

Symbolic Computation Algebraic Biology I

Bud Mishra

Courant Inst, NYU

NYU SoM, TIFR, MSSM

Systems Biology

- Introduction to Biology
- Regulatory & Metabolic Processes
- Algebraic Models in Biology

Symbolic Computation Algebraic Biology II

Bud Mishra

Courant Inst, NYU

NYU SoM, TIFR, MSSM

Model Checking

- Temporal Logic
- Kripke Models
- Model Checking
- Biologically Faithful Models

Symbolic Computation

Algebraic Biology III

Bud Mishra

Courant Inst, NYU

NYU SoM, TIFR, MSSM

Semi-Algebraic Geometry

- Real Closed Field
- Tarski Algebra
- Decision Theories
- Hybrid Models
- Algorithmic Algebraic Model

Symbolic Computation

Algebraic Biology IV

Bud Mishra

Courant Inst, NYU

NYU SoM, TIFR, MSSM

Hybrid Systems

- Hybrid Models
- Algorithmic Algebraic Models & Model Checking
- O-minimal Systems & SaCoRe
- IDA
- Open Problems

Desideratum

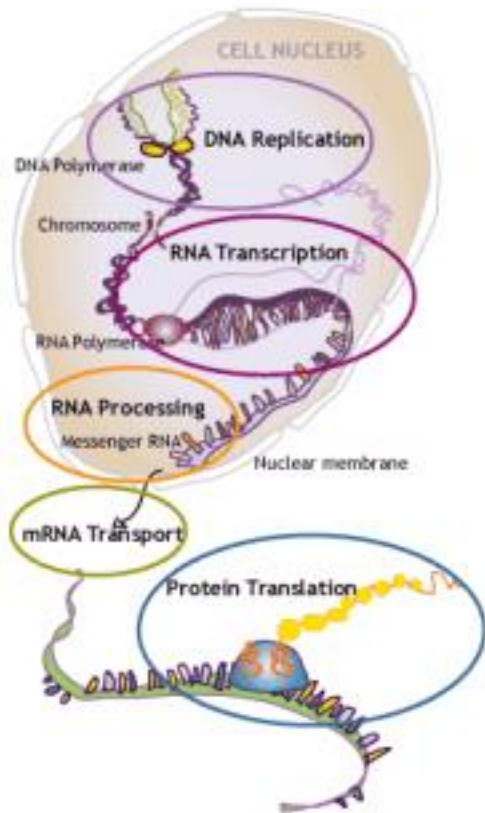
- We would like to answer questions like the following?
 - Given a description of a biological system as a (discrete+continuous) automaton (with some of the systems parameters known, but many unknown: c_1, c_2, \dots) and a temporal logic formula Φ ...
 - Are there specific relations among the unknowns that must be satisfied for Φ to be true

Biology

- Can we determine relations among the unknowns, C , to determine various properties of the biological system? E.g., $\text{Vol}(C) \approx \text{robustness}$? $\text{Reachable}(C) \cap \text{BadState}$?
- Invariants: For realistic values in C , what other properties Ψ hold true?
- Distinction: Given a wildtype, W and a mutant, M , are there properties Ψ_M true for mutant but false for wildtype?

Biological Regulatory Networks

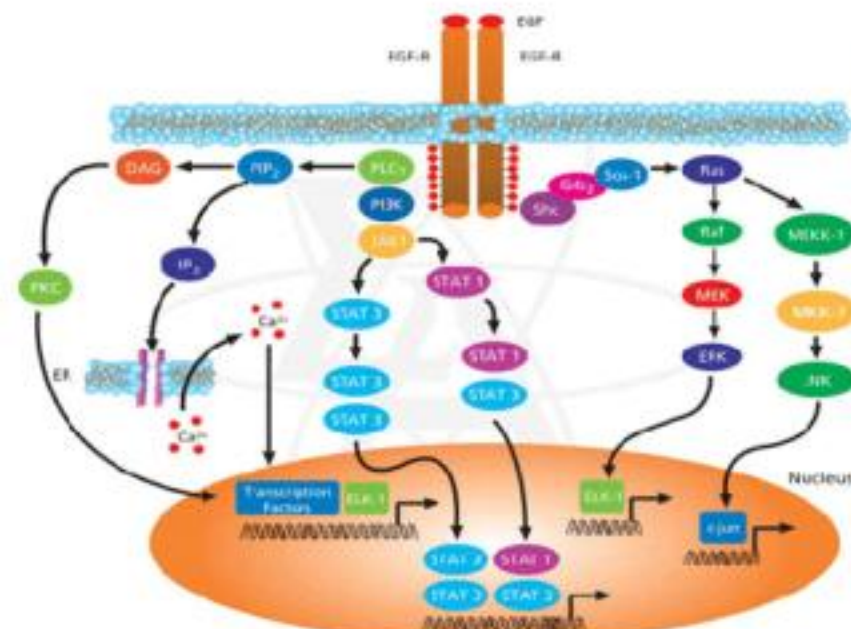
1. Genetic Regulation



- Players: DNA, RNA, Proteins
- Processes: Transcription, Translation
- Other factors: promoters, repressors, transcription factors, splicing...

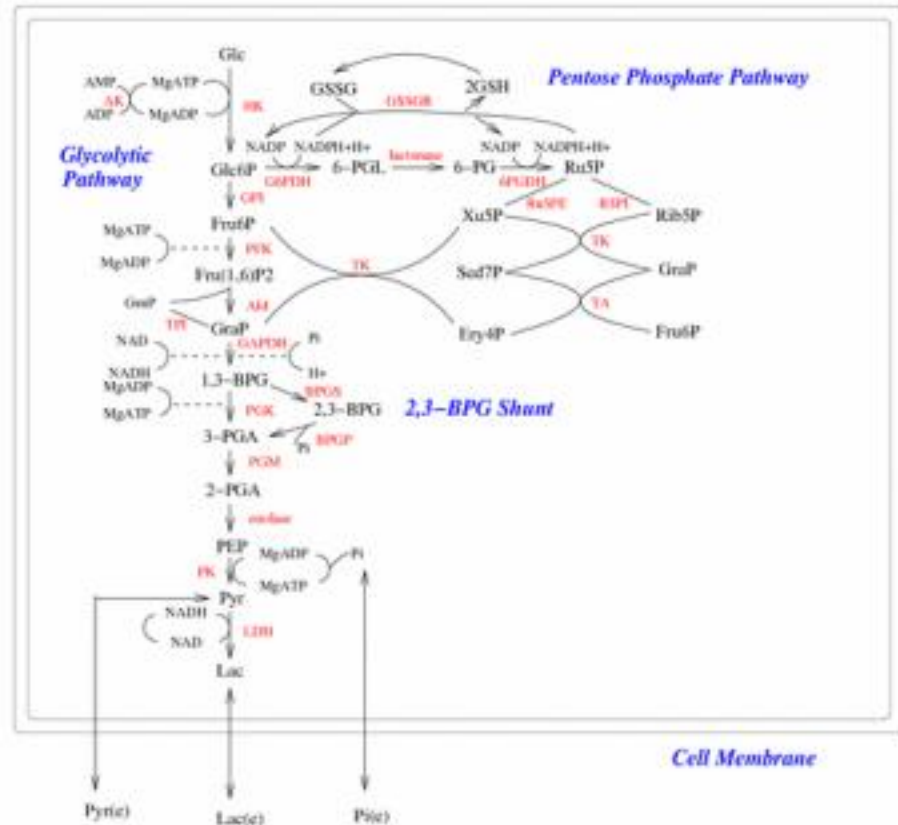
2. Signal Transduction

- Players: Receptors, Ligands, Second Messengers, Transcription factors...
- Processes: Binding of signaling molecule with receptor; ... Modification of structure of receptor ; ... Dispatching of second messengers; ... Entry into nucleus and gene regulation

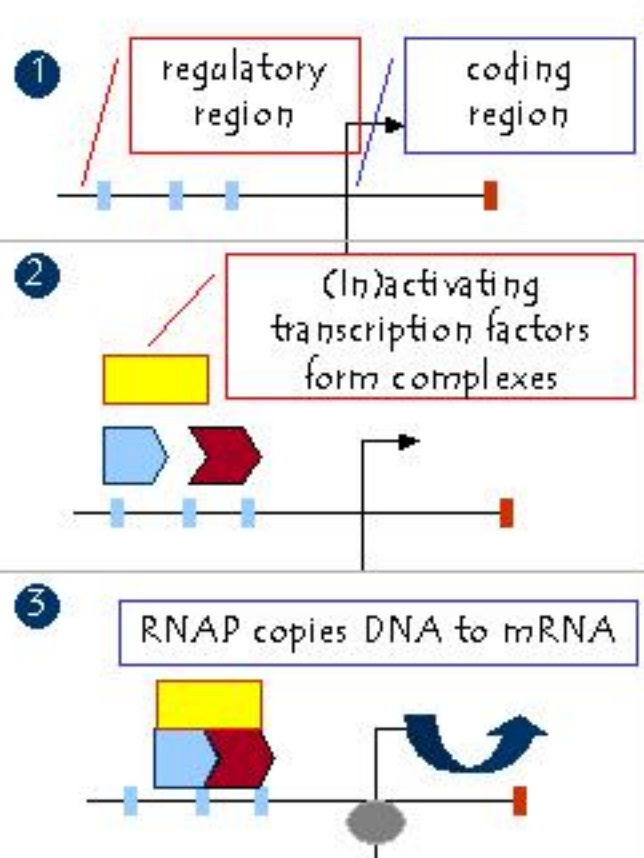


3. Metabolic Pathways

- Carbohydrate Metabolism: Glycolysis, Phosphate...
- Lipid Metabolism: Triacyl Glycerol, Fatty Acids...
- Amino Acid Metabolism: Glutamate, Urea...
- Energy Metabolism: ATP,

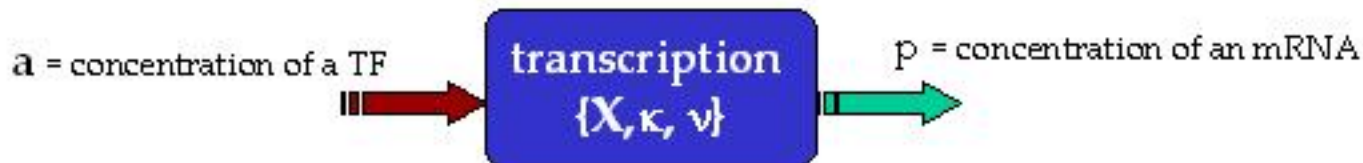


Details: Transcription Initiation



- Typically, TFs (Transcription Factors) do not bind singly, but in complexes:
- Once bound to the DNA, TF complex allows RNA polymerase (RNAP) to bind to the DNA upstream of the coding region.
- RNAP forms a transcriptional complex that separates the two strands of DNA, thus forming an open complex, and transcribes the coding region into mRNA.

Model of transcription

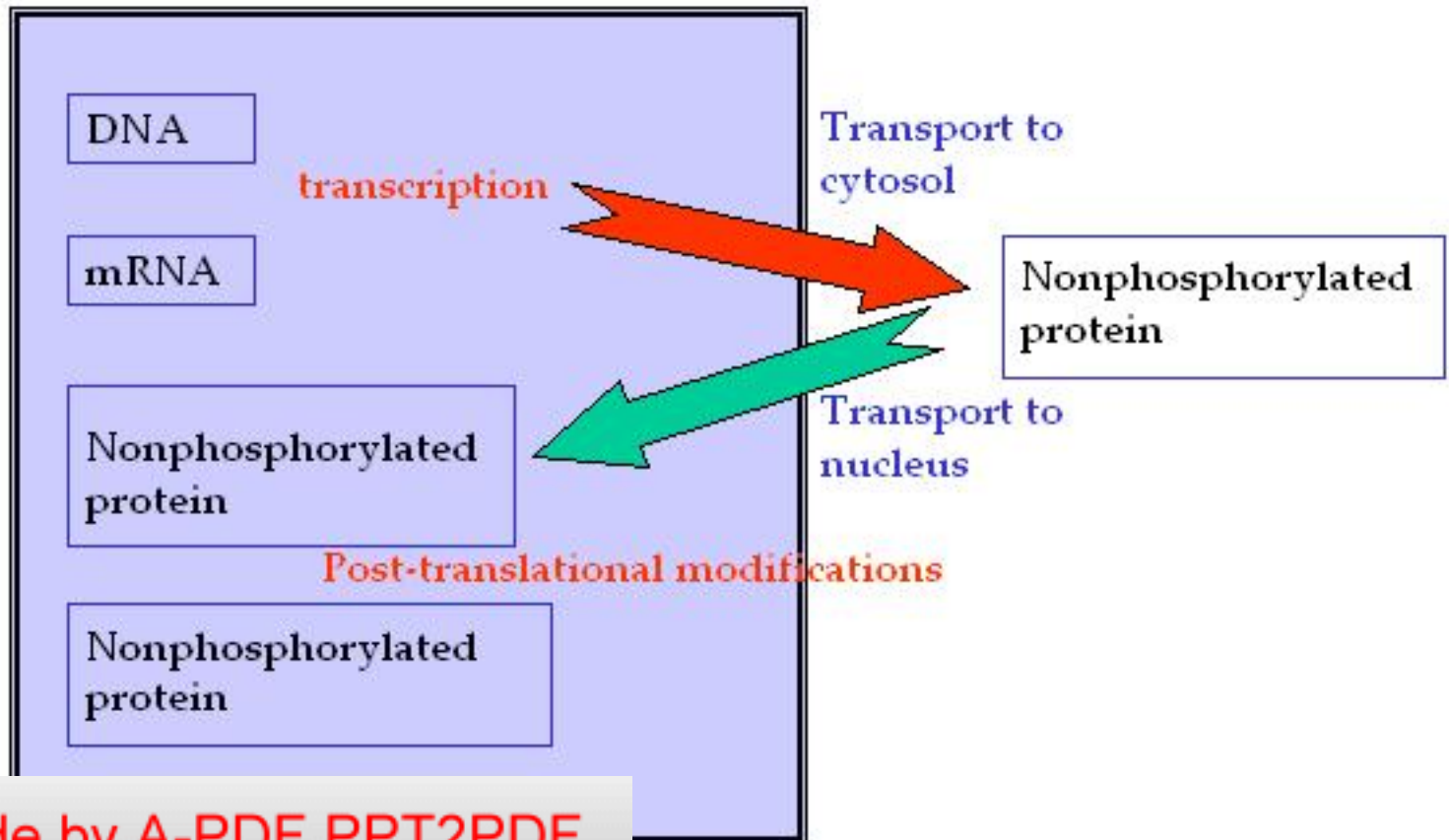


- v = Cooperativity coefficient
- κ = Concentration of a at which transcription of m is "half-maximally" activated.
- $dp/dt = \Phi(a, \kappa, v) = V a^v / [\kappa^v + a^v]$
- A graph of function Φ = Sigmoid Function
- If $v = 1$ then, the transcription activation function resembles the classical Michaelis-Menten.

