems with modes constructed over a next-time operator, and can become the foundational framework for the emerging field of systems biology. These models can lead to more rigorous algorithmic analysis of large amounts of biological data, produced as (numerical) *traces* of *in vivo*, *in vitro* and *in silico* experiments—currently a central activity for many biologists and biochemists. Since modeling biological systems requires a careful consideration of both qualitative and quantitative aspects, our automata-based tools can effectively assist the working biologists to make predictions, generate falsifiable hypotheses and design well-focused experiments—activities in which the *time* dimension and a properly designed *query language* cannot be left out of consideration.
Thus, ultimately, the aim of our work is to elucidate the role played by *automata* in modeling biological systems and to investigate the potential of such tools when combined with more "classical" approaches used in the past to devise models and experiments in biology. Our discussion here is based primarily on our experience with a novel system that we introduced recently (called, XS-systems) and used it to implement algorithms and software tools (Simpathica). These conceptual tools have been integrated with prototype implementations, and are currently undergoing many interesting and growing sets of enhancements and optimizations

1. Modeling and automata construction

Our approach, exemplified in the study of metabolic pathways, was motivated by the desire to design tools capable of using both, data suitable for "classical" mathematical modeling activities as well as, whenever available, information on the control mechanisms underlying the system. In other words, our tools were designed to combine the two somewhat disparate traditions in a single system, *deducing* an automata structure from experimental data as well as from *trajectories* derived by mathematical models.
The basic modeling tools we used were the so-called S-systems: the dynamical systems proposed in [5] and characterized by differential equations of the form:

$$\dot{X}_i = \alpha_i \prod_{j=1}^{n+m} X_j^{g_{ij}} - \beta_i \prod_{j=1}^{n+m} X_j^{h_{ij}}$$

The basic idea of XS-systems (introduced in [4]) is to associate an S-system *S* with a finite automaton, obtained by suitably encoding a set of *traces* on *S*. Essentially, each trace on *S* can be encoded into a simple automaton, where states correspond to the trace elements (i.e., the values of the system variables observed at each step), and transitions reflect the sequence structure of the trace itself.
The notion of *hybrid automata* [1] has also been used in order to refine our modeling procedure and capture a larger and more expressive family of systems. S-system automata retain only quantitative information

maintained as the values of the variables in instants corresponding to steps. The values at instants between two steps are lost. This situation becomes particularly dangerous when we apply a reduction operation such as *collapsing*. We circumvented this problem by using the continuous component of hybrid automata to maintain also some approximate information about the values of the variables between two steps.

2. The query language

In [2], a language called ASySA (*Automata S-systems Simulation Analysis* language) has been presented to inspect and formulate queries on the simulation results of XS-systems. The aim of this language is to provide the biologists with a tool to formulate various queries against a repository of simulation traces. AsySA is essentially a *Temporal Logic* language (an English version of *Computation Tree Logic*) with a specialized set of predicate variables, introduced with the goal to ease the formulation of queries on numerical quantities.

Example. A given automaton can satisfy the formula $\mathbf{Eventually(Always(X_2 > 1))}$ *which means that the system admits a trace such that, from a certain point on,* $\mathbf{X_2}$ *is always greater than* $\mathbf{1}$*. Similarly, it might not satisfy the formula* $\mathbf{Always(Eventually(X_1 > X_2))}$ *since it reaches a steady state in which* $\mathbf{X_1}$ *is less than* $\mathbf{X_2}$*.*

Because the notion of steady state plays a fundamental role in biological systems, a predicate "steady state" has been introduced in the ASySA language. This predicate is satisfied by a system (S-system automaton) if there exists an instant (a state) after which all the derivatives will always be equal to zero, i.e. the system ends in a loop involving only one state.

3. Conclusion

We are left with the following questions: Have we arrived at the "right" automata definitions that naturally capture the biologist's intuitions? Are the "unitary steps" that emerge from these definitions at the right level from a biologist's viewpoint? Is it too detailed, obscuring the central principles of biological control mechanisms? Is it too coarse, missing the effects of principles, central to biology?
As time enters the picture, further ingredients should probably be added: different scales and levels of *granularity*, a certain modularity and freedom in the choice of the tools to be used for the quantitative analysis, the availability of a powerful environment to facilitate the interface with database data or to have automatic check on the formalisms employed.

References

[1] R. Alur, C. Belta, F. Ivancic, V. Kumar, M. Mintz, G. J. Pappas, H. Rubin, and J. Schug. Hybrid Modeling and Simulation of Biomolecular Networks. In *Hybrid Systems: Computation and Control*, volume 2034 of *Lecture Notes in Computer Science*, pages 19–32, 2001.

[2] M. Antoniotti, B. Mishra, A. Policriti, and N. Ugel. Xs-systems: extended ssystems and algebraic differential automata for modeling cellular behavior. In V.K. Prasanna S. Sahni and U. Shukla, editors, *Proceedings of the International Conference on High Performance Computing, HiPC 2002*, volume 2552 of *LNCS*, pages 431–442, Bangalore (INDIA), December 2002. Springer Verlag.

[3] M. Antoniotti, A. Policriti, N. Ugel, and B. Mishra. Model building and model checking for biochemical processes. Cell Biochemistry and Biophysic, 38(3):271–286, 2003.

[4] M. Antoniotti, B. Mishra, C. Piazza, A. Policriti, and M. Simeoni. Modelling cellular behavior with hybrid automata: Bisimulation and collapsing. In C. Priami, editor, *International workshop on Computational Methods in Systems Biology, (CMSB'03)*, volume 2602 of *LNCS*, pages 57–74, Rovereto (ITALY), February 2003. Springer Verlag. Extended version to appear on *Theoretical Computer Science*.

[5] E. O. Voit. Computational Analysis of Biochemical Systems. A Pratical Guide for Biochemists and Molecular Biologists. Cambridge University Press, 2000.