On the Roles of Informatics in Biology

Oded Maler

CNRS - VERIMAG
Grenoble, France

2007

Systems Biology

- Systems Biology: the new gold rush for many mathematical and technical disciplines
- Biophysics, Biomatics, Bioinformatics, Biostatistics...

Systems Biology

- Systems Biology: the new gold rush for many mathematical and technical disciplines
- Biophysics, Biomatics, Bioinformatics, Biostatistics...
- The generic template:
  - I do X (for my pleasure, because I studied it, that's what I know) but... X can also be useful for Biology

Systems Biology

- Systems Biology: the new gold rush for many mathematical and technical disciplines
- Biophysics, Biomatics, Bioinformatics, Biostatistics...
- The generic template:
  - I do X (for my pleasure, because I studied it, that's what I know) but... X can also be useful for Biology
- So Here I am, trying to sell my own X, a certain species of Informatics / Computer Science
- When you have a hammer, everything looks like a nail

Summary

- Computers are more than technology, it is also mathematics/physics
- Automata as dynamical systems
- Verification illustrated
- On hybrid discrete-continuous models and their verification
- Timed models and their applications
- Back to the big picture

Computer Technology

- Computers, Networks, Operating Systems
- Data-Bases, Web, Search Engines, Graphics,
- Embedded Systems, Sensors, Programming Languages
- Word Processing, Computer Control, Robotics, Security
- ..
- Influence on all domains of human activity

Including Biology

- String Processing for DNA
- Statistical Computations
- Simulation, Animation
- Date-Bases
- Micro-Arrays
- Ontologies and Description Languages
- Communication and Data Sharing
- Lab Management

In all those activities the computer is a useful *material tool* in the *service* of others

A More Noble Role, Perhaps

- Biology seems to be trying to go through a kind of Newtonian revolution
- The essence of such revolution is to upgrade (as much as possible) descriptive "models" by *dynamic* models with stronger predictive power and refutability

A More Noble Role, Perhaps

- Biology seems to be trying to go through a kind of Newtonian revolution
- The essence of such revolution is to upgrade (as much as possible) descriptive "models" by *dynamic* models with stronger predictive power and refutability
- Classical models of dynamical systems are clearly not sufficient for efficient modeling of biological phenomena
- Models, insights and computer-based tools developed within Informatics can help

What Is Informatics ?

- Among other things informatics is:
- The (pure and applied) study of *discrete-event dynamical systems* (automata, transition systems)

What Is Informatics ?

- Among other things informatics is:
- The (pure and applied) study of *discrete-event dynamical systems* (automata, transition systems)
- This point view is more natural for the "reactive systems" parts of informatics (hardware, protocols, real-time, stream processing)
- Especially for people working on modeling and verification of such systems

Dynamical System Models

- State variables whose set of valuations determine the state space

Dynamical System Models

- State variables whose set of valuations determine the state space
- Time domain along which these values evolve

Dynamical System Models

- State variables whose set of valuations determine the state space
- Time domain along which these values evolve
- Dynamic law which says how state variables evolve over time, possibly under the influence of external disturbances

Dynamical System Models

- State variables whose set of valuations determine the state space
- Time domain along which these values evolve
- Dynamic law which says how state variables evolve over time, possibly under the influence of external disturbances
- System behaviors are progressions of states in time
- Having such a model, knowing an initial state $x(0)$ one can predict, to some extent, the value of $x(t)$

Dynamical System Models

- State variables whose set of valuations determine the state space
- Time domain along which these values evolve
- Dynamic law which says how state variables evolve over time, possibly under the influence of external disturbances
- System behaviors are progressions of states in time
- Having such a model, knowing an initial state $x(0)$ one can predict, to some extent, the value of $x(t)$
-
- Remark: Variables in Biology can be of various natures and granularities (concentrations, states of individual molecules, stages in processes, etc.)

Classical Dynamical Systems

- State variables: real numbers (location, velocity, energy, voltage, concentration)

Classical Dynamical Systems

- State variables: real numbers (location, velocity, energy, voltage, concentration)
- Time domain: the real time axis $\mathbb{R}$ or a discretization of it

Classical Dynamical Systems

- State variables: real numbers (location, velocity, energy, voltage, concentration)
- Time domain: the real time axis $\mathbb{R}$ or a discretization of it
- Dynamic law: differential equations

$$\dot{x} = f(x, u)$$

or their discrete-time approximations

$$x(t+1) = f(x(t), u(t))$$

Classical Dynamical Systems

- State variables: real numbers (location, velocity, energy, voltage, concentration)
- Time domain: the real time axis $\mathbb{R}$ or a discretization of it
- Dynamic law: differential equations

$$\dot{x} = f(x, u)$$

or their discrete-time approximations

$$x(t+1) = f(x(t), u(t))$$

- Achievements: Apples, Stars, Missiles, Electricity, Heat, Chemical processes
- Theorems, Papers, Simulation tools

Automata as Dynamical Systems

- Abstract discrete state space, state variables need not have a numerical meaning

Automata as Dynamical Systems

- Abstract discrete state space, state variables need not have a numerical meaning
- Logical time domain defined by the events (order but not metric)

Automata as Dynamical Systems

- Abstract discrete state space, state variables need not have a numerical meaning
- Logical time domain defined by the events (order but not metric)
- Dynamics defined by transition tables: input event a takes the system from state s to state s'

Automata as Dynamical Systems

- Abstract discrete state space, state variables need not have a numerical meaning
- Logical time domain defined by the events (order but not metric)
- Dynamics defined by transition tables: input event a takes the system from state s to state s'
- Behaviors are sequences of states and events

Automata as Dynamical Systems

- Abstract discrete state space, state variables need not have a numerical meaning
- Logical time domain defined by the events (order but not metric)
- Dynamics defined by transition tables: input event a takes the system from state s to state s'
- Behaviors are sequences of states and events
- Composition of large systems from small ones
- Different modes of interaction: synchronous/asynchronous, state-based/event-based

Automata can Model many Phenomena and Devices

- Software, hardware,
- ATMs, user interfaces
- Administrative procedures
- Communication protocols
- Cooking recipes, Manufacturing instructions

Automata can Model many Phenomena and Devices

- Software, hardware,
- ATMs, user interfaces
- Administrative procedures
- Communication protocols
- Cooking recipes, Manufacturing instructions
- Any process that can be viewed as a sequence of steps

Automata can Model many Phenomena and Devices

- Software, hardware,
- ATMs, user interfaces
- Administrative procedures
- Communication protocols
- Cooking recipes, Manufacturing instructions
- Any process that can be viewed as a sequence of steps
- But what can we do with these models?
- There are no analytical tools as in continuous systems
- We can simulate and sometimes do formal verification

What is Verification ?

- Given a complex discrete dynamical systems with some uncontrolled inputs or unknown parameters
- Check whether ALL its behaviors satisfy some properties

What is Verification ?

- Given a complex discrete dynamical systems with some uncontrolled inputs or unknown parameters
- Check whether ALL its behaviors satisfy some properties
- Properties:
  - Never reach some part of the state space
  - Always come eventually to some (equilibrium) state
  - Never exhibit some pattern of behavior
  - Quantitative properties..

What is Verification ?