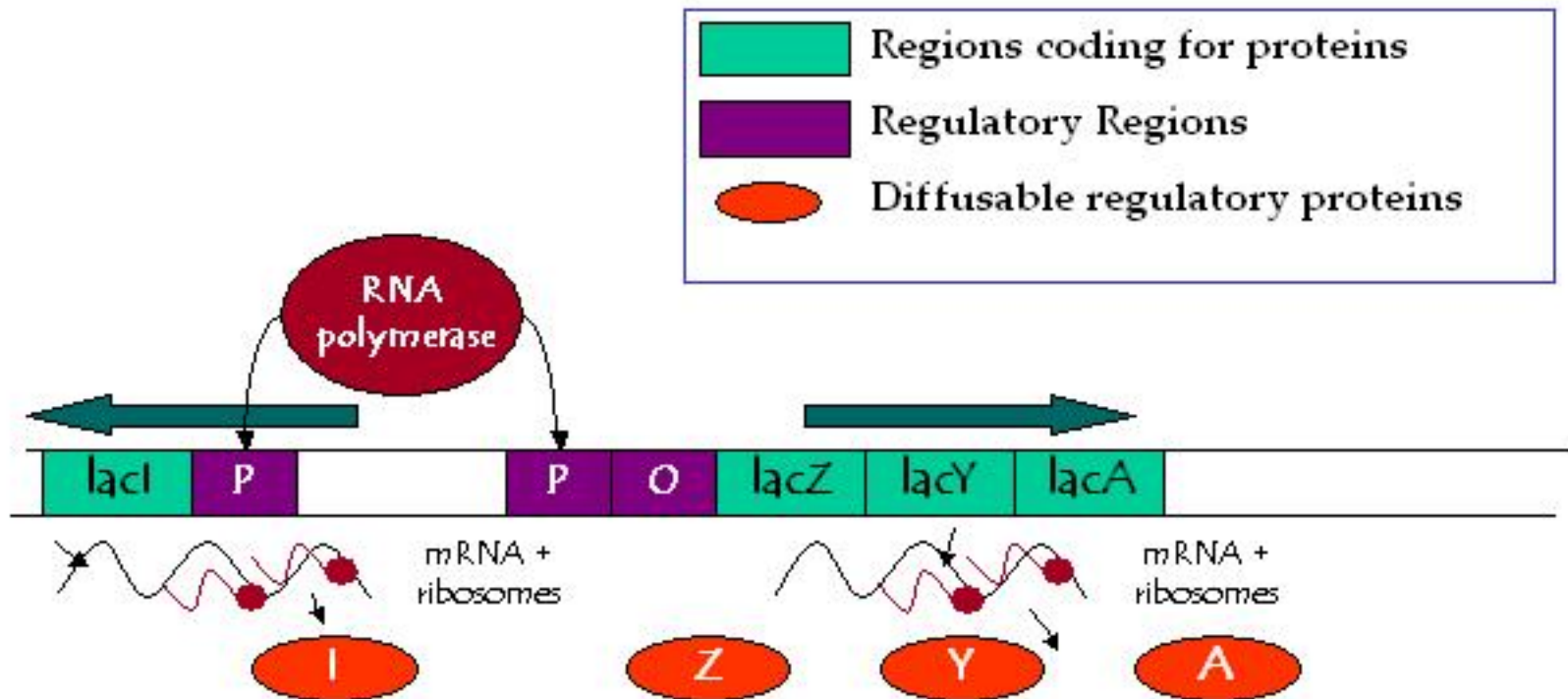
Regulatory Networks

- All cells in an organism have the same genomic data, but the proteins synthesized in each vary according to cell type, time and environmental factors
- There are network of interactions among various biochemical entities in a cell (DNA, RNA, protein, small molecules)

Gene Regulation



Transcriptional Regulation: Example: The lac Operon



The *lac* Operon

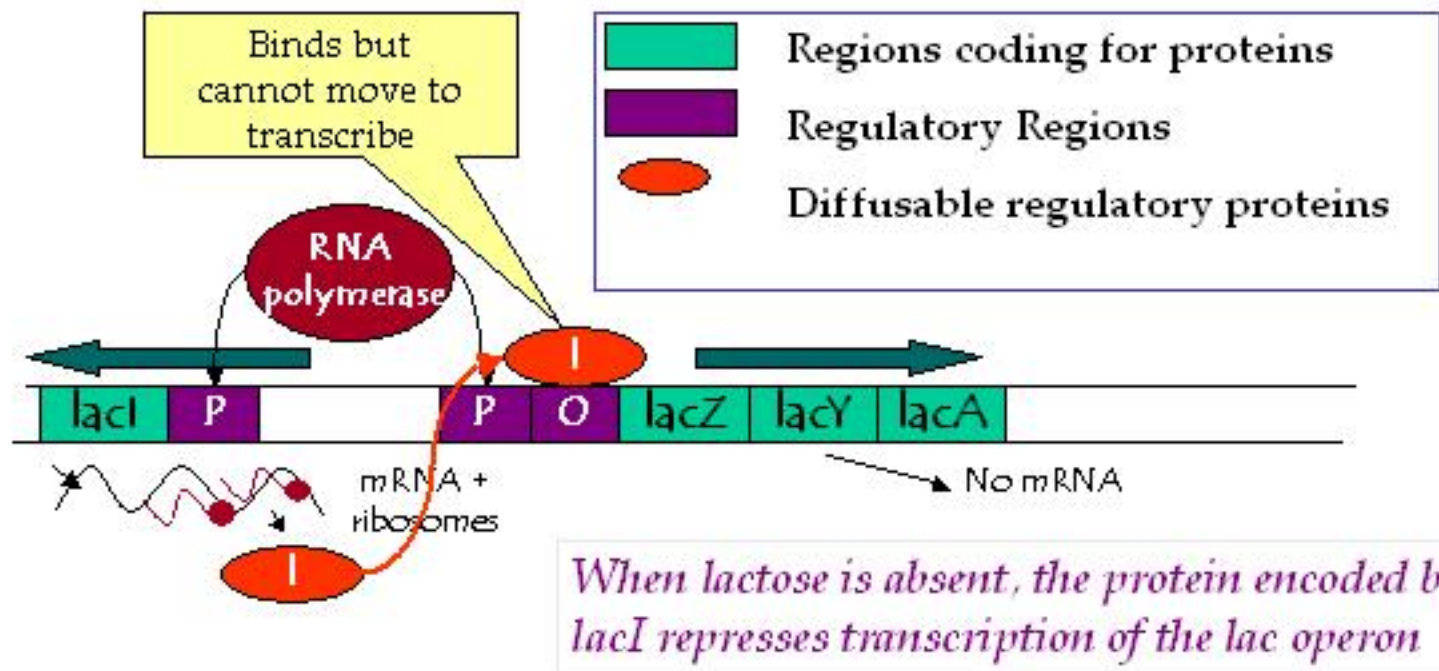
- Regulates utilization of lactose by the bacterium *E. coli*.
- Lactose is not generally available to *E. coli* as a food substrate, so the bacterium does not usually synthesize the enzymes necessary for its metabolic use.
- There is an operon, called the *lac* operon, normally turned off, that codes for three enzymes:
 - β -galactoside permease, β -galactosidase and β -thiogalactoside acetyl transferase.

Activation of the *lac* operon

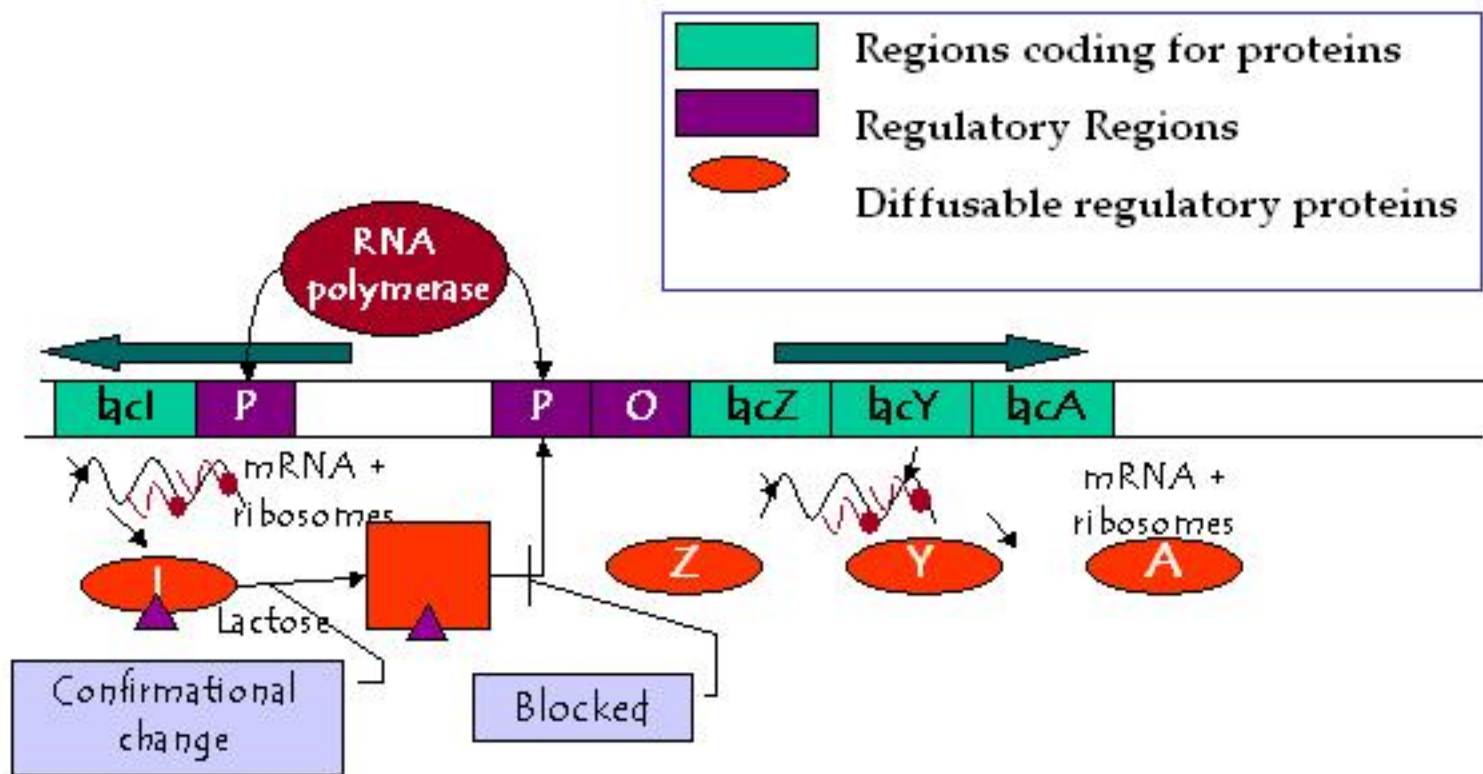
- If the bacterium is exposed to lactose, these enzymes work together to
 - transport lactose into the cell and
 - isomerizes lactose into allolactose (an allosteric isomer of lactose).
- The allolactose binds with a repressor molecule to keep it from repressing the production of mRNA.
- Production of allolactose turns on the production of mRNA, which then leads to production of more enzyme, enabling production of more lactose to allolactose...

An autocatalytic reaction..

