Input/Output Automata: Basic, Timed, Hybrid, Probabilistic, Dynamic,...

Nancy Lynch

MIT, Cambridge, MA 02139, USA
lynch@theory.lcs.mit.edu
http://theory.lcs.mit.edu/~lynch

Abstract. The term *Input/Output Automata* refers to a family of system modeling frameworks based on interacting infinite-state machines. The models come in several flavors, based on which features (fairness, time, continuous behavior, probability, etc.) they can express. In each of these frameworks, automata can be composed in parallel to form more complex automata, and automata can be related using levels of abstraction. Properties of automata can be proved by hand or with the assistance of theorem-proving programs.

The first framework of this kind, which appeared in 1987, was the basic fair asynchronous I/O automata modeling framework of Lynch and Tuttle. It was used originally to describe and analyze a simple network resource allocation algorithm at multiple levels of abstraction. Since then, I/O automata have been used extensively to model distributed algorithms and distributed systems, and even to prove impossibility results. For example, they have been used for algorithms that implement atomic shared memory and for systems that provide group communication services.

Next came the "timed I/O automata" framework of Lynch and Vaandrager, which augmented the "unfair" portion of the basic model with *time-passage steps*. Timed I/O automata have been used to describe a variety of timing-based algorithms, including timeout-based failure detectors and consensus algorithms, communication protocols, and clock synchronization algorithms. They have also been used to analyze performance of many algorithms and systems.

A more recent development was the "hybrid I/O automata" framework, which supports modeling and analysis of hybrid discrete/continuous systems. The main addition here is a set of *trajectories*, which can be used to describe the evolution of system state over intervals of time. Hybrid I/O automata have been used for many case studies, ranging from simple toy examples of vehicles on tracks to complex helicopter control applications.

"Probabilistic I/O automata", defined by Segala in 1995, allow probabilistic choice of the next state, in addition to nondeterministic choice. They have been used for describing and analyzing randomized distributed algorithms and security protocols. Finally, "dynamic I/O automata" were introduced recently by Attie; they add, to the basic "unfair" model, the capability for processes to create other processes and to destroy themselves.

R. Amadio, D. Lugiez (Eds.): CONCUR 2003, LNCS 2761, pp. 191–192, 2003.

In this CONCUR talk, I will define the various I/O automata modeling frameworks in some detail, will survey some of the ways in which they have been used, and will describe current research and open problems. Our current research includes re-formulating timed I/O automata as a restricted version of hybrid I/O automata, and expressing a large set of results about timed systems as theorems about the resulting framework. We are also working on developing the probabilistic I/O automata model further, emphasizing compositionality results. In the longer run, we would like to have a combined model that includes both probabilistic and hybrid continuous/discrete behavior. In fact, eventually, we would like to have a comprehensive I/O-automata-style modeling framework that can express all of the features described above – fairness, time, continuous behavior, probabilistic behavior, and dynamic behavior – but that can be specialized appropriately when only some of these features are needed.

This talk is based on work by many people, notably Mark Tuttle, Frits Vaandrager, Roberto Segala, Paul Attie, and Dilsun Kirli Kaynar.