- Given a complex discrete dynamical systems with some uncontrolled inputs or unknown parameters
- Check whether ALL its behaviors satisfy some properties
- Properties:
  - Never reach some part of the state space
  - Always come eventually to some (equilibrium) state
  - Never exhibit some pattern of behavior
  - Quantitative properties..
- Existing tools can do this type of analysis for huge systems by sophisticated graph algorithms

Illustration: The Coffee Machine

- Consider a machine that takes money and distributes drinks
- Modern (artificial) systems are decomposed into physical part and information processing part

Illustration: The Coffee Machine

- Consider a machine that takes money and distributes drinks
- Modern (artificial) systems are decomposed into physical part and information processing part
- The physical interface takes care of transforming sensors and actuators to/from information
- For example, detect coin insertion or button pressing, initiate water heating

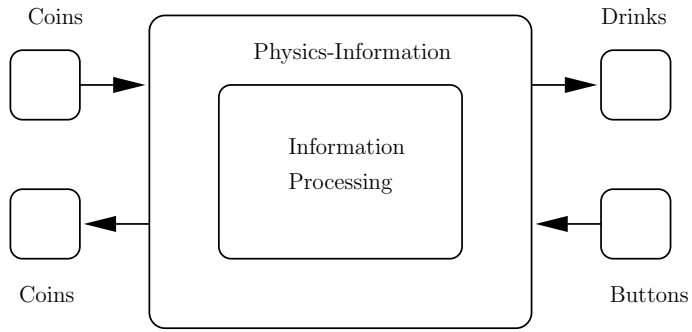
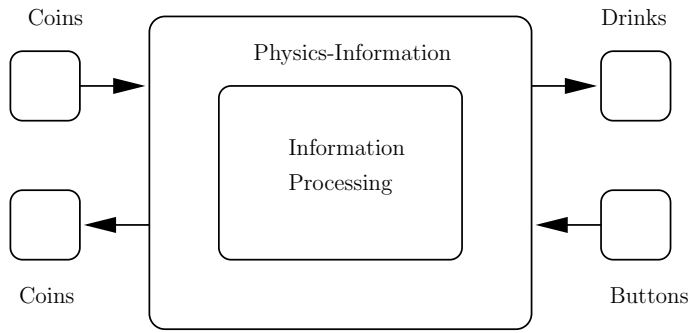


Illustration: The Coffee Machine

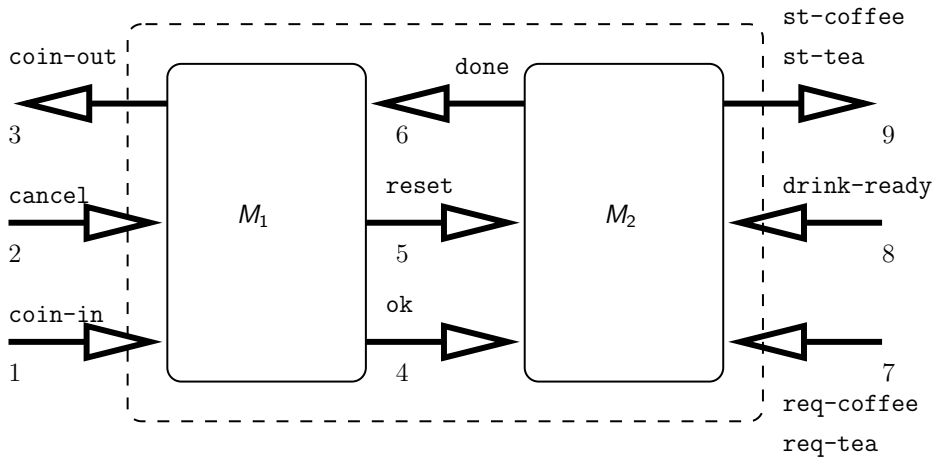
- Consider a machine that takes money and distributes drinks
- Modern (artificial) systems are decomposed into physical part and information processing part
- The physical interface takes care of transforming sensors and actuators to/from information
- For example, detect coin insertion or button pressing, initiate water heating



- In the information-processing system there are only events and reactions regardless of their physical origin

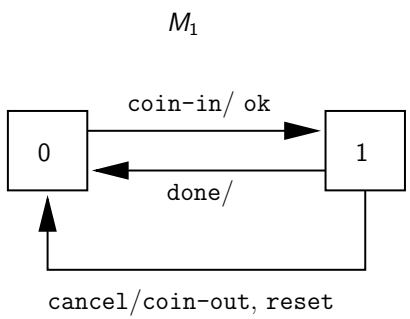
The Coffee Machine: Subsystems

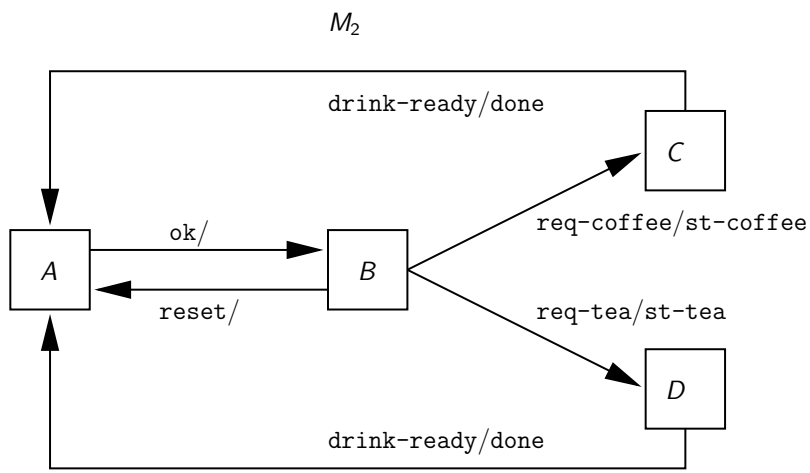
- The system is built from two subsystems, one that takes care of financial matters, and one which handles choice and preparation of drinks
- They communicate by sending messages



Automaton Models

- The two systems are models as automata (state-transition systems)
- transitions are triggered by external events and events coming from the other subsystem



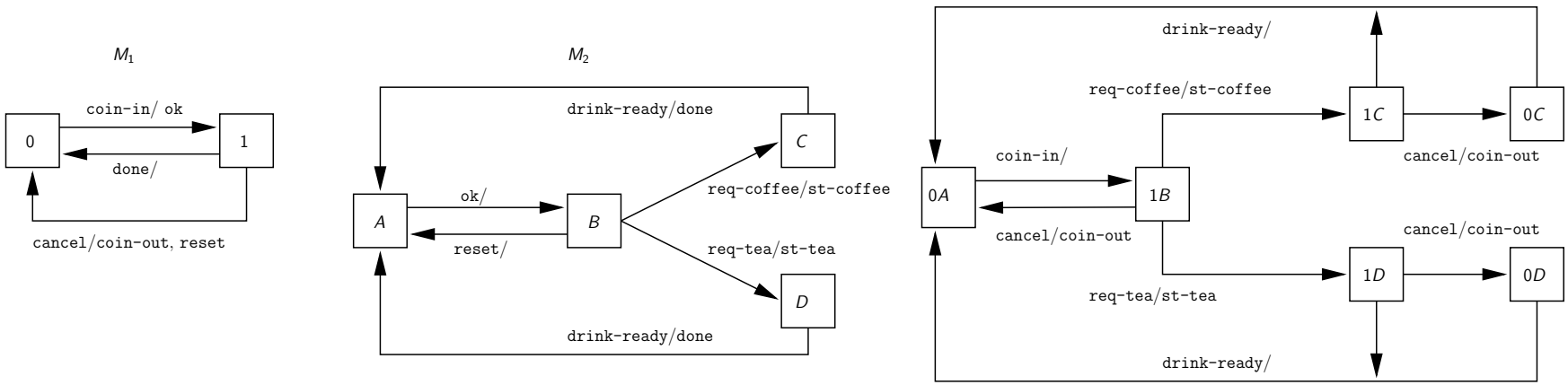


The Global Model

- The behavior of the whole system is captured by a composition (product) $M_1 \parallel M_2$ of the components
- States are elements of the Cartesian product of the respective sets of states, indicating the state of each component