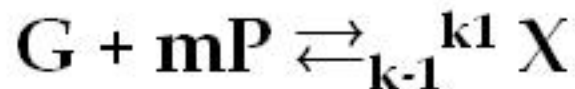
Transcriptional Regulation: Example: The lac Operon



Transcriptional Regulation: Example: The lac Operon



Mathematical Model



- Production of enzyme is turned on by m molecules of the product allolactose P ...
- G =Inactive state of the gene
- X =Active state of the gene
- In a large population of genes, the percentage of active genes is given by the chemical equilibrium:

$$p = [P]^m / (k_{eq}^m + [P]^m)$$

Production of mRNA

- The differential equation governing the (average) production of mRNA

$$\frac{dM}{dt} = M_0 + k_1 [P]^m / (k_{eq}^m + [P]^m) - k_2 M,$$

- where M is the concentration of mRNA that codes for the enzyme.
- Production of the enzymes (responsible for transforming into allolactose substrate):

$$\frac{dE_1}{dt} = c_1 M - d_1 E_1;$$

$$\frac{dE_2}{dt} = c_2 M - d_2 E_2.$$

Lactose states

- S_0 = Concentration of the lactose that is exterior to the cell.
- S = Concentration of the lactose that is interior to the cell.
- $[P]$ = Concentration of allolactose.

$$dS_0/dt = -\sigma_0 E_1 S_0/(k_0 + S_0)$$

$$dS/dt = \sigma_0 E_1 S_0/(k_0 + S_0) - \sigma_1 E_2 S/(k_s + S)$$

$$d[P]/dt = \sigma_1 E_2 S/(k_s + S) - \sigma_2 E_2 [P]/(k_p + [P])$$

Simplification

- Assume: mRNA is in quasi-steady state:

$$M = (k_1/k_2)[P]^m/(k_{eq}^m + [P]^m) + M_0/k_2;$$

- Assume: $d_1 = d_2$. Degradation is slow compared to cell growth. Also, $E_1 = E_2$.

$$dE_1/dt =$$

$$c_1 M_0/k_2 + (c_1 k_1/k_2)[P]^m/(k_{eq}^m + [P]^m) - d_1 E_1;$$

- Assume: No delay in conversion of the lactose into allolactose:

$$d[P]/dt = \sigma_0 E_1 S_0/(k_0 + S_0) - \sigma_2 E_1 [P]/(k_p + [P]).$$

Dimensionless Form

- Dimensionless variables: $S_0 = k_0 s$, $[P] = k_p p$, $E_1 = e_0 e$, and $t = t_0 \tau$...

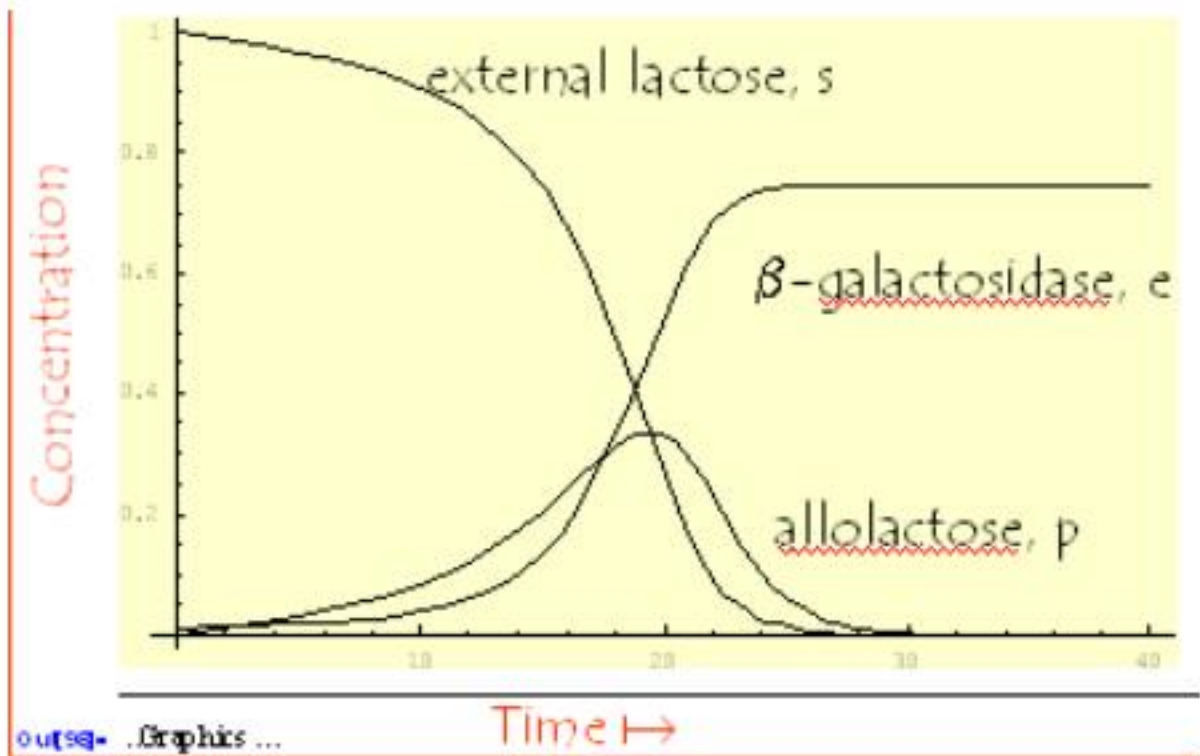
$$de/d\tau = m_0 + p^m/(\kappa^m + p^m) - \varepsilon e,$$

$$dp/d\tau = \mu e[s/(s+1) - \lambda p/(p+1)],$$

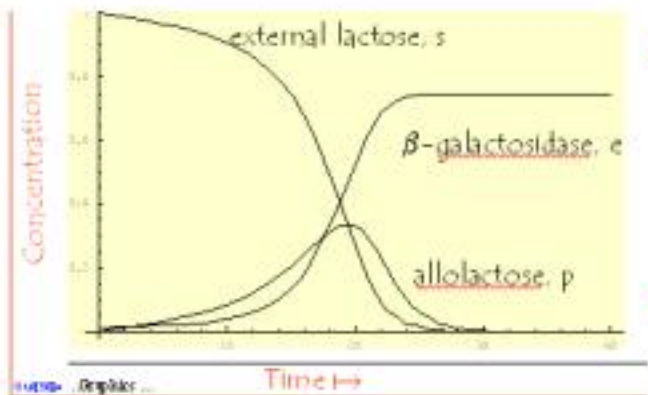
$$ds/d\tau = -e s/(s+1),$$

- where $e_0^2 = c_1 k_0 k_1 / (\sigma_0 k_2)$, $t_0 = k + 0 / (e_0 \sigma_0)$, $\lambda = \sigma_2 / \sigma_0$, $\mu = k_0 / k_p$, $\kappa = k / k_p$, $m_0 = M_0 / k_1$, and $\varepsilon = t_0 d_1$...

Time Evolution



The *lac* operon

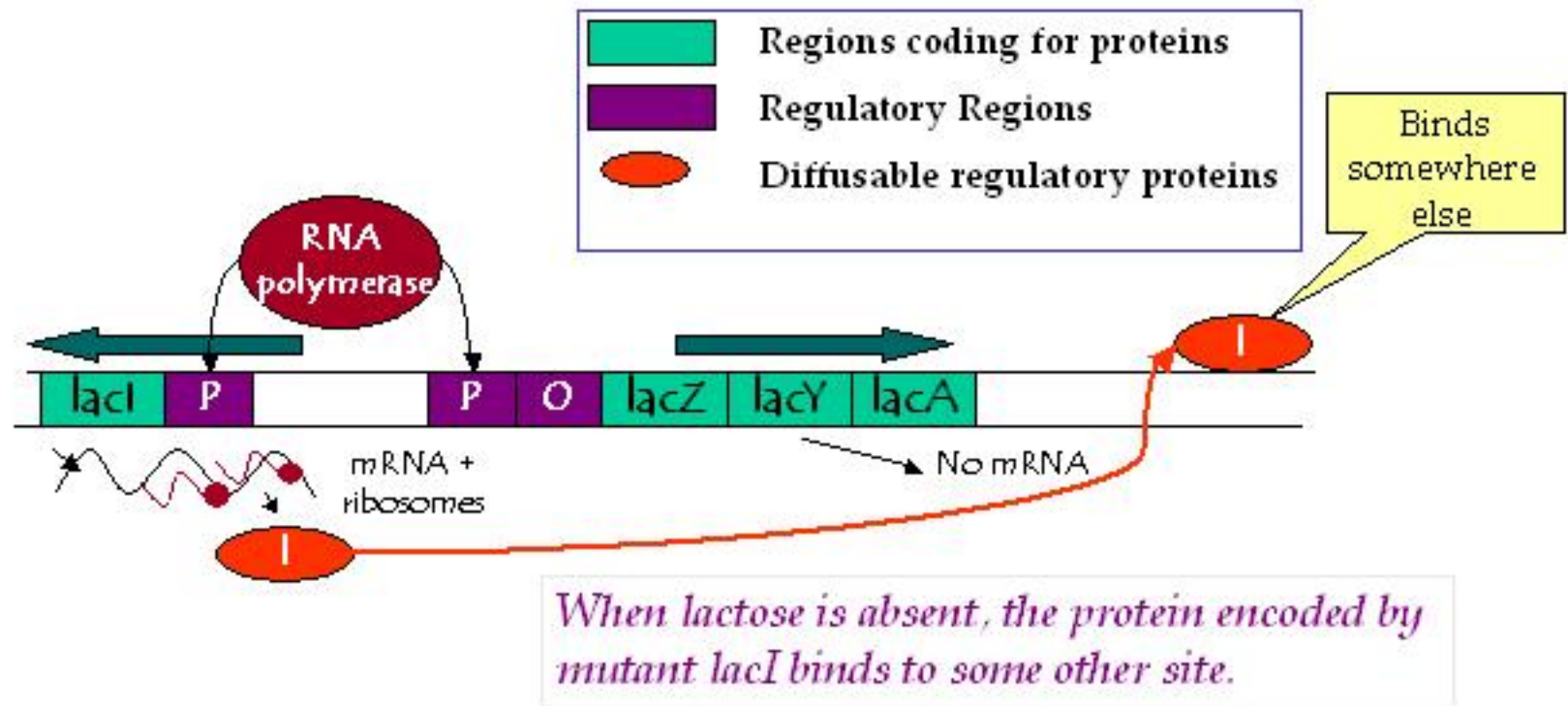


- If the amount of lactose is too small, then the lactose is gradually depleted, although there is no increase in enzyme concentration.
- However, if the lactose dose is sufficiently large, then there is an autocatalytic response, as the *lac* operon is turned on and enzyme is produced.
- The production of enzyme shuts down when the lactose stimulus is consumed, and the enzyme concentration gradually declines...

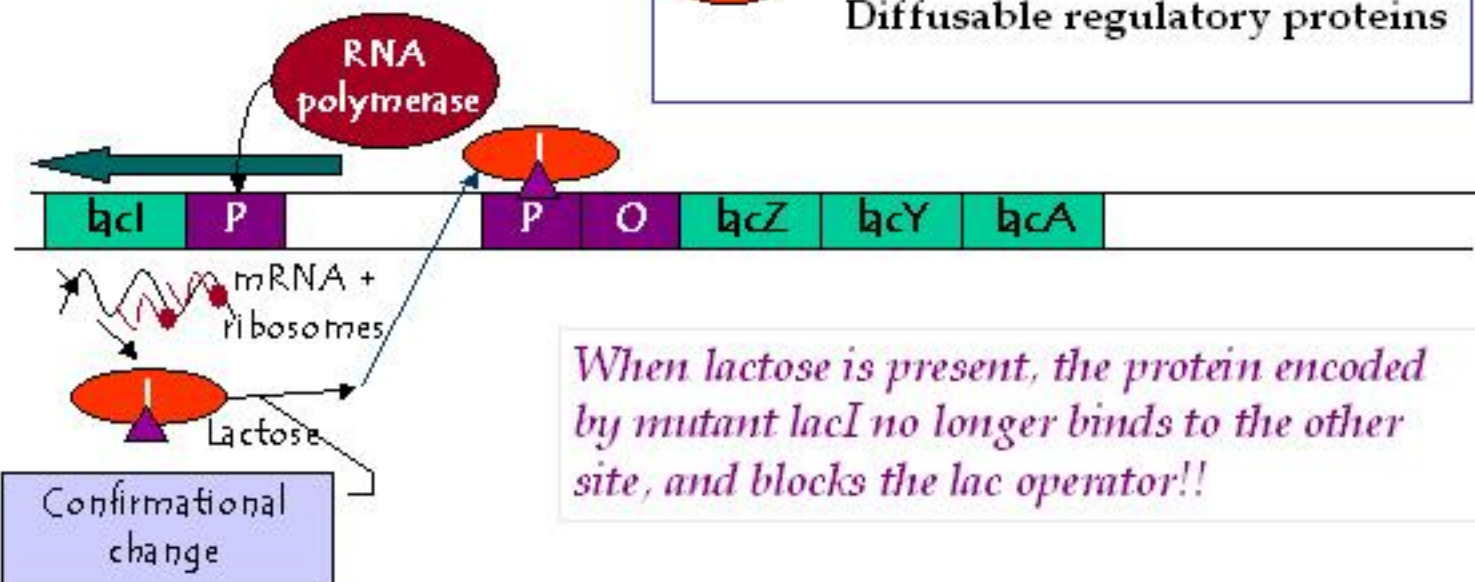
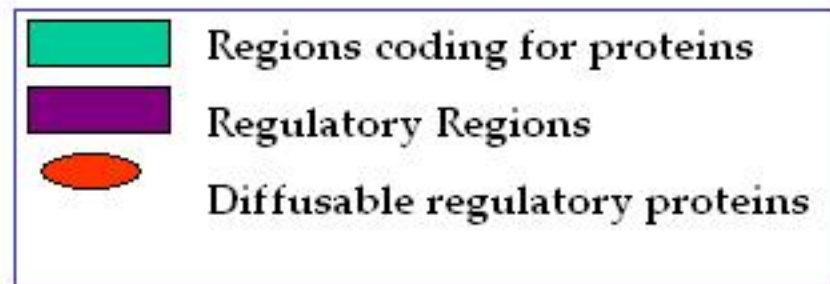
Example of Competition

- The mutant Lac repressor X186:
 - This mutant represses transcription of the *lac* genes in the presence of lactose...
 - The mutant binds DNA so tightly that, in the absence of inducer (allolactose), it is sequestered on non-operator DNA sites.
 - The inducer weakens the binding of the mutant repressor; thus, allowing it bind to the *lac* operon.