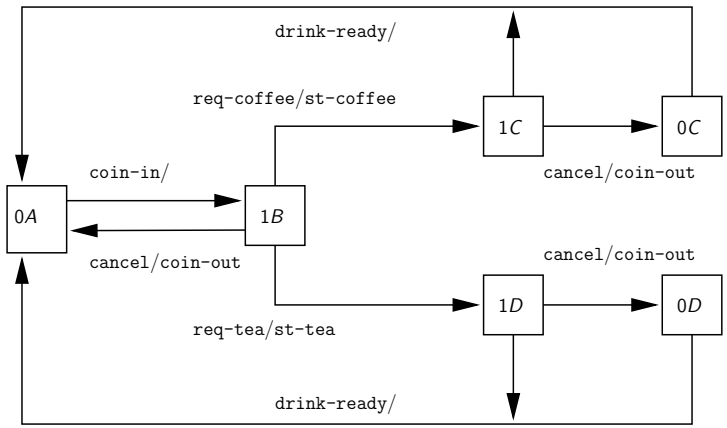
The Global Model

- The behavior of the whole system is captured by a composition (product) $M_1 \parallel M_2$ of the components
- States are elements of the Cartesian product of the respective sets of states, indicating the state of each component
- Some transitions are independent and some are synchronized, taken by the two components simultaneously



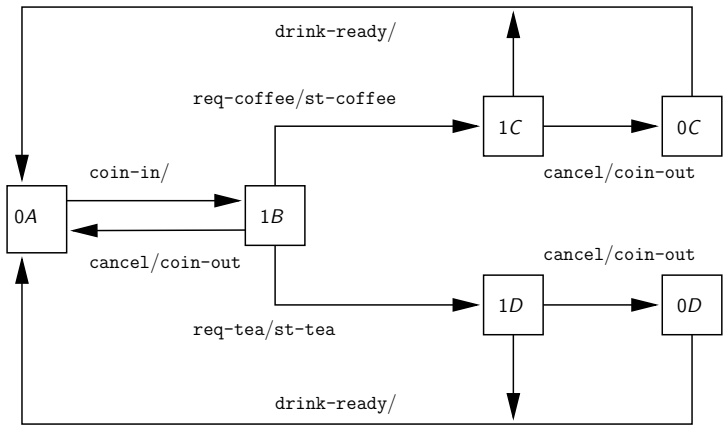
Normal Behaviors

- Behaviors of the systems are paths in this transition graph



Normal Behaviors

- Behaviors of the systems are paths in this transition graph

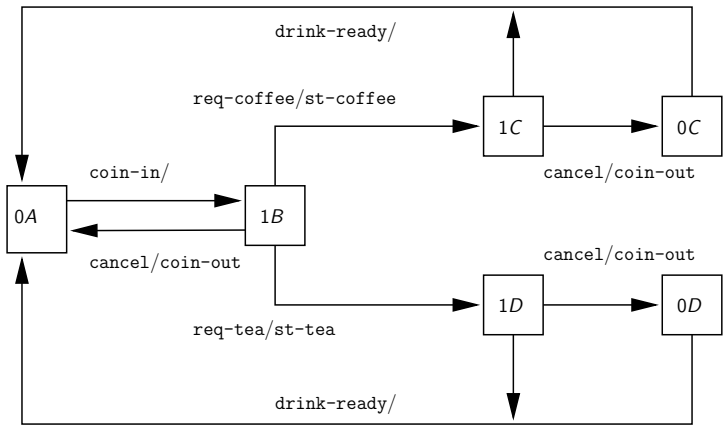


- Customer puts coin, then sees the bus arriving, cancels and gets the coin back

$$0A \texttt{ coin-in } 1B \texttt{ cancel coin-out } 0A$$

Normal Behaviors

- Behaviors of the systems are paths in this transition graph



- Customer puts coin, then sees the bus arriving, cancels and gets the coin back

$$0A \texttt{ coin-in } 1B \texttt{ cancel coin-out } 0A$$

- Customer inserts coin, requests coffee, gets it and the systems returns to initial state

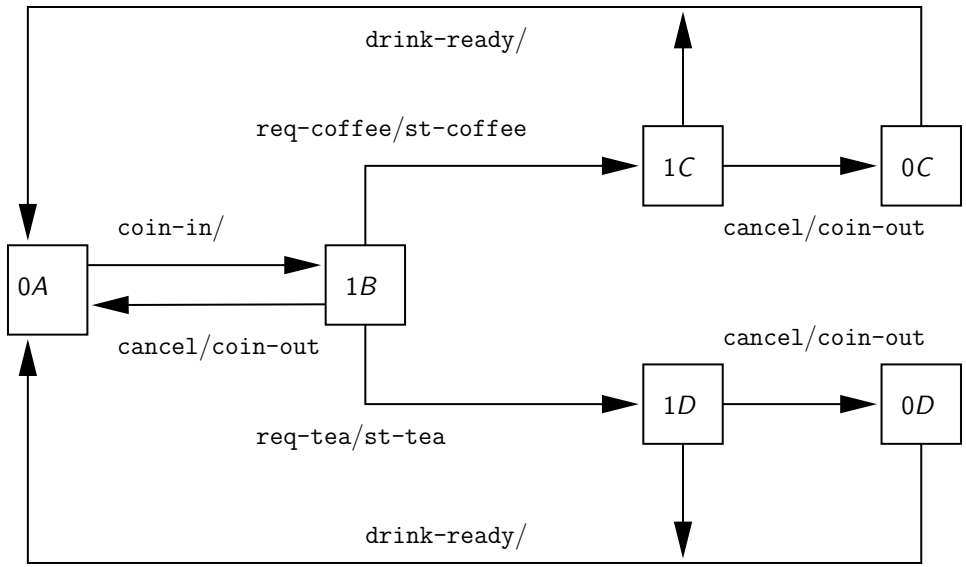
$$0A \texttt{ coin-in } 1B \texttt{ req-coffee st-coffee } 1C \texttt{ drink-ready } 0A$$

An Abnormal Behavior

- Suppose the customer presses the cancel button *after* the coffee starts being prepared..

An Abnormal Behavior

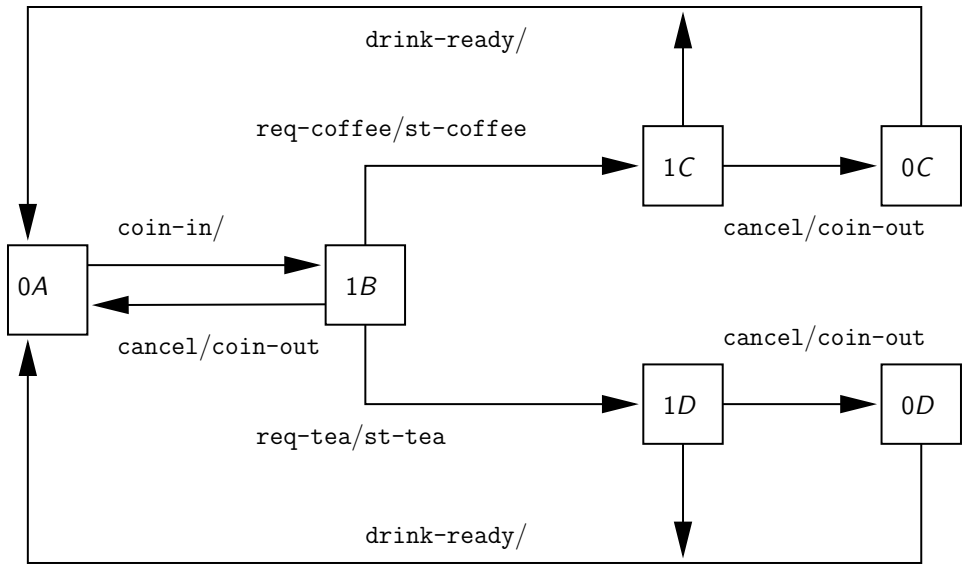
- Suppose the customer presses the cancel button *after* the coffee starts being prepared..