Lac repressor X 186



Lac repressor X 186



*When lactose is present, the protein encoded by mutant *lacI* no longer binds to the other site, and blocks the lac operator!!*

A General Scheme

Modular Description of Biochemical Pathways

Canonical Form:

$$\begin{cases} \dot{X}_i = \alpha_i \prod_{j=1}^{n+m} X_j^{g_{ij}} - \beta_i \prod_{j=1}^{n+m} X_j^{h_{ij}} & i=1 \dots n \\ C_l(X_1(t), \dots, X_{n+m}(t)) = \sum (\gamma_l \prod_{j=1}^{n+m} X_j^{f_{lj}}) = 0 \end{cases}$$

Characteristics:

- **Predefined Modular Structure**
- **Automated Translation from Graphical to Mathematical Model**

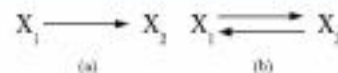


Figure 1: Representation of an unmodified and of a reversible reaction.

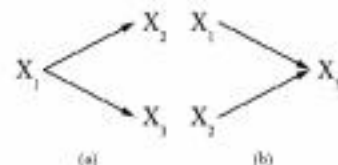


Figure 2: Representation of a divergence and of a convergence branch point (the two processes in each reaction are independent of each other).

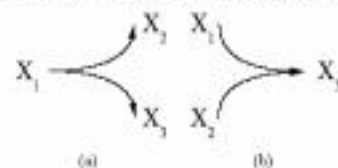


Figure 3: Representation of a single splitting reaction generating two products, X_2 and X_3 , in stoichiometric proportions and of a single synthetic reaction involving two source components, X_1 and X_2 always in stoichiometric proportions.

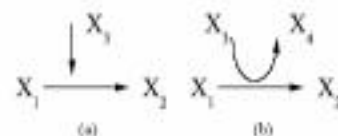
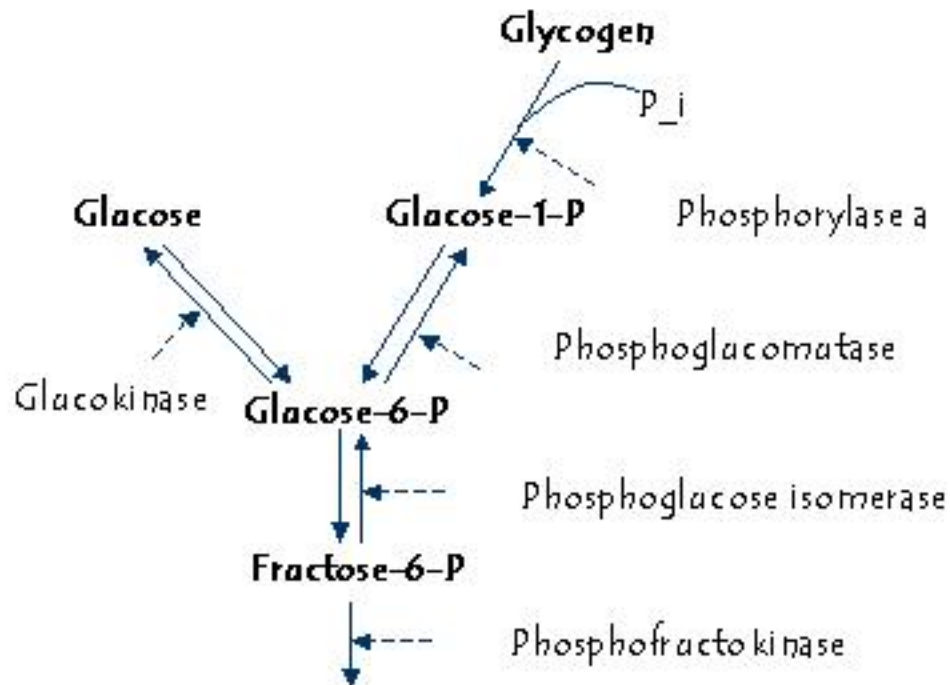


Figure 4: The conversion of X_1 into X_2 is modulated (stimulation or inhibition is represented by the sign of the arrow) by X_3 . The reaction between X_1 and X_2 requires coenzyme X_4 , which in the process is converted into X_2 .

Example



Glycolysis

SIMPATHICA

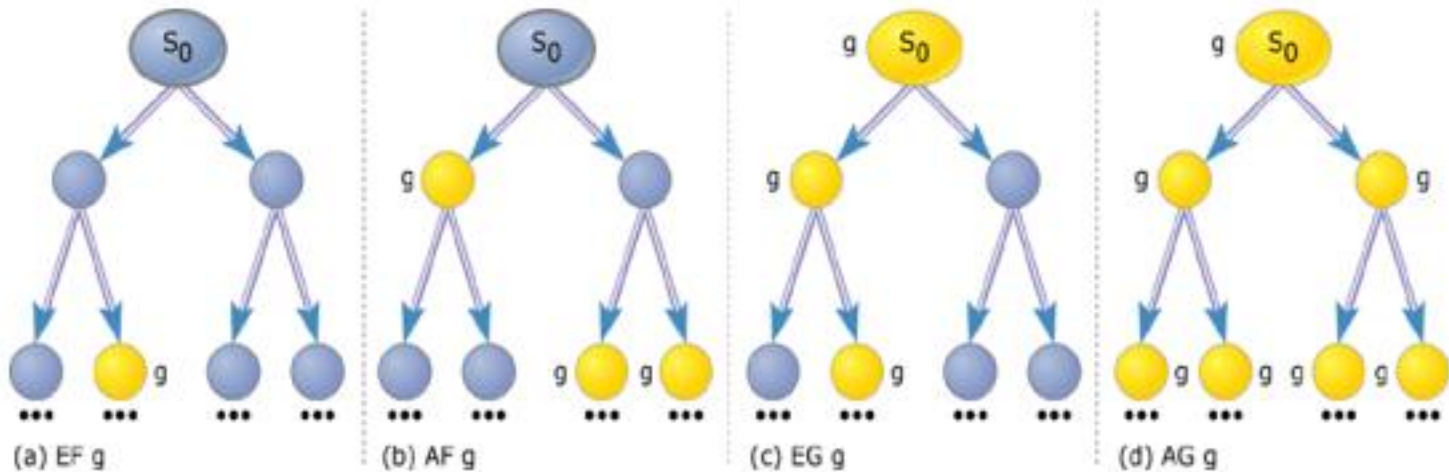
Discrete Models

- By quantizing time and state space, the models can be converted into a Kripke Structure... labeled Finite State Model...
- The properties of these systems can be scrutinized by queries structured in a Temporal Logic.

Semantics for CTL

- For $p \in AP$:
 $s \models p \Leftrightarrow p \in L(s)$ $s \models \neg p \Leftrightarrow p \notin L(s)$
- $s \models f \wedge g \Leftrightarrow s \models f$ and $s \models g$
- $s \models f \vee g \Leftrightarrow s \models f$ or $s \models g$
- $s \models EX f \Leftrightarrow \exists \pi = \langle s_0 s_1 \dots \rangle$ from s $s_1 \models f$
- $s \models E(f U g) \Leftrightarrow \exists \pi = \langle s_0 s_1 \dots \rangle$ from s
 $\exists j \geq 0 [s_j \models g \text{ and } \forall i : 0 \leq i < j [s_i \models f]]$
- $s \models EG f \Leftrightarrow \exists \pi = \langle s_0 s_1 \dots \rangle$ from s $\forall i \geq 0: s_i \models f$

Some CTL Operators



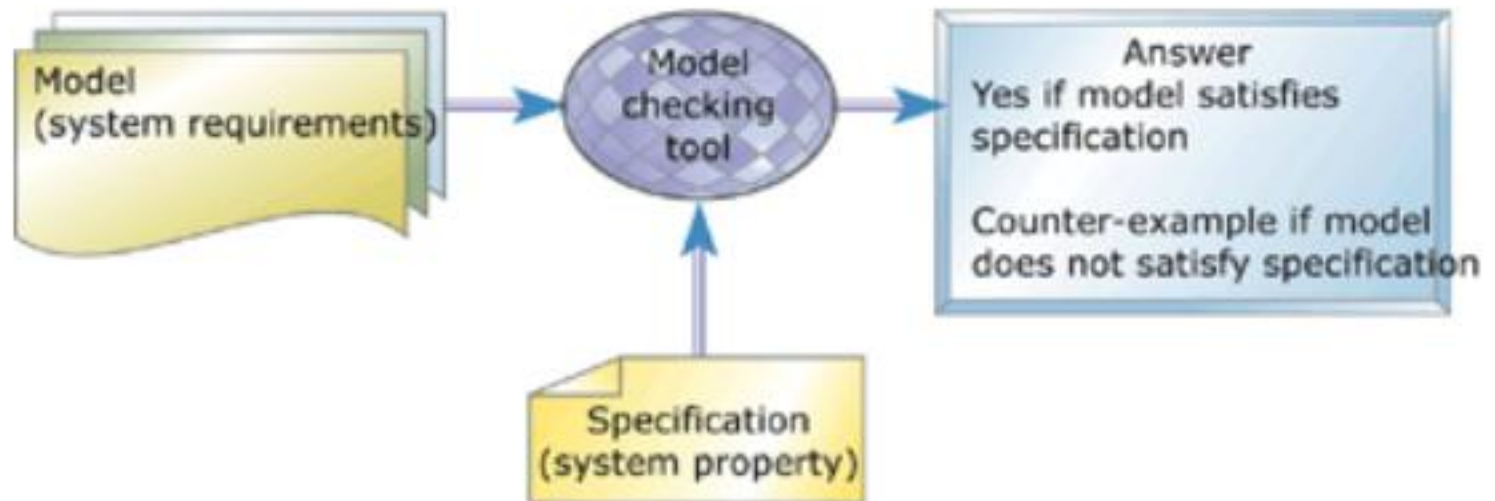
EF g

AF g

EG g

AG g

Model Checking in Engineering



Next Step

- Explore possible confluence of the theory of **hybrid automata** and the techniques of **algorithmic algebra** and **model checking** to create a computational basis for **systems biology**.
- **Simplest Scenario:**
- Devise a method to compute bounded reachability by combining Taylor polynomials and cylindric algebraic decomposition algorithms.
- What are the power and limitations of this framework .

Algorithmic Algebraic Model Checking

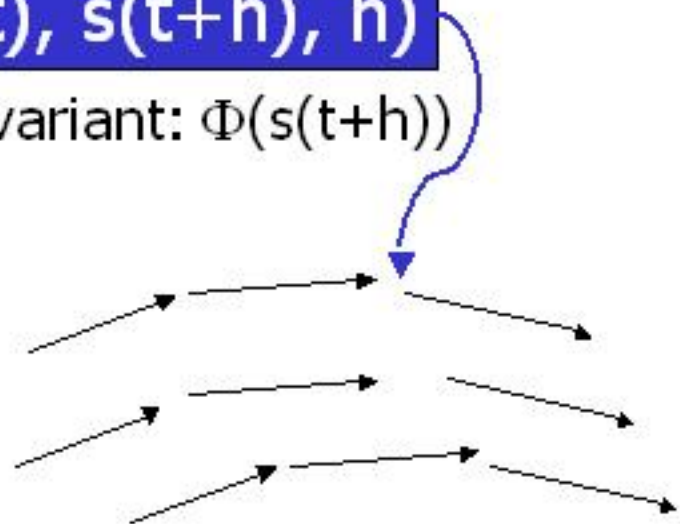
- Replacing numerical integration by a symbolic step:
- Generalizing Euler forward Numerical integration:
$$f(X,t+h) \sim f(X,t) + c_1.f'(X,t) h + \dots + c_k.f''(X,t) h^k$$
- Expression in “X”, “t” and “h”
- Error: integration discretization approximation
- Model Checking = iterative process of checking what is true now and at “next” time
- Possible over “semi-algebraic sets” using “quantifier elimination”

Symbolic Analysis

Invariant: $\Phi(s(t))$

$f(s(t), s(t+h), h)$

Invariant: $\Phi(s(t+h))$



$\Phi(s) =$

$\dots \vee [\forall(s(t)) \wedge f(s(t), s(t+h), h) \Rightarrow X(s(t+h))]$

Symbolic Model Checking

- Take the following question: Is a semi-algebraic formula Φ an invariant of the system?
- Given Φ is true at t , is it true at $t+h$?

$$\forall_t \Phi(s(t)) \Rightarrow \Phi(s(t+h))?$$

**The above statement can be expressed as
a Tarski sentence...**

Theorem

- *Let Ψ be a Tarski sentence. There is an effective decision procedure for Ψ .*

Let Ψ be a Tarski formula. There is a quantifier-free formula Φ logically equivalent to Ψ .

- *If Ψ involves only polynomials with rational coefficients, then so does the sentence Φ . \square*

Why approximate?

Can we get the complete and exact picture.. even when the system parameters remain unknown?

Differential Algebraic Approaches

- **Ritt-Kolchin:** Ideal Theoretic approach
- **Kolchin-Singer:** Galois Theoretic Approach
- **Lie:** Group Theoretic Approach
 - Understand their interrelationship
 - Determine how effective these approaches are...

Differential Algebra

Assume that the system (SISO) is described as shown below:

$$\begin{aligned}\dot{x}_1 &= p_1(X, u, \dot{u}, \dots, u^{(k)}) \\ &\vdots \\ \dot{x}_r &= p_r(X, u, \dot{u}, \dots, u^{(k)}) \\ 0 &= q_1(X, u) \\ &\vdots \\ 0 &= q_s(X, u) \\ y &= h(X, u)\end{aligned}$$

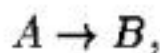
Consider the following differential ideal I in the differential ring $\mathbb{R}\{X, u, y\}$:

$$I = [\dot{x}_1 - p_1, \dots, \dot{x}_r - p_r, q_1, \dots, q_s, y - h].$$

The input-output relation is then obtained by finding the contraction I^c of the ideal I to the ring $\mathbb{R}\{u, y\}$. The generators of $I^c = I \cap \mathbb{R}\{u, y\}$ give the differential equation, the underlying algorithmic questions for which are largely unsolved.

Example System

Example Consider the following system (adapted from Forsman [Forsman92]):



with the following kinetic equations:

$$[\dot{B}] = [A]^{0.5} - [B]^{0.5}.$$

The input u controls the concentration $[A]$ as follows:

$$[\dot{A}] = u[A]^{-2} - [A]^{-1.5},$$

and the output y is simply $[B]$:

$$y = [B].$$

We can simplify the above system to a polynomial system by following transformations:

$$x_1^2 = [A] \quad \text{and} \quad x_2^2 = [B].$$

Input-Output Relations

Thus,

$$I = [2x_1^5\dot{x}_1 + x_1 - u, 2x_2\dot{x}_2 + x_2 - x_1, x_2^2 - y].$$

After eliminating x_1 and x_2 , we obtain the following input-output relation:

$$\begin{aligned} & (20\dot{y}^8y^2 - 4\dot{y}^{10}y - 40\dot{y}^6y^3 + 40\dot{y}^4y^4 - 20\dot{y}^2y^5 + 4y^6)\ddot{y}^2 \\ & + (4u\dot{y}^5y - 4\dot{y}^6y - 20\dot{y}^4y^2 + 40u\dot{y}^3y^2 + 20\dot{y}^2y^3 + 20u\dot{y}y^3 + 4y^4)\ddot{y} \\ & - \dot{y}^2y^5 + 5\dot{y}^4y^4 - 10\dot{y}^6y^3 + 20u\dot{y}^3y^2 + 10\dot{y}^8y^2 + y^2 - 8\dot{y}^6y + 10u\dot{y}^5y \\ & - u^2y + 2u\dot{y}y - \dot{y}^2y - 5\dot{y}^{10}y + \dot{y}^{12} + 8\dot{y}^2y^3 + 2u\dot{y}y^3 = 0. \square \end{aligned}$$

Obstacles

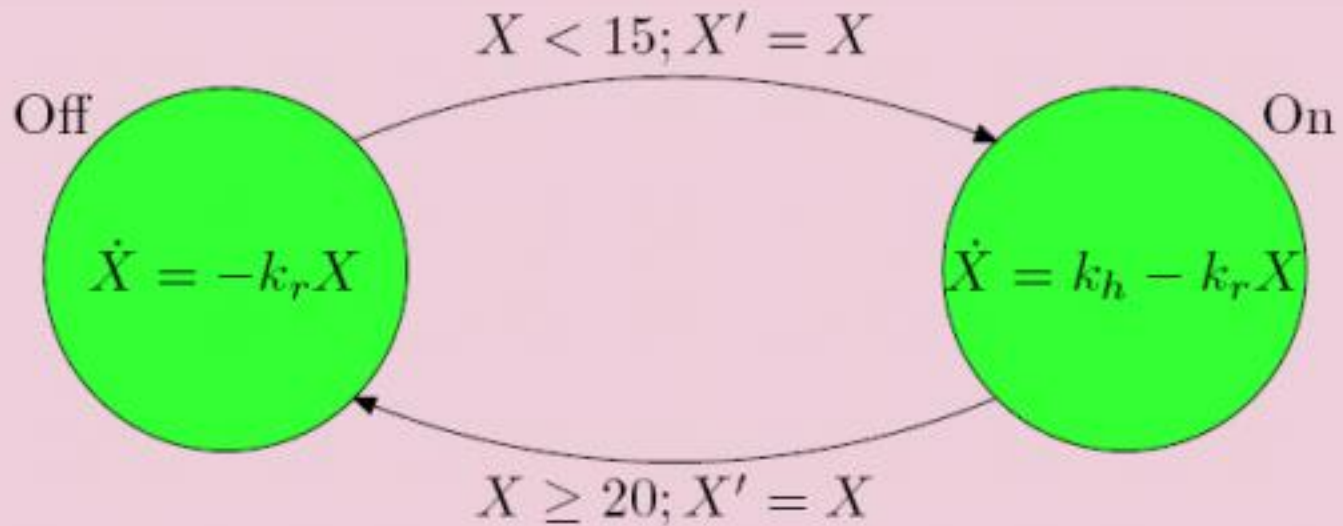
- Various Approaches:
- Ideas based on Groebner Bases
- Ideas based on Ritt's Characteristic Set]
- Obstacles:
 - Failure of a Hilbert Basis like theorem (only a weaker version, Ritt-Radenbush theorem holds), existence of non-recursive differential ideals, etc.

Hybrid Automaton

- A hybrid automaton (of dimension k) $H = \langle Z, Z', V, E, \text{Inv}, \text{Dyn}, \text{Act}, \text{Reset} \rangle$ (over M), consists of the following components:
 1. $Z = (Z_1, \dots, Z_k)$ and $Z' = (Z'_1, \dots, Z'_k)$ are two vectors of variables ranging over the reals, \mathbb{R} ;
 2. $\langle V, E \rangle$ is a finite directed graph; the vertices of V are called locations, or control modes, the directed edges in E , control switches;
 3. Each $v \in V$ is labeled by the two formulæ $\text{Inv}(v)[Z]$ and $\text{Dyn}(v)[Z, Z', T]$ such that if $\text{Inv}(v)[p]$ holds (in M), then $\text{Dyn}(v)[p, p, 0]$ holds as well;
 4. Each $e \in E$ is labeled by the formulæ $\text{Act}(e)[Z]$ and $\text{Reset}(e)[Z, Z']$.

Thermostat

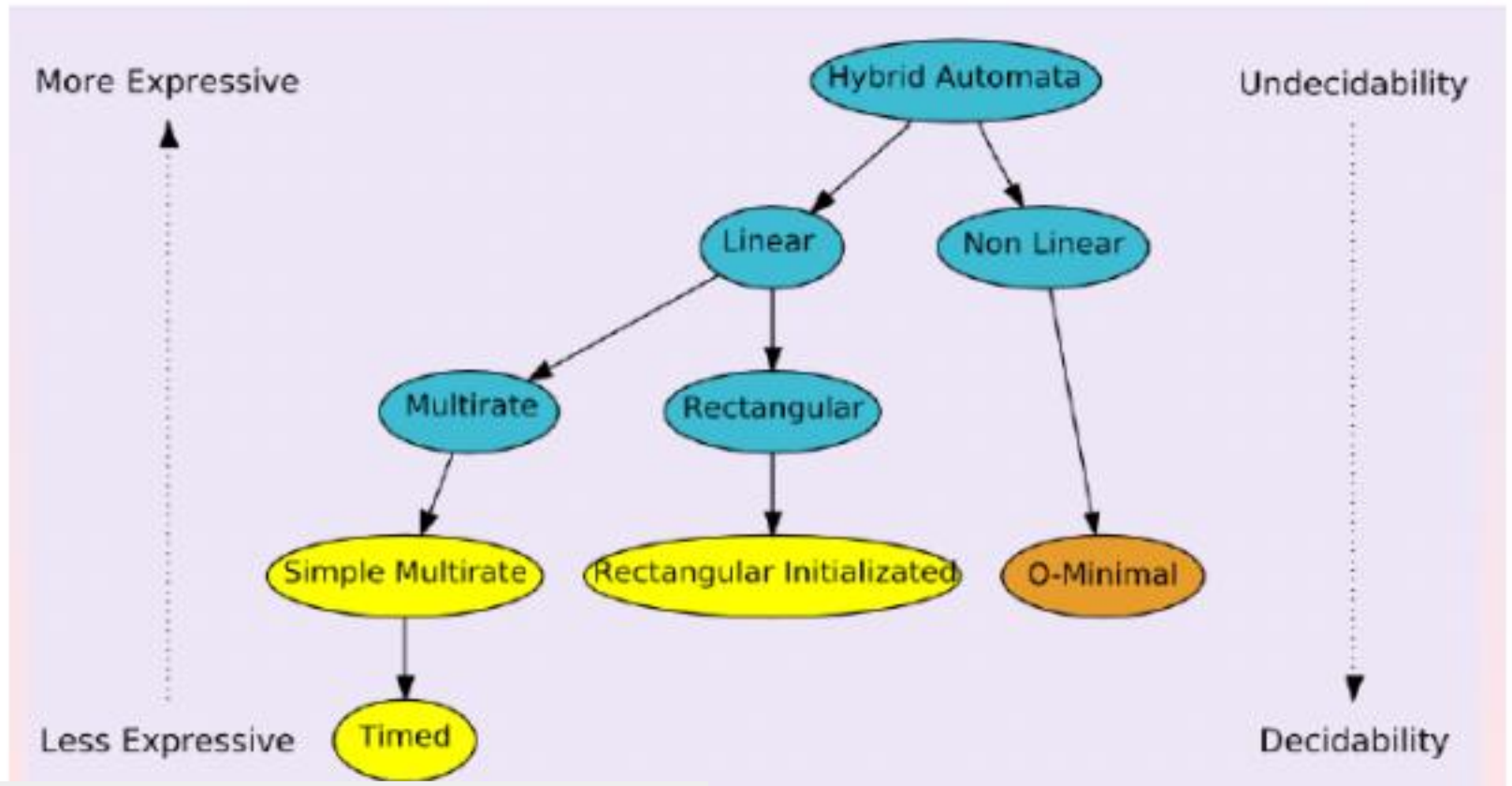
A thermostat model



Subclasses Of Hybrid Systems

- **Timed Automaton** - a discrete transition system where the only continuous variable allowed is the clock
- **Multirate Automaton** - a discrete transition system where there can be many continuous variables with a constant flow
- **Rectangular Automaton** is a discrete transition system where the flows are allowed to vary within a range
- **Linear Systems** - The reachability problem for subclasses of linear hybrid systems have been proved
- **Time-Invariant Systems:** flow equations, guards and state invariants are all independent of time
- **O-Minimal Systems** - restricted jump condition: the new continuous state cannot depend on the old state, and the system is assumed to be time-invariant

Decision problems



SaCoRe

- Hybrid Automata's inclusion dynamics, approximated by semi-algebraic formula.
 $\text{Dyn}[X, X', T] \equiv \text{Semialgebraic Set}$
- A more realistic approximation, for time invariant systems:
 - $\text{Dyn}[X, X', h]$
 $\approx \{X' \mid X' = X + \mathcal{F}(X, 0) h + \delta, |\delta| < \varepsilon\},$
for a suitably chosen
$$\varepsilon \equiv |\mathcal{F}(X, 0) h^2/2! + \mathcal{F}(X, 0) h^3/3! + \dots|$$

Another Example: Biological Pattern Formation

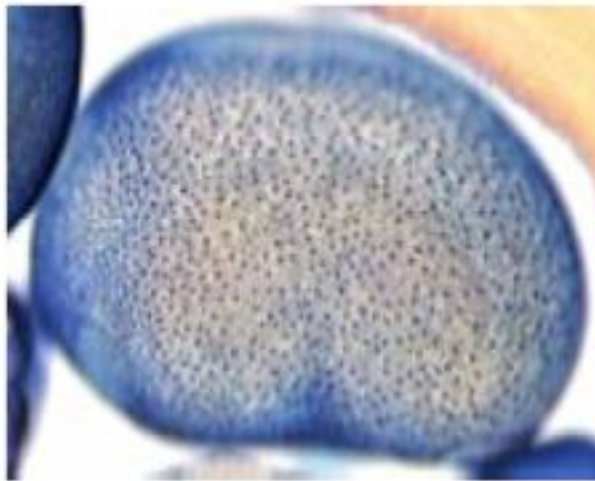
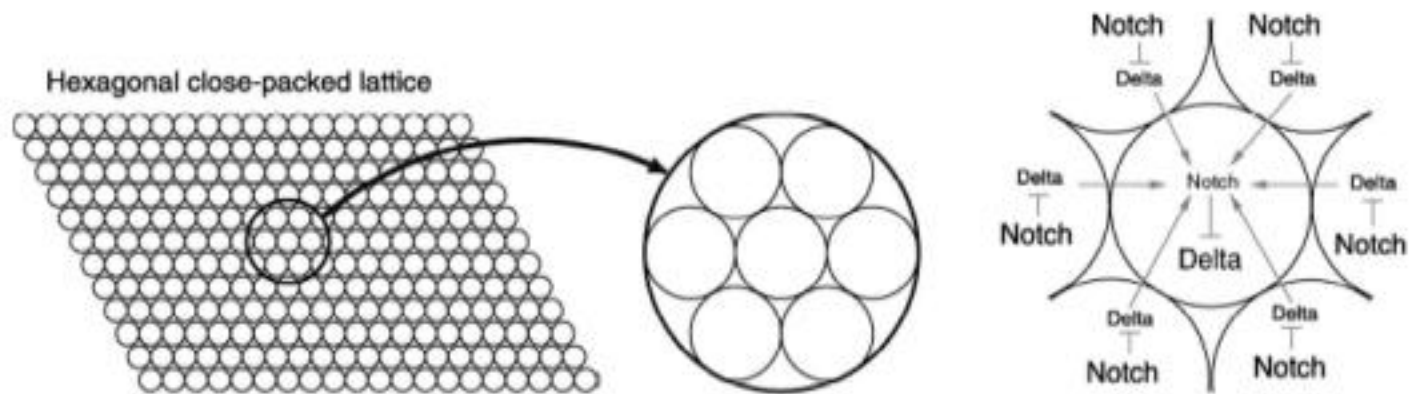