$0A$ `coin-in` $1B$ `req-coffee` `st-coffee` $1C$ `cancel` `coin-out` $0C$ `drink-ready` $0A$

An Abnormal Behavior

- Suppose the customer presses the cancel button *after* the coffee starts being prepared..

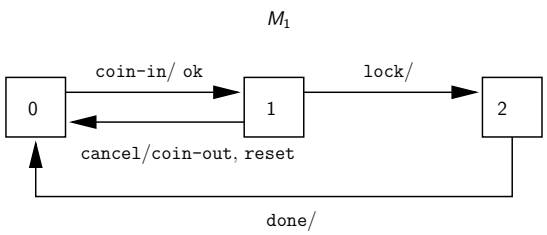


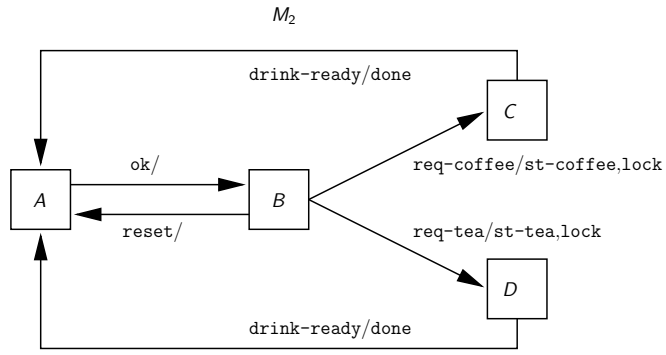
$0A$ `coin-in` $1B$ `req-coffee` `st-coffee` $1C$ `cancel` `coin-out` $0C$ `drink-ready` $0A$

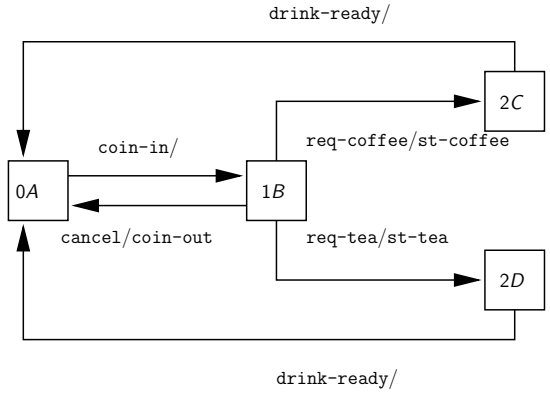
- Not so attractive for the owner of the machine

Fixing the Bug

- When M_2 starts preparing coffee it emits a `lock` signal
- When M_1 received this message it enters a new state where `cancel` is refused







The Moral of the Story

- Many systems can be modeled as a composition of interacting automata
- Behaviors of the system correspond to paths in the global transition graph of the system

The Moral of the Story

- Many systems can be modeled as a composition of interacting automata
- Behaviors of the system correspond to paths in the global transition graph of the system
- The size of this graph is exponential in the number of components (state explosion, curse of dimensionality)
- These paths are labeled by input events representing influences of the outside environment

The Moral of the Story

- Many systems can be modeled as a composition of interacting automata
- Behaviors of the system correspond to paths in the global transition graph of the system
- The size of this graph is exponential in the number of components (state explosion, curse of dimensionality)
- These paths are labeled by input events representing influences of the outside environment
- Each input sequence may generate a different behavior
- We want to make sure that a system responds correctly to all conceivable inputs, that it behaves properly in any environment

The Moral of the Story

- We want to make sure that a system responds correctly to all conceivable inputs, that it behaves properly in any environment
- For every individual input sequence we can simulate the reaction of the system. But we cannot do it exhaustively
- Verification is a collection of automatic and semi-automatic methods to analyze all the paths in the graph
- This is hard for humans to do and even for computers

Hybrid Systems: Motivation

- Hybrid systems combine the discrete dynamics of automata with continuous dynamics defined by differential equations
- Each state may correspond to a mode of a system (a gene is on, a valve/heater is closed, the car is in a second gear)

Hybrid Systems: Motivation

- Hybrid systems combine the discrete dynamics of automata with continuous dynamics defined by differential equations
- Each state may correspond to a mode of a system (a gene is on, a valve/heater is closed, the car is in a second gear)
- In each state there is a different continuous dynamics
- The system may switch between modes according to the values of the continuous variables
- For example, the heater is turned off when temperature is high, a valve is opened when the water level crosses a threshold

Hybrid Systems Analysis is Difficult

- Purely continuous systems (especially linear ones) admit a lot of mathematical analysis techniques
- Hybrid systems are much harder to analyze because switching breaks their nice mathematical properties

Hybrid Systems Analysis is Difficult

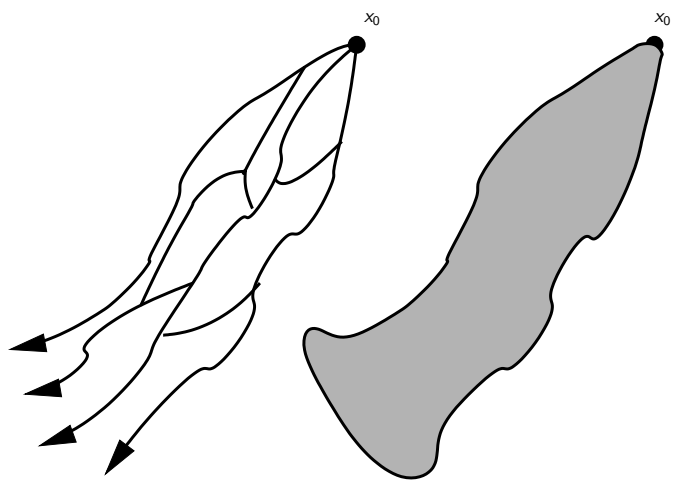
- Purely continuous systems (especially linear ones) admit a lot of mathematical analysis techniques
- Hybrid systems are much harder to analyze because switching breaks their nice mathematical properties
- New techniques inspired by discrete verification are being developed
- Combination of numerical analysis, graph algorithms and computational geometry

Verification for Continuous Systems

- The problem: a dynamical system $\dot{x} = f(x, p, u)$ where u is an external disturbance and p is a parameter
- Both u and p are not known exactly but are bounded

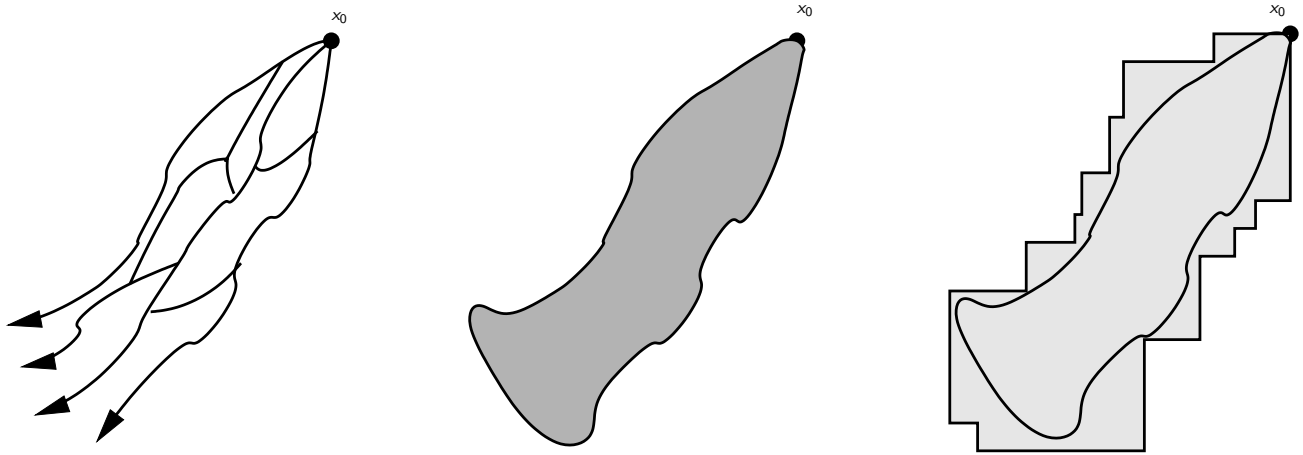
Verification for Continuous Systems

- The problem: a dynamical system $\dot{x} = f(x, p, u)$ where u is an external disturbance and p is a parameter
- Both u and p are not known exactly but are bounded
- Can something be said about all the possible behaviors of the system for all range of parameters and all external disturbances?