Figure 3: *Xenopus* embryo labeled by a marker for ciliated cell precursors seen as black dots.¹

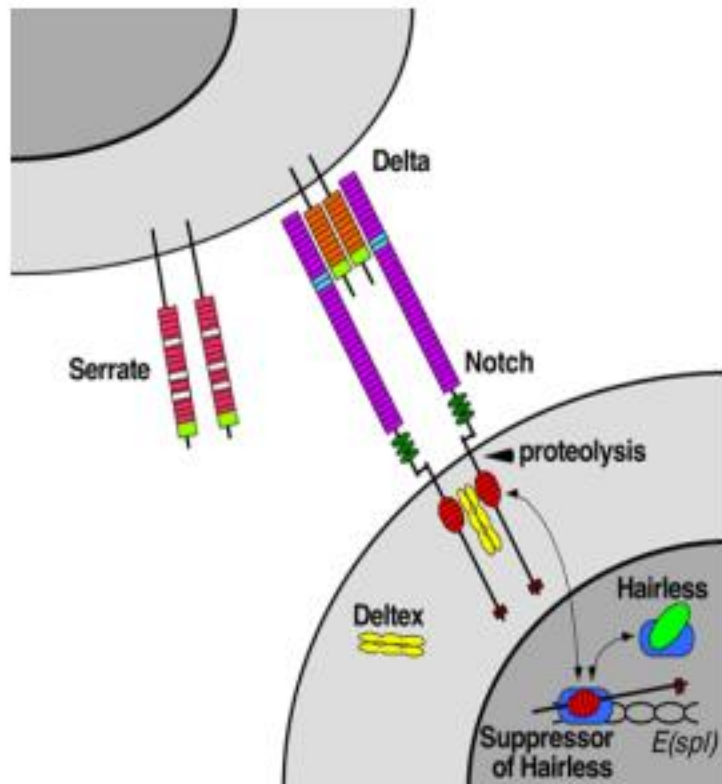
- Embryonic Skin Of The South African Claw-Toed Frog
- “Salt-and-Pepper” pattern formed due to lateral inhibition in the *Xenopus* epidermal layer

Delta-Notch Signaling



Physically **adjacent** cells **laterally inhibit** each other's ciliation (Delta production)

Delta-Notch Pathway



- Delta binds and activates its receptor Notch in neighboring cells
- Activated Notch suppresses ligand (Delta) production in the cell
- A cell producing more ligands forces its neighboring cells to produce less

Pattern formation by lateral inhibition with feedback: a mathematical model of Delta-Notch intercellular signaling ... Collier et al.(1996)

$$\frac{d(N_P/N_0)}{d\tau} = F(\bar{D}_P/D_0) - \mu N_P/N_0,$$

$$\frac{d(D_P/D_0)}{d\tau} = G(N_P/N_0) - \rho D_P/D_0.$$

Rewriting...

$$\dot{n}_P = f(\bar{d}_P) - n_P,$$

$$\dot{d}_P = v\{g(n_P) - d_P\}.$$

Where:

$$f(x) = \frac{x^k}{a + x^k}, \quad g(x) = \frac{1}{1 + bx^h},$$

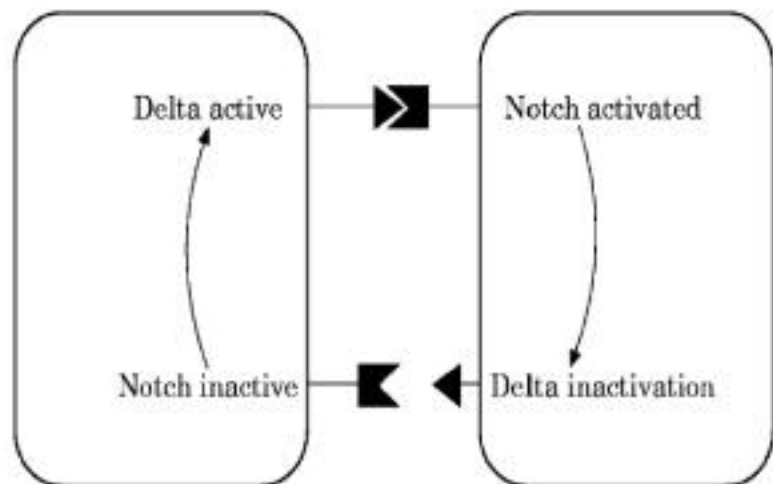
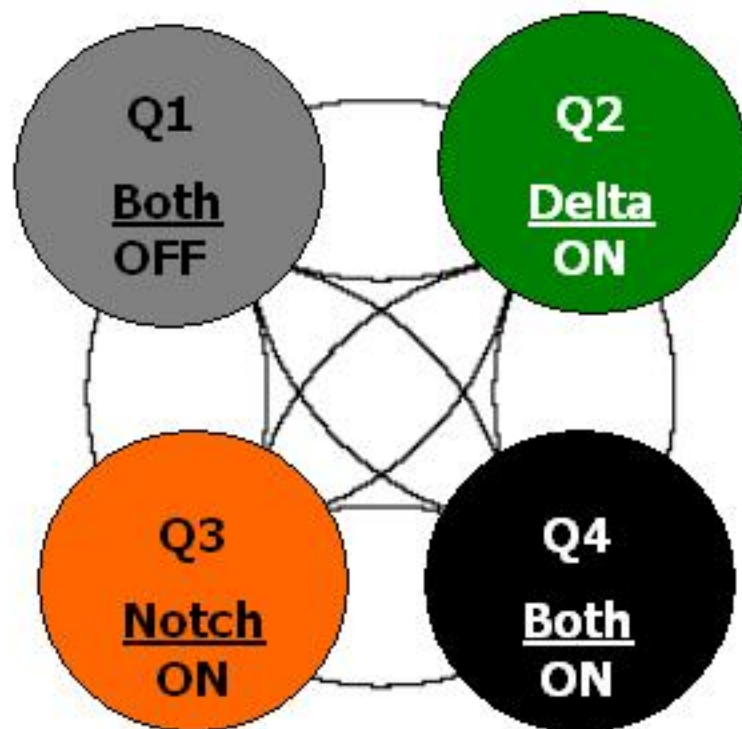


FIG. 1. Diagrammatic representation of the effective feedback loop between Notch and Delta in neighbouring cells. Details of the Notch signalling pathway are omitted for clarity. Key: \blacktriangleright Delta; \blacktriangleleft Notch.

Collier et al.

One-Cell Delta-Notch Hybrid Automaton

Ghosh et al.



(a) Transition diagram for a single cell

$$\mathcal{U}_{one_cell} = (Q, X, \Sigma, Init, f, Inv, R)$$

$$Q = q_1, q_2, q_3, q_4$$

$$X = (x_1, x_2)^T \in \mathbb{R}^2$$

$$\Sigma = \left\{ u_N = \sum_{i=1}^6 x_{Delta,i} \right\}$$

$$Init = Q \times \{X \in \mathbb{R}^2 : x_1, x_2 > 0\}$$

$$f(q, x) = \begin{cases} [-\lambda_D x_1; -\lambda_N x_2]^T & \text{if } q = q_1 \\ [R_D - \lambda_D x_1; -\lambda_N x_2]^T & \text{if } q = q_2 \\ [-\lambda_D x_1; R_N - \lambda_N x_2]^T & \text{if } q = q_3 \\ [R_D - \lambda_D x_1; R_N - \lambda_N x_2]^T & \text{if } q = q_4 \end{cases}$$

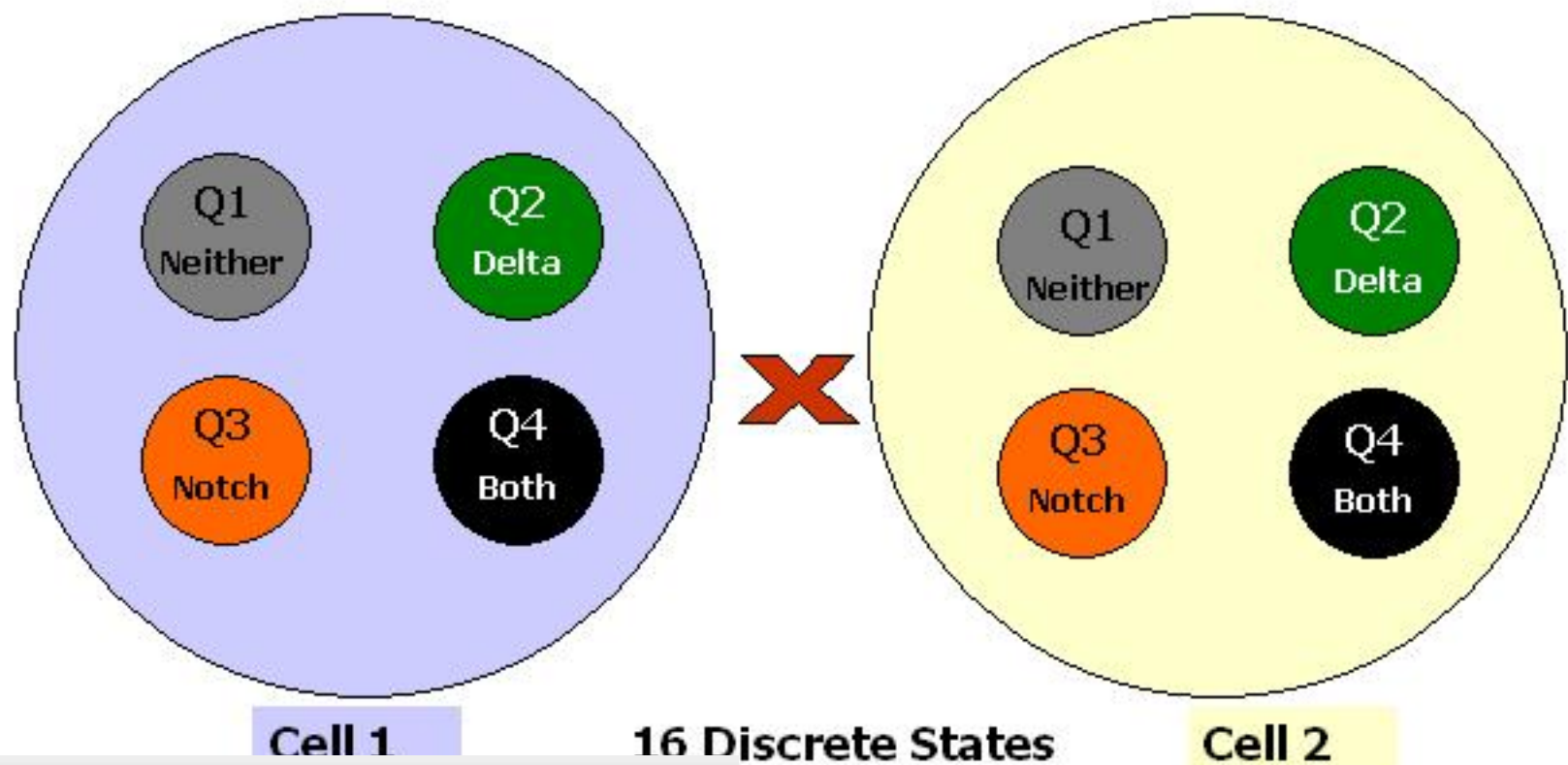
$$Inv = \{q_1, \{-x_2 < h_D, u_N < h_N\}\} \cup$$

$$\{q_2, \{-x_2 \geq h_D, u_N < h_N\}\} \cup$$

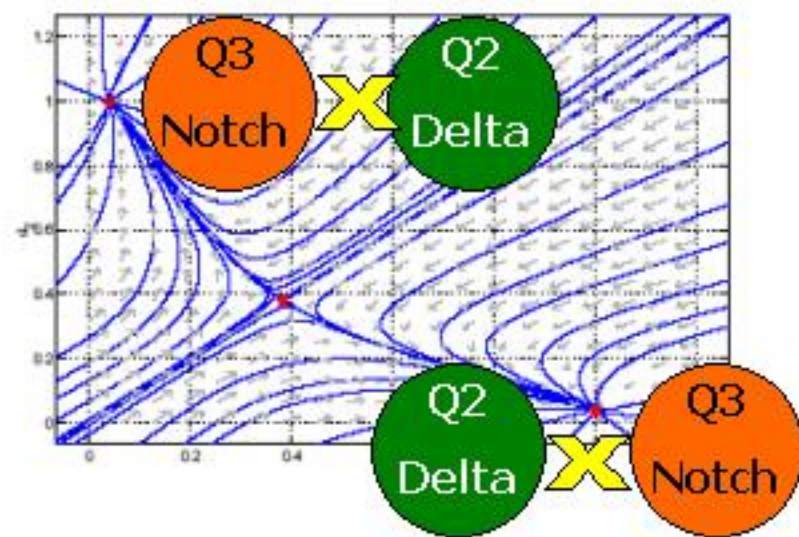
$$\{q_3, \{-x_2 < h_D, u_N \geq h_N\}\} \cup$$

$$\{q_4, \{-x_2 \geq h_D, u_N \geq h_N\}\}$$

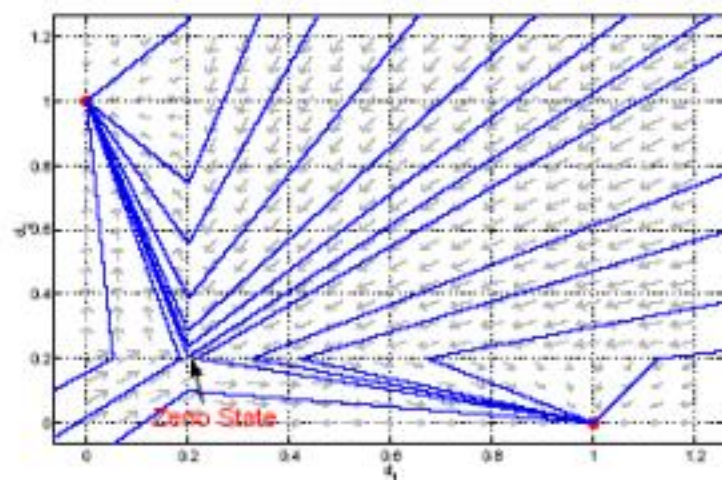
Two-Cell Delta-Notch System



System Properties: True & Approximate



(a) Nonlinear model



(b) Hybrid systems model

Fig. 7. Phase plane projections for two cell system showing equilibria. Labels d_1 and d_2 are the Delta protein concentrations in cell 1 and 2 respectively.

State Reachability



Reaching State q_7 (2,3) When we ask $True \exists \mathcal{U} [-2n_1 > -1 \wedge 5d_2 < 1 \wedge -2n_2 < -1 \wedge 5d_1 > 1]$, we get:

Iteration 1: $5d_1 - 1 \geq 0 \wedge 2n_1 - 1 \leq 0 \wedge 5d_2 - 1 \leq 0 \wedge 2n_2 - 1 \geq 0$

Iteration 2: $n_1 - 1 \leq 0 \wedge [[2n_1 - 5d_1 \leq 0 \wedge 5d_2 - 1 \leq 0 \wedge 8n_2 - 5d_2 - 3 \geq 0 \wedge n_2 + n_1 - 1 = 0] \vee [8n_1 - 5d_1 - 3 \leq 0 \wedge 4d_2 + d_1 - 1 = 0 \wedge 2n_2 - 1 \geq 0 \wedge 8n_2 + 5d_1 - 5 \geq 0] \vee [5d_1 - 1 \geq 0 \wedge 2n_1 - 5d_1 \leq 0 \wedge 5d_2 + 2n_1 - 2 \leq 0 \wedge 2n_2 - 1 \geq 0] \vee [5d_1 - 1 \geq 0 \wedge 2n_1 - 1 \leq 0 \wedge 5d_2 - 1 \leq 0 \wedge 8n_2 - 5d_2 - 3 \geq 0] \vee [2n_1 - 1 \leq 0 \wedge 5d_2 - 1 \leq 0 \wedge 8n_2 - 5d_2 - 3 \geq 0 \wedge 8n_2 + 5d_1 - 5 \geq 0] \vee [2n_1 - 5d_1 \leq 0 \wedge 5d_2 - 1 \leq 0 \wedge 2n_2 - 1 \geq 0 \wedge 8n_2 + 5d_1 - 5 \geq 0]]$
 $\equiv f_7$ (say).

State Reachability



Reaching State q_{10} (3,2) When we ask $True \exists \mathcal{U} [-2n_1 < -1 \wedge 5d_2 > 1 \wedge -2n_2 > -1 \wedge 5d_1 < 1]$, we get:

Iteration 1: $5d_1 - 1 \leq 0 \wedge 2n_1 - 1 \geq 0 \wedge 5d_2 - 1 \geq 0 \wedge 2n_2 - 1 \leq 0$

Iteration 2: $n_2 - 1 \leq 0 \wedge [(2n_1 - 1 \geq 0 \wedge 5d_2 + 8n_1 - 5 \geq 0 \wedge d_2 + 4d_1 - 1 = 0 \wedge 2n_2 + 5d_1 - 2 \leq 0) \vee [2n_1 - 1 < 0 \wedge 8n_1 - 5d_1 - 3 \geq 0 \wedge 5d_2 + 8n_1 - 5 \geq 0 \wedge n_2 + n_1 - 1 = 0] \vee [8n_1 - 5d_1 - 3 \geq 0 \wedge 5d_2 + 8n_1 - 5 < 0 \wedge 5d_2 + 2n_1 - 2 \geq 0 \wedge n_2 + n_1 - 1 = 0] \vee [2n_1 - 1 \geq 0 \wedge 5d_2 - 1 \geq 0 \wedge 2n_2 + 5d_1 - 2 \leq 0 \wedge n_2 + n_1 - 1 < 0] \vee [5d_1 - 1 \leq 0 \wedge 2n_1 - 1 \geq 0 \wedge 5d_2 + 8n_1 - 5 \geq 0 \wedge 2n_2 - 5d_2 \leq 0] \vee [5d_1 - 1 \leq 0 \wedge 2n_1 - 1 \geq 0 \wedge 5d_2 + 8n_1 - 5 \geq 0 \wedge 2n_2 - 1 \leq 0] \vee [8n_1 - 5d_1 - 3 \geq 0 \wedge 5d_2 - 1 \geq 0 \wedge 2n_2 + 5d_1 - 2 \leq 0 \wedge 2n_2 - 1 \leq 0]]$

Impossibility Of Reaching Wrong Equilibrium:



$$f_7 \wedge \neg f_{10} = n_1 - 1 \leq 0 \wedge [(2n_1 - 5d_1 \leq 0 \wedge 5d_2 - 1 < 0 \wedge 8n_2 - 5d_2 - 3 \geq 0 \wedge n_2 + n_1 - 1 = 0) \vee [2n_1 - 1 \leq 0 \wedge 5d_2 - 1 \leq 0 \wedge 8n_2 - 5d_2 - 3 \geq 0 \wedge 2n_2 + 5d_1 - 2 > 0] \vee [2n_1 - 1 \leq 0 \wedge 5d_2 + 2n_1 - 2 \leq 0 \wedge 4d_2 + d_1 - 1 = 0 \wedge n_2 + n_1 - 1 > 0] \vee [2n_1 - 5d_1 \leq 0 \wedge 5d_2 - 1 \leq 0 \wedge n_2 + n_1 - 1 > 0 \wedge 2n_2 - 1 \geq 0] \vee [2n_1 - 1 \leq 0 \wedge 5d_2 - 1 < 0 \wedge 8n_2 - 5d_2 - 3 \geq 0 \wedge 8n_2 + 5d_1 - 5 \geq 0] \vee [8n_1 - 5d_1 - 3 < 0 \wedge 4d_2 + d_1 - 1 = 0 \wedge 2n_2 - 1 \geq 0 \wedge 8n_2 + 5d_1 - 5 \geq 0] \vee [5d_1 - 1 \geq 0 \wedge 2n_1 - 5d_1 < 0 \wedge 5d_2 + 2n_1 - 2 \leq 0 \wedge 2n_2 - 1 \geq 0] \vee [5d_1 - 1 \geq 0 \wedge 2n_1 - 1 \leq 0 \wedge 5d_2 - 1 < 0 \wedge 8n_2 - 5d_2 - 3 \geq 0] \vee [2n_1 - 1 < 0 \wedge 5d_2 - 1 \leq 0 \wedge 8n_2 - 5d_2 - 3 \geq 0 \wedge 8n_2 + 5d_1 - 5 \geq 0]]$$

Since we have assumed no upper bound on the initial values and since we have been able to compute only two iterations, this formula does *not* evaluate to *True* given $n_1 < n_2 \wedge d_1 > d_2$. However, when *Qepcad* simplifies the above formula assuming that $n_1 > n_2 \wedge d_1 < d_2$, it immediately evaluates

Semi-Algebraic Hybrid Automata.

- **Definition.** A k -dimensional hybrid automaton is a 7-tuple, $H = (Z, V, E, \text{Init}, \text{Inv}, \text{Flow}, \text{Jump})$, consisting of the following components:
 - $Z = \{Z_1, \dots, Z_k\}$ a finite set of variables ranging over the reals \mathbb{R} ;
 - $\dot{Z} = \{\dot{Z}_1, \dots, \dot{Z}_k\}$ denotes the first derivatives with respect to the time $t \in \mathbb{R}$ during continuous change;
 - $Z' = \{Z'_1, \dots, Z'_k\}$ denotes the set of values at the end of a discrete change;
- (V, E) is a directed graph; the vertices of V are called control modes, the edges of E are called control switches;

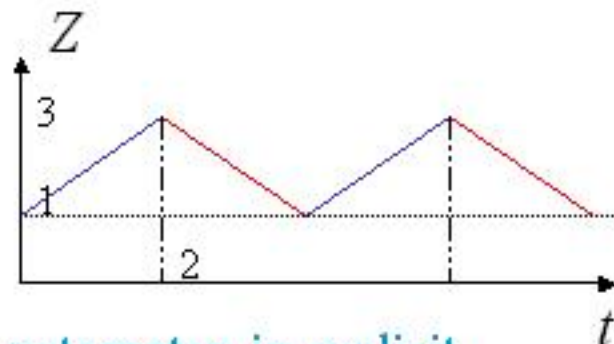
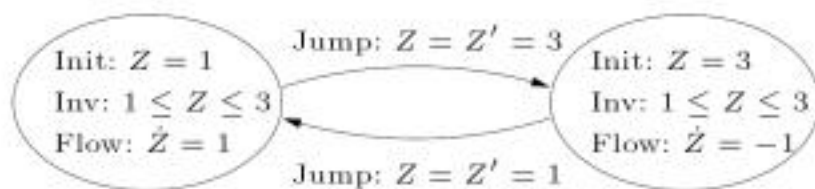
Semi-Algebraic Hybrid Automata.

- Each vertex $v \in V$ is labeled by “initial”, “invariant” and “flow” labels: $\text{Init}(v)$, $\text{Inv}(v)$, and $\text{Flow}(v)$; the labels $\text{Init}(v)$ and $\text{Inv}(v)$ are constraints whose free variables are in Z ; the label $\text{Flow}(v)$ is a constraint whose free variables are in $Z \cup \dot{Z}$;
- – Each edge $e \in E$ is labeled by “jump” conditions: $\text{Jump}(e)$, which is a constraint whose free variables are in $Z \cup Z'$.
- We say that H is **semi-algebraic** if the constraints in Init , Inv , Flow , and Jump are unquantified first-order formulæ over the reals (i.e., over $(\mathbb{R}, +, \cdot, =, <)$).

Semi-Algebraic Hybrid Automata.

- We say that H is in explicit form if each $\text{Flow}(v)$ is of the form
 - $\bigwedge_{i=1}^k \dot{Z}_i = f_i(Z_1, \dots, Z_k). \square$
- Here, we consider only semi-algebraic hybrid automata in explicit form.

Example



- Consider the following semialgebraic automaton in explicit form. $\dot{Z} = 1$
- The initial mode of this hybrid automaton is shown on the left, where from the starting value of $Z = 1$, Z grows with a constant rate of 1. At time $t = 2$, when the automaton reaches a value of $Z = 3$, it jumps to the other mode on the right. In this second mode, Z wanes with a constant rate of -1 and upon reaching the value of $Z = 1$, it jumps back to the initial mode.

Hybrid Automata: Semantics

- Let H be a hybrid automaton of dimension k . For any given control mode $v \in V$, denote with $\Phi(v)$ the set of functions from \mathbb{R}_+ to \mathbb{R}^k satisfying the constraints in $\text{Flow}(v)$.
- In addition, for any given $r \in \mathbb{R}^k$, we use $\text{Init}(v)(r)$ ($\text{Inv}(v)(r)$ and $\text{Flow}(v)(r)$) to denote the Boolean value obtained by pairwise substitution of r with Z in $\text{Init}(v)$ ($\text{Inv}(v)$ and $\text{Flow}(v)$, respectively).
- Similarly, for any given $r, s \in \mathbb{R}^k$, we use $\text{Jump}(e)(r, s)$ to denote the boolean value obtained by pairwise substitution of r with Z and s with Z' in $\text{Jump}(e)$.

Hybrid Automata: Semantics

- **Definition:** Let $H = (Z, V, E, \text{Init}, \text{Inv}, \text{Flow}, \text{Jump})$ be a hybrid automaton of dimension k . A location ℓ of H is a pair $\langle v, r \rangle$, where $v \in V$ is a state and $r \in \mathbb{R}^k$ is an assignment of values to the variables of Z .
 - A location $\langle v, r \rangle$ is said to be admissible if $\text{Inv}(v)(r)$ is satisfied. The continuous reachability transition relation, \rightarrow_C , between admissible locations is defined as follows:

$$\langle v, r \rangle \rightarrow_C \langle v, s \rangle \text{ iff}$$

$$\exists t > 0, f \in \Phi(v)(f(0) = r \wedge f(t) = s \wedge \forall t' \in [0, t](\text{Inv}(v)(f(t')))).$$

- The discrete reachability transition relation, \rightarrow_D , between admissible locations is defined as follows:

$$\langle v, r \rangle \rightarrow_D \langle u, s \rangle \text{ iff } \langle v, u \rangle \in E \wedge \text{Jump}(\langle v, u \rangle)(r, s)$$

Hybrid Automata: Semantics

- A trace of H is a sequence $\ell_0, \ell_1, \dots, \ell_n, \dots$ of admissible locations such that

$$\forall i \geq 0 \ell_i \rightarrow_C \ell_{i+1} \vee \ell_i \rightarrow_D \ell_{i+1}. \square$$

The Bounded Reachability Problem

- Let H be a semi-algebraic k -dimensional hybrid automaton in explicit form...
 - Let $S \subseteq \mathbb{R}^k$ be a set of “start states”, characterized by the first order formula $S(Z)$, and $B \subseteq \mathbb{R}^k$ be a set of “bad states”, characterized by the first order formula $B(Z)$.
- We wish to check that there exists no trace of H starting from a location of the form $\langle v, s \rangle$ with $s \in S$ and reaching a location of the form $\langle u, b \rangle$ with $b \in B$ within a specified time interval $[0, \text{end}]$.
 - If such traces exist we are interested in a characterization of the points of S which reach B in the time interval $[0, \text{end}]$.