Verification for Continuous Systems

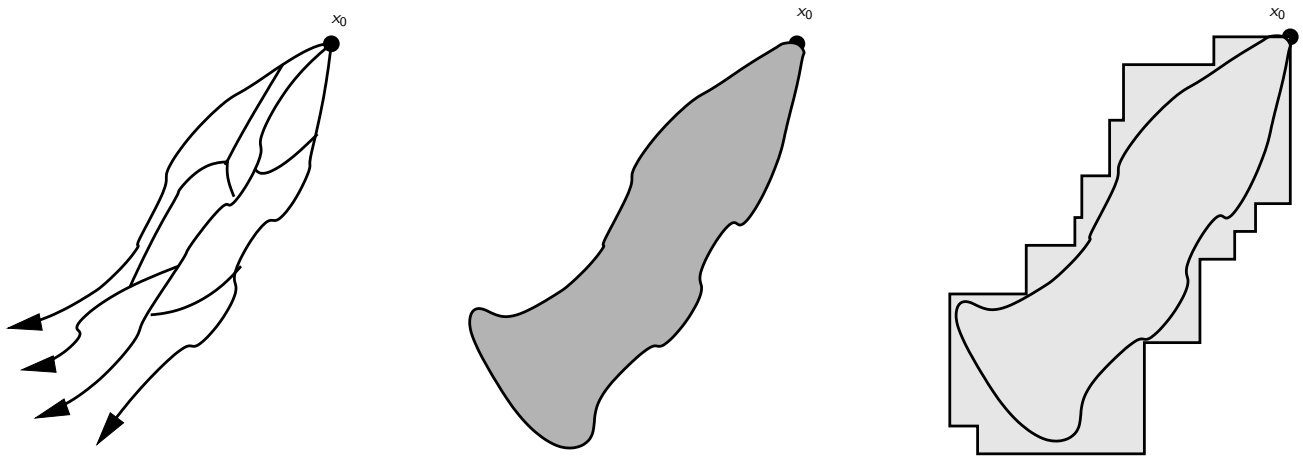
- A kind of set-based numerical integration to approximate the set of states reachable by all possible inputs and parameters



- Can replace an infinite number of simulations

Verification for Continuous Systems

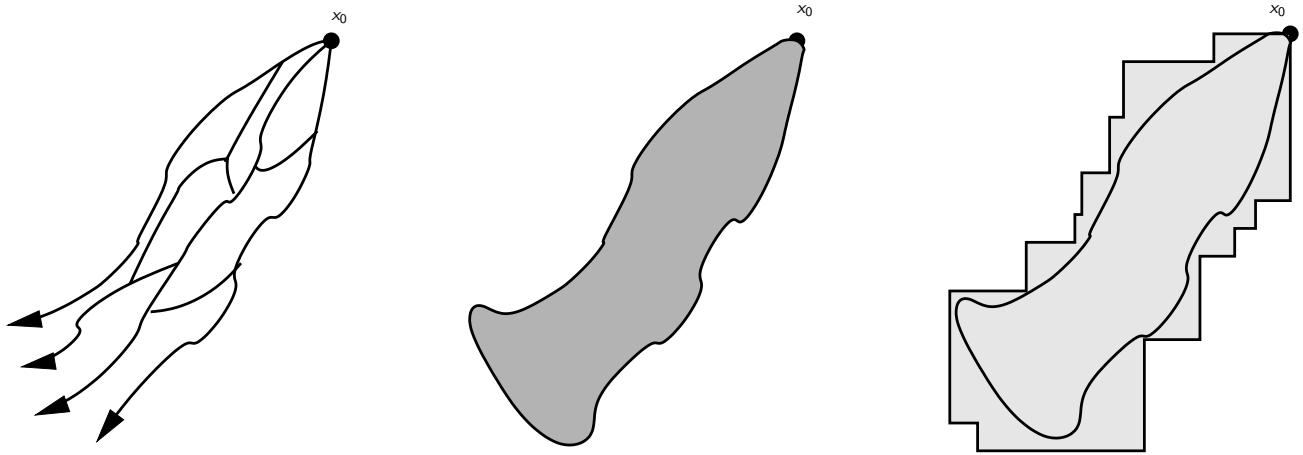
- A kind of set-based numerical integration to approximate the set of states reachable by all possible inputs and parameters



- Can replace an infinite number of simulations
- Useful for Biological models where exact parameters are hard or impossible to obtain

Verification for Continuous Systems

- A kind of set-based numerical integration to approximate the set of states reachable by all possible inputs and parameters



- Can replace an infinite number of simulations
- Useful for Biological models where exact parameters are hard or impossible to obtain
- State-of the-art: tools at various levels of sophistication and maturity can analyze linear systems with hundreds of state variables, as well as small nonlinear ones

Timed System

- An important level of abstraction between the discrete and the continuous

Timed System

- An important level of abstraction between the discrete and the continuous
- Continuous description: how the concentration of some product evolves over time

Timed System

- An important level of abstraction between the discrete and the continuous
- Continuous description: how the concentration of some product evolves over time
- Discrete description: the product levels moves from low to high

Timed System

- An important level of abstraction between the discrete and the continuous
- Continuous description: how the concentration of some product evolves over time
- Discrete description: the product levels moves from low to high
- Timed description: the product levels moves from low to high and this process takes between 3 and 5 hours to complete

Timed System

- An important level of abstraction between the discrete and the continuous
- Continuous description: how the concentration of some product evolves over time
- Discrete description: the product levels moves from low to high
- Timed description: the product levels moves from low to high and this process takes between 3 and 5 hours to complete
- At this level the dynamical models are timed automata, automata with auxiliary clock variables
- Illustration: adding time to the discrete models of R. Thomas (see also Siebert and Bockmayr)

Genetic Regulatory Networks for (and by) Dummies

- A set $G = \{g_1, \ldots, g_n\}$ of genes
- A set $P = \{p_1, \ldots, p_n\}$ of products (proteins)

Genetic Regulatory Networks for (and by) Dummies

- A set $G = \{g_1, \ldots, g_n\}$ of genes
- A set $P = \{p_1, \ldots, p_n\}$ of products (proteins)
- Each gene is responsible for the production of one product

Genetic Regulatory Networks for (and by) Dummies

- A set $G = \{g_1, \ldots, g_n\}$ of genes
- A set $P = \{p_1, \ldots, p_n\}$ of products (proteins)
- Each gene is responsible for the production of one product
- Genes are viewed as Boolean variables (On/Off)
- When $g_i = 1$ it will tend to increase the quantity of p_i
- When $g_i = 0$ the quantity of p_i will decrease (degradation)

Genetic Regulatory Networks for (and by) Dummies

- A set $G = \{g_1, \ldots, g_n\}$ of genes
- A set $P = \{p_1, \ldots, p_n\}$ of products (proteins)
- Each gene is responsible for the production of one product
- Genes are viewed as Boolean variables (On/Off)
- When $g_i = 1$ it will tend to increase the quantity of p_i
- When $g_i = 0$ the quantity of p_i will decrease (degradation)
- Feedback from products concentrations to genes: when the quantity of a product is below/above some threshold it may set one or more genes on or off

Continuous and Discrete Models

- Product quantities can be viewed as integer (quantity) or real (concentration of molecules in the cell) numbers
- The system can be viewed as a hybrid automaton with discrete states corresponding to combinations of gene activations states
- The evolution of product concentrations can be described using differential equations on concentration

Continuous and Discrete Models

- Product quantities can be viewed as integer (quantity) or real (concentration of molecules in the cell) numbers
- The system can be viewed as a hybrid automaton with discrete states corresponding to combinations of gene activations states
- The evolution of product concentrations can be described using differential equations on concentration
- Alternatively, the domain of these concentrations can be discretized into a finite (and small) number of ranges
- The most extreme of these discretizations is to to consider a Boolean domain $\{0, 1\}$ indicating *present* or *absent*

The Discrete Model of R. Thomas

- Gene activation is specified as a Boolean function over the presence/absence of products
- When a gene changes its value, its corresponding product will follow within some unspecified delay

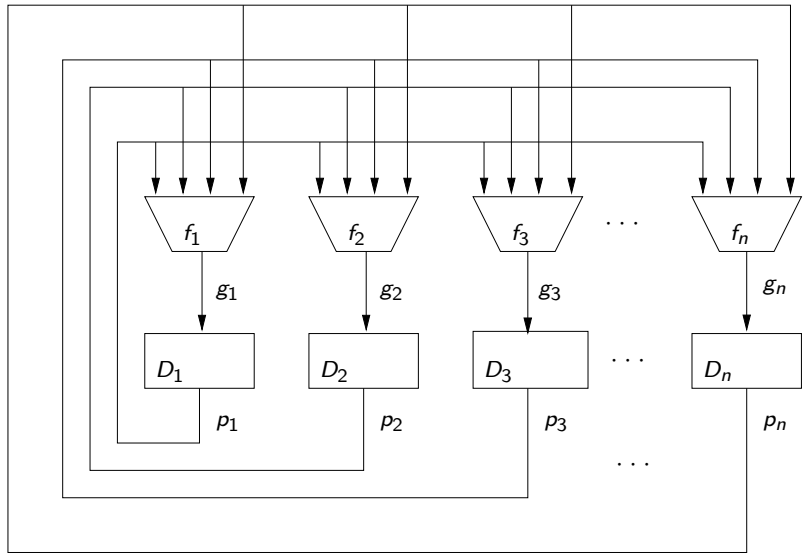
The Discrete Model of R. Thomas

- Gene activation is specified as a Boolean function over the presence/absence of products
- When a gene changes its value, its corresponding product will follow within some unspecified delay
- The resulting model is equivalent to an asynchronous automaton
- The relative speeds of producing different products are not modeled
- The model admits many behaviors which are not possible if these speeds are taken into account

The Discrete Model of R. Thomas

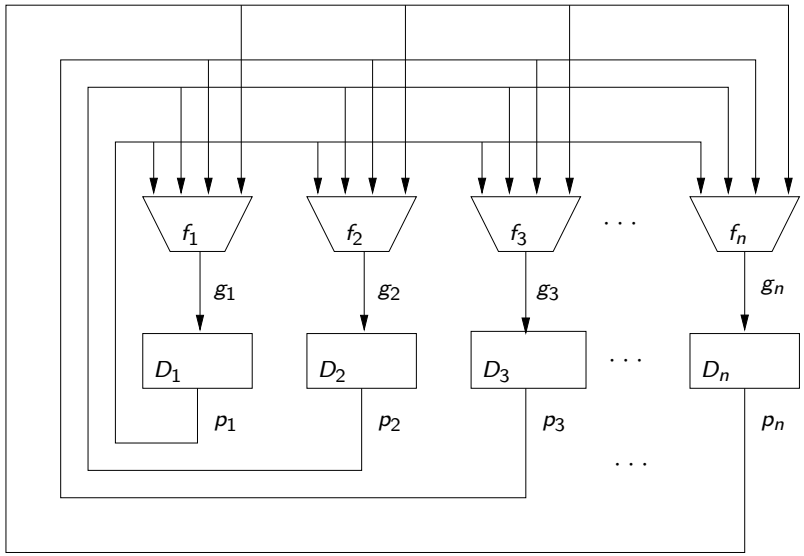
- Gene activation is specified as a Boolean function over the presence/absence of products
- When a gene changes its value, its corresponding product will follow within some unspecified delay
- The resulting model is equivalent to an asynchronous automaton
- The relative speeds of producing different products are not modeled
- The model admits many behaviors which are not possible if these speeds are taken into account
- We want to add this timing information in a systematic manner as we did in the past for asynchronous digital circuits [Maler and Pnueli 95]

Boolean Delay Networks



- A change in the activation of a gene is considered instantaneous once the value of f has changed

Boolean Delay Networks



- A change in the activation of a gene is considered instantaneous once the value of f has changed
- This change is propagated to the product within a non-deterministic but bi-bounded delay specified by an interval

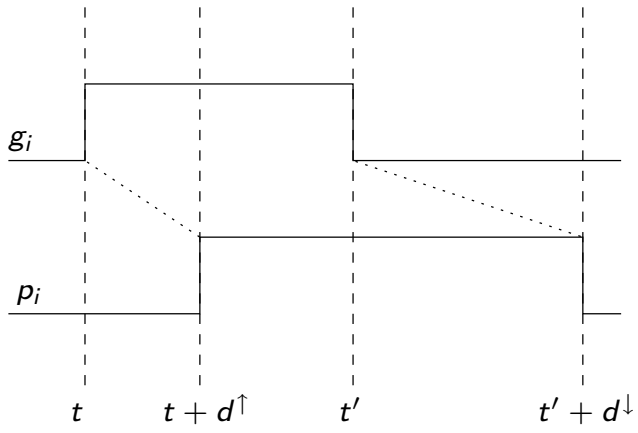
The Delay Operator

- For each i we define a delay operator D_i, a function from Boolean signals to Boolean signals characterized by 4 parameters

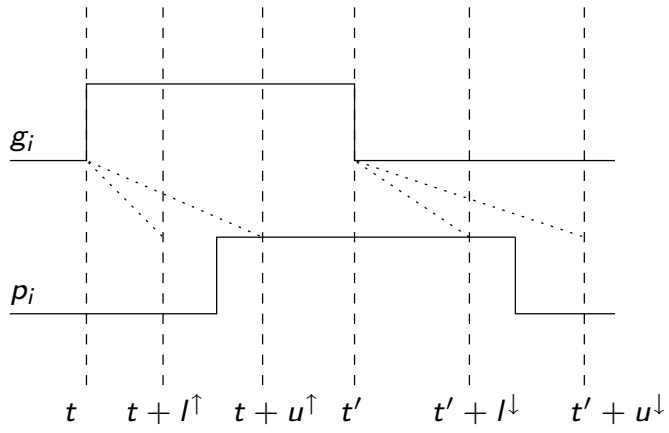
p_i	g_i	p_i'	Δ
0	0	0	–
0	1	1	$[l^{\uparrow}, u^{\uparrow}]$
1	0	0	$[l^{\downarrow}, u^{\downarrow}]$
1	1	1	–

- When $p_i \neq g_i$, p_i will catch up with g_i within $t \in [l^{\uparrow}, u^{\uparrow}]$ (rising) or $t \in [l^{\downarrow}, u^{\downarrow}]$ (falling)