An Algebraic Approach

- Apply algebraic/semi-algebraic methods to the descriptions of the traces of the hybrid automaton.
- Thus, the evolution of the automaton can be described even in cases where system parameters and initial conditions are unspecified.
 - Semialgebraic decision procedures provide a succinct description of algebraic constraints over the initial values and parameters for which proper behavior of the system can be expected.
 - By keeping track of conservation principles (e.g., of mass and energy) in terms of constraint or invariant manifolds on which the system must evolve, we avoid many of the obvious pitfalls of numerical approaches.

Some Inherent Shortcomings

- Nonetheless, our method has an inherent incompleteness:
 - We proceed on the traces using a time step δ which implies that our answer is relative to a limited time interval.
 - When the solutions of the differential equations cannot be computed we approximate them using the first few terms of the corresponding Taylor polynomials, hence the error we accumulate depends on δ .

The Basic Case

- Consider a system of differential equations of the form $\dot{Z} = f(Z)$, where \dot{Z} and Z are vectors of length k and f is a function that operates on them.
- Let $S, B \subseteq \mathbb{R}^k$ be characterized by the formulæ $S(Z)$ and $B(Z)$, respectively. As before, let $[0, \text{end}]$ be a time interval and $0 < \delta \leq \text{end}$ be a time step.
- We use $p_j(Z_0, \delta)$ to denote the Taylor polynomial of degree j relative to the solution $Z(t)$ centered in Z_0 with a step size of δ . For instance, $p_1(Z_0, \delta)$ is the vector expression $Z_0 + f(Z_0) \cdot \delta$.

Semialgebraic formulation

- Consider the following first-order formula over the reals

$$\mathbb{F}_\delta(Z_0, Z) \equiv S(Z_0) \wedge \exists \delta' (Z = p_j(Z_0, \delta') \wedge 0 \leq \delta' \leq \delta).$$

- The points reachable from S in the time interval $[0, \delta]$ can be approximated with the set of points satisfying the formula $\exists Z_0 (\mathbb{F}_\delta(Z_0, Z))$. Hence, the points in B and reachable from S in $[0, \delta]$ can be approximated by the formula

$$\exists Z_0 (\mathbb{F}_\delta(Z_0, Z)) \wedge \mathbb{B}(Z).$$

Solving the System

- Symbolic algebraic techniques can be applied in order to both simplify (e.g., eliminate quantifiers) and decide the satisfiability of this formula.
- From these interesting counter-examples can be constructed. Note that:
 - (1) The only approximation we have introduced is due to the use of the Taylor polynomials;
 - (2) We have only used existential quantified formulæ;
 - (3) The degree of the Taylor polynomial together with the degrees of the f_i 's influence the complexity of the first-order formulæ we create and the number of steps needed to get a sufficient precision.

Approximation Issues

- When the derivative of order $j + 1$ of f is bounded we can use the Lagrange Remainder Theorem to both under and over approximate the set of points reachable within the time interval $[0, \text{end}]$ and to estimate the error.
- This method can be generalized to the case in which the f_i 's are rational functions,

Complexity Issues

- When we terminate, we are left with deciding the satisfiability of a semialgebraic formula involving
 - # variables = $n = 2 + k \cdot \lceil \text{end}/\delta \rceil + N(\mathbb{S}) + N(\mathbb{B})$
 - degree = $d = \max[j + \deg(f), \deg(\mathbb{S}), \deg(\mathbb{B})]$,
where N and \deg denote the number of variables and total degree, respectively used in the semialgebraic description of \mathbb{S} and \mathbb{B} .
 - Assume that the coefficients of the polynomials can be stored with at most L bits
- The total time complexity (bit-complexity) of the decision procedure is
$$(L \log L \log \log L) d^{O(n)}.$$

A Toy Example

- Dynamics described by a system of ODE's

$$\dot{Z} = 2Z^2 + Z,$$

- with S and B characterized by $S \equiv Z > 4$ and $B \equiv Z^2 < 4$.
- Now, consider the time interval $[0, 0.5]$ and time step 0.5. After time 0.5, using an approximation with Taylor polynomial of degree 2, we get

$$\exists Z_0, \delta' (Z_0 > 4$$

$$\wedge Z = Z_0 + (2Z_0^2 + Z_0) \cdot \delta' + (8Z_0^3 + 6Z_0^2 + Z_0) \cdot (\delta')^2/2$$

$$\wedge 0 \leq \delta' \leq 0.5 \wedge Z^2 < 4).$$

- This formula is **unsatisfiable**, thus implying that the dynamical system reaches no bad states in the specified time interval $[0, 0.5]$.

Other Related Ideas

- The formulæ involved in our method can be easily simplified, if we introduce further approximations.
- For instance, we may approximate reachability by first evaluating the maxima and the minima of the j -th Taylor polynomial $p_j(Z, \delta')$ s over S and $[0, \delta]$, and then using them as upper and lower bounds.

Example

- Consider the differential equation

$$\dot{Z} = 2Z,$$

- with S and B characterized by

$$S \equiv 2 \leq Z \leq 4 \text{ and } B \equiv 3 < Z < 5.$$

- The Taylor polynomial of degree 1 with $\delta = 0.5$ is $Z + 2Z \cdot \delta'$, i.e., $2Z$.
- Note that since the maximum and the minimum in S are 8 and 4, respectively, and since the interval $[4, 8]$ intersects $(3, 5)$, S reaches B in time 0.5.

The General Case

- We consider a polynomial k-dimensional hybrid automaton H in explicit form.
 - Given a mode v of H , we use the notation $p_{j,v}(Z, \delta)$ to denote the Taylor polynomial of degree j in the mode v centered in Z . The first-order formula

$$\mathbf{F}[v, S](Z_0, Z) \equiv S(Z_0) \wedge \exists \delta' (Z = p_{j,v}(Z_0, \delta') \wedge 0 \leq \delta' \leq \delta \\ \wedge \forall \delta'' (0 \leq \delta'' \leq \delta' \rightarrow \text{Inv}(v)(p_{j,v}(Z_0, \delta''))))$$
 characterizes the points reached within time δ in the mode v , under the approximation implied by the Taylor polynomial.
 - As before, the formula

$$\exists Z_0 (\mathbf{F}[v, S](Z_0, Z) \wedge \mathbf{B}(Z))$$
 is satisfiable if and only if the set B can be reached from S without leaving mode v within the time step δ .

•

- In this case, the points of S which reach B are characterized by

$$\exists Z(\mathbb{F}[v, S](Z_0, Z) \wedge \mathbb{B}(Z)).$$

- If the preceding formula is not satisfiable, we have to consider all possible alternative situations: that is, either we continue to evolve within the mode v or we discretely jump to another mode, $u \in V$.

••

- Define the formula S_{δ}^{vu}

$$S_{\delta}^{vu}(Z) \equiv \begin{cases} \exists Z_0 (\mathbb{F}_{\delta}^v(Z_0, Z)), & \text{if } u = v; \\ \exists Z_0, Z_1 (\mathbb{F}_{\delta}^v(Z_0, Z_1) \wedge \text{Jump}(\langle v, u \rangle)(Z_1, Z)), & \text{otherwise.} \end{cases}$$

- representing the states reached within time δ in the mode u .
- In the worst case one generates $|E|$ satisfiable formulæ on which one must iterate the method, treating them as we treated $S(Z)$ in the first step.

...

- At each step, we must check the minimum elapsed time before a jump can be taken. Let $\mathbb{M}(Z) \equiv S^{\forall, u, \dots, w}_{\delta}(Z)$ be one of the formulæ obtained after some number of iterations.
- Suppose now that we intend to jump from this mode w to the next mode z . This can be formulated as:

$$\begin{aligned} T(w, z, \mathbb{M})(T) \equiv \exists Z_0, Z_1, Z \Big(& \mathbb{M}(Z_0) \wedge Z_1 = pj_w(Z_0, T) \wedge 0 \leq T \leq \delta \\ & \wedge \forall T' (0 \leq T' \leq T \rightarrow Inv(v)(pj_v(Z_0, T'))) \wedge Jump(\langle w, z \rangle)(Z_1, Z) \Big). \end{aligned}$$

....

- The minimum amount of time can now be computed as solution of the formula

$$\text{Min}(w, z, \mathbf{M})(T) \equiv T(w, z, \mathbf{M})(T) \wedge \forall T' (T' < T \rightarrow \neg T(w, z, \mathbf{M})(T')).$$

- Along each generated path we have to iterate until the sum of the minimum amounts reaches end.
- If all the paths accumulate a total amount of time greater than end and B is never reached we can be sure that B cannot be reached from S in the time interval $[0, \text{end}]$.



- If B is reached, i.e., one of the formulæ involving B is satisfiable before m iterations, then we can be sure that B is reachable from S in the time interval $[0, \text{end}]$.
- If B is reached after the first m iterations, then B is reachable from S but we are not sure about the elapsed time, since we keep together flows of different length.
- It is possible that some paths do not accumulate a total time greater than end , e.g., the sequence of the minimum times converges rapidly to 0. In this case our method could not converge.
- We can also exploit here the Lagrange Remainder Theorem to both under and over approximate the set of reachable points and to estimate errors.

Time-Complexity

- Assume that no path accrues more than M discrete jumps (i.e., the method has converged). Thus we have a quantified semialgebraic formula with $O(M)$ alternations and involving
 - #variables = $n = k \cdot [\lceil \text{end}/\delta \rceil + O(M)] + N(S) + N(\mathbb{B})$
 - degree = $d = \max[j + \deg(\text{Init}, \text{Inv}, \text{Jump}), \deg(S), \deg(\mathbb{B})]$, where N and \deg denote the number of variables and total degree, respectively as before.
 - Assume that the coefficients of the polynomials can be stored with at most L bits. Then the total time complexity (bit-complexity) of the decision procedure is
$$(L \log L \log \log L) d^{2^{O(n)}}.$$
double-exponential in the number of variables.

Future History of Systems Biology

Challenges from Hybrid Systems

- **Problem 1:** What are the most important hybrid systems models for biology?
 - High Fidelity
 - Expressivity
 - Decidability
 - Computational Efficiency

Dynamics

- Replacing differential equations by "equivalent" dynamics:

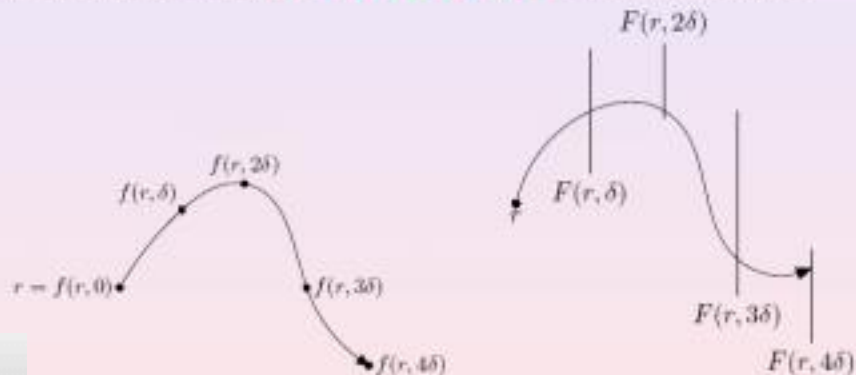
If $f(X, T)$ is the solution of $\dot{X} = \mathcal{F}(X, T)$, then

$$\dot{X} = \mathcal{F}(X, T) \quad \text{and} \quad \text{Dyn}[X, X', T] \equiv X' = f(X, T),$$

are equivalent

Inclusion Dynamics

We are interested in **inclusion dynamics** defined by formulæ



$$\text{Dyn}[X, X', T] \equiv X' = f(X, T) \quad \text{Dyn}[X, X', T] \equiv X' \in F(X, T)$$

Michael's Form

- Let $F_x^V(T) \equiv \{X' \mid \text{Dyn}(v)[X, X', T] \wedge \text{Inv}(v)[X']\}$
- A Hybrid automaton is in Michael's form if
 - F_x^V is lower semi-continuous
 - For each $t \in I_x^V$ the set $F_x^V(t)$ is closed and convex
 - where I_x^V is the largest $[0, t')$ such that $F_x^V(t) \neq \emptyset, \forall t \in [0, t')$.

Theorem

If H is in Michael's form, then $s \in F_r^V(t)$ iff $\langle v, r \rangle \xrightarrow{t}_C \langle v, s \rangle$.

Reachability

Michael's Form and Reachability

For each automaton in Michael's form, we can write a formula $Reach(H, ph)[X, X', T]$, where $ph = v_0, \dots, v_n$ is a path on $\langle \mathcal{V}, \mathcal{E} \rangle$, such that

$$\begin{array}{ccc} Reach(H, ph)[X, X', T] & \iff & \begin{array}{l} H \text{ reaches } \langle v_n, X' \rangle \\ \text{from } \langle v_0, X \rangle \text{ with a} \\ \text{trajectory} \\ \text{corresponding to } ph \end{array} \\ \text{holds} & & \end{array}$$

Time must not be infinite!!

Two New Models

FOCoRe and IDA