The Delay Operator

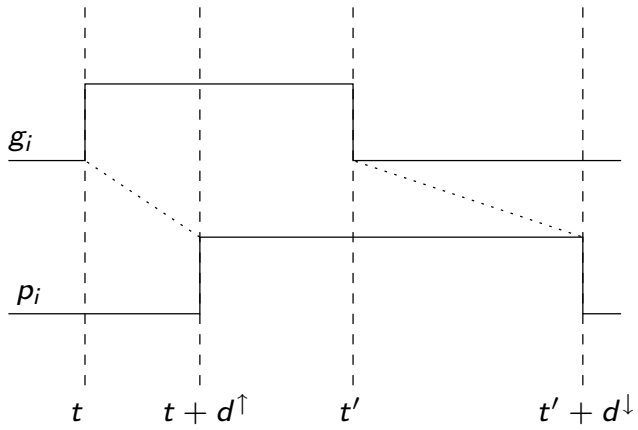


Determinisitc

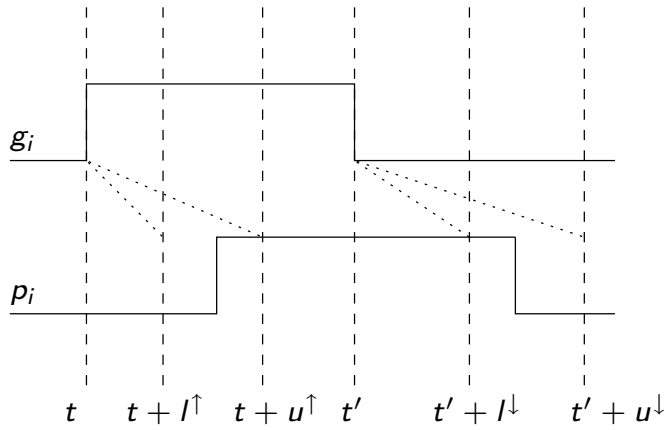


Nondeterministic

The Delay Operator



Determinisitc



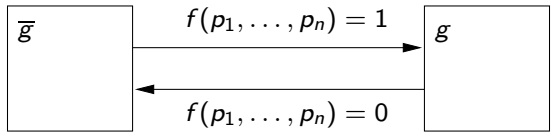
Nondeterministic

- The semantics of the network is the set of all Boolean signals satisfying the following set of signal inclusions

$$g_i = f_i(p_1, \ldots, p_n)$$
$$p_i \in D_i(g_i)$$

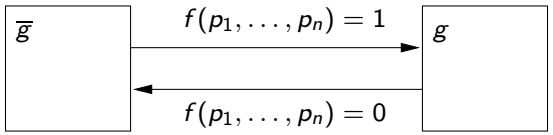
Modeling with Timed Automata

- For each equation $g_i = f_i(p_1, \ldots, p_n)$ we build the automaton

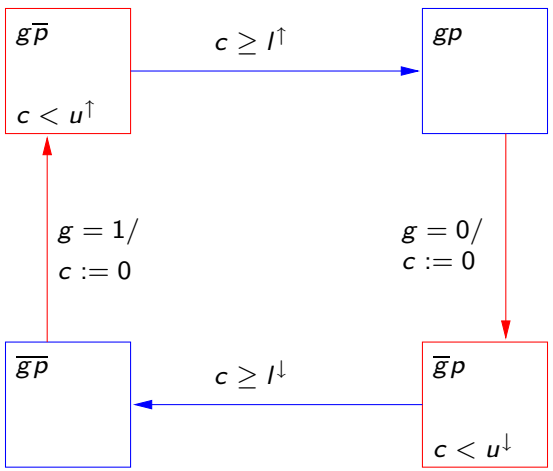


Modeling with Timed Automata

- For each equation $g_i = f_i(p_1, \ldots, p_n)$ we build the automaton

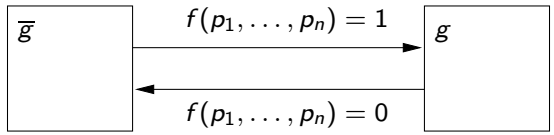


- For each delay inclusion $p_i \in D_i(g_i)$ we build the automaton

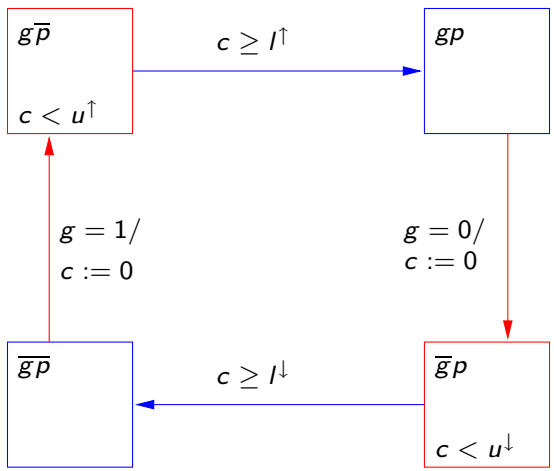


Modeling with Timed Automata

- For each equation $g_i = f_i(p_1, \ldots, p_n)$ we build the automaton



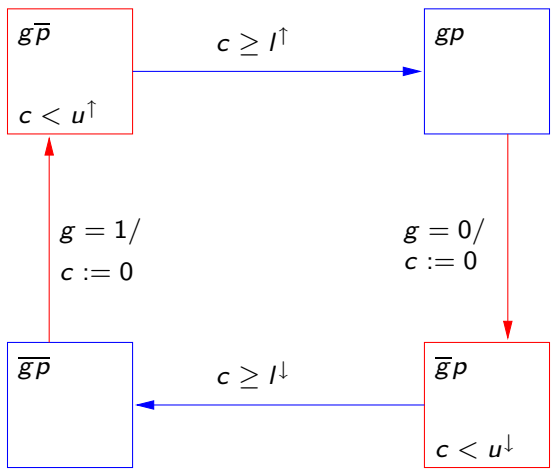
- For each delay inclusion $p_i \in D_i(g_i)$ we build the automaton



- Composing these automata together we obtain a timed automaton whose semantics coincides with that of the system of signal inclusions

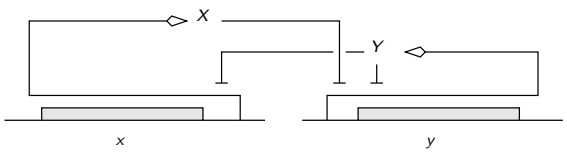
The Delay Automaton

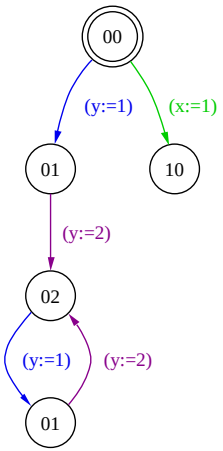
- The automaton has two stable states gp and $\overline{gp}$ where the gene and the product agree
- When g changes (excitation) it moves to the unstable state and reset a clock to zero
- It can stay in an unstable state as long as $c < u$ and can stabilize as soon as $c > l$.



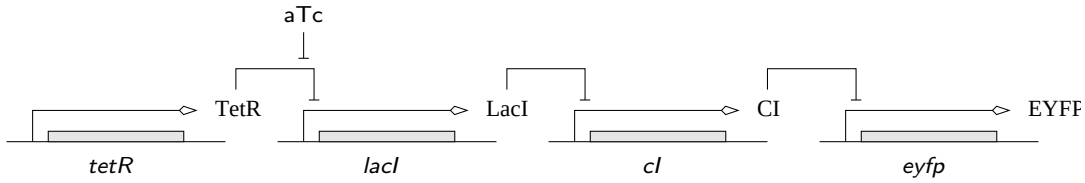
Implementation and Experiments

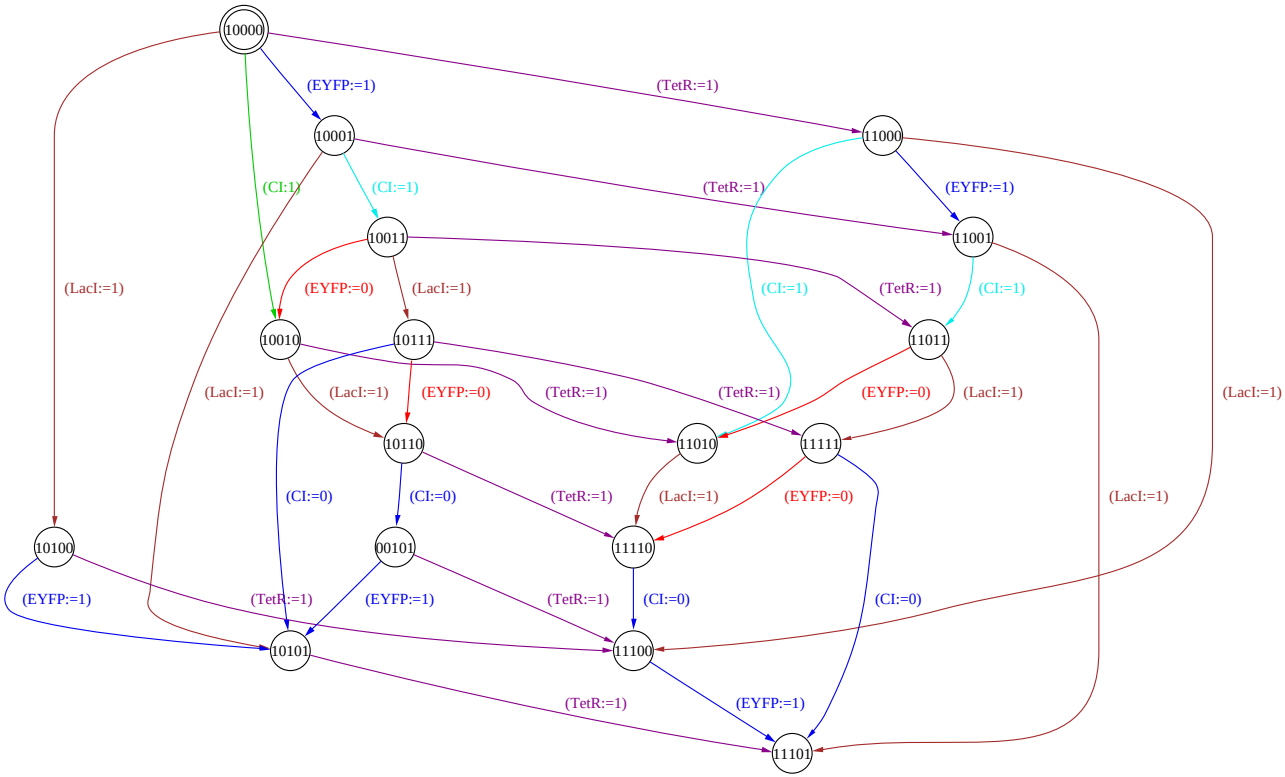
- Existing tools can take a description of such a timed automaton and compute all the possible behaviors under all choices of delays
- We use our IF toolbox and demonstrate its capabilities on several examples (not much biological significance at this point)
- Example 1: a cross inhibition network (also modeled by [Siebert and Bockmayr 06])





Transcription Cascade for E. Coli



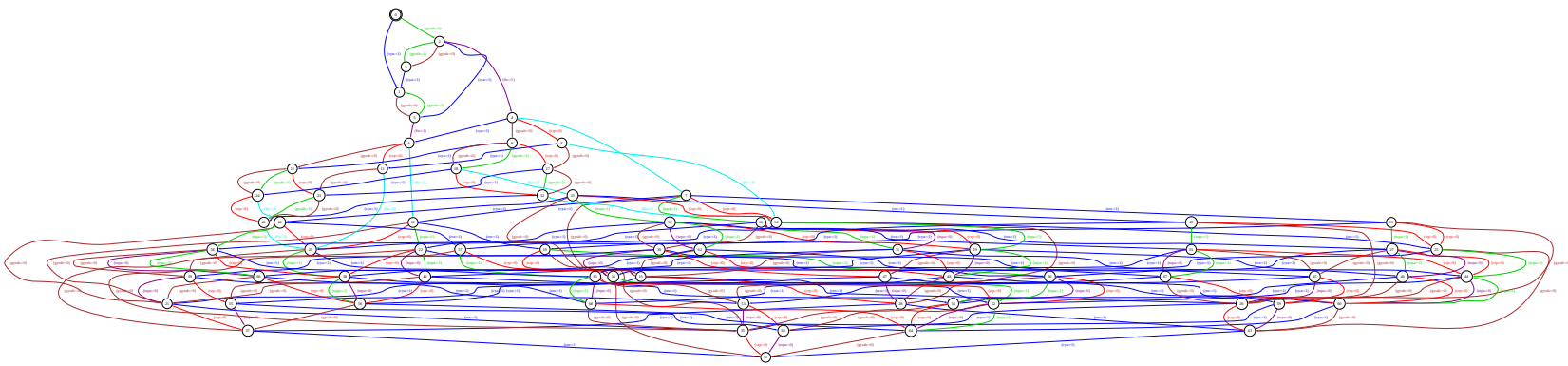


Nutritional Stress Response for E. Coli

- Six genes, six proteins and one additional variable encoding the presence or absence of nutrition
- Arbitrarily chosen delays, cyclic behaviors
- Reachability graph with 69 states and 209 transitions, computed in less than one second

Nutritional Stress Response for E. Coli

- Six genes, six proteins and one additional variable encoding the presence or absence of nutrition
- Arbitrarily chosen delays, cyclic behaviors
- Reachability graph with 69 states and 209 transitions, computed in less than one second



Back to the Big Picture

- Biology needs (among other things) more dynamic models to form verifiable predictions

Back to the Big Picture

- Biology needs (among other things) more dynamic models to form verifiable predictions
- These models can benefit from the accumulated understanding of dynamical system within informatics and cannot rely only on 19th century mathematics
- The views of dynamical system developed within informatics is, sometimes, more adapted to the complexity and heterogeneity of Biological phenomena

Back to the Big Picture

- Biology needs (among other things) more dynamic models to form verifiable predictions
- These models can benefit from the accumulated understanding of dynamical system within informatics and cannot rely only on 19th century mathematics
- The views of dynamical system developed within informatics is, sometimes, more adapted to the complexity and heterogeneity of Biological phenomena
- Biological modeling should be founded on various types of dynamical models: continuous, discrete, hybrid and timed

Back to the Big Picture

- Biology needs (among other things) more dynamic models to form verifiable predictions
- These models can benefit from the accumulated understanding of dynamical system within informatics and cannot rely only on 19th century mathematics
- The views of dynamical system developed within informatics is, sometimes, more adapted to the complexity and heterogeneity of Biological phenomena
- Biological modeling should be founded on various types of dynamical models: continuous, discrete, hybrid and timed
- These models should be strongly supported by computerized analysis tools offering a range of capabilities from simulation to verification and synthesis

Back to the Big Picture

- Systems Biology should combine insights from:

Back to the Big Picture

- Systems Biology should combine insights from:
- Engineering disciplines: modeling and analysis of very complex man-made systems (chips, control systems, software, networks, cars, airplanes, chemical plants)

Back to the Big Picture

- Systems Biology should combine insights from:
- Engineering disciplines: modeling and analysis of very complex man-made systems (chips, control systems, software, networks, cars, airplanes, chemical plants)
- Physics: experience in mathematical modeling of natural systems with measurement constraints

Back to the Big Picture

- Systems Biology should combine insights from:
- Engineering disciplines: modeling and analysis of very complex man-made systems (chips, control systems, software, networks, cars, airplanes, chemical plants)
- Physics: experience in mathematical modeling of natural systems with measurement constraints
- Mathematics and Informatics as a unifying theoretical framework

Thank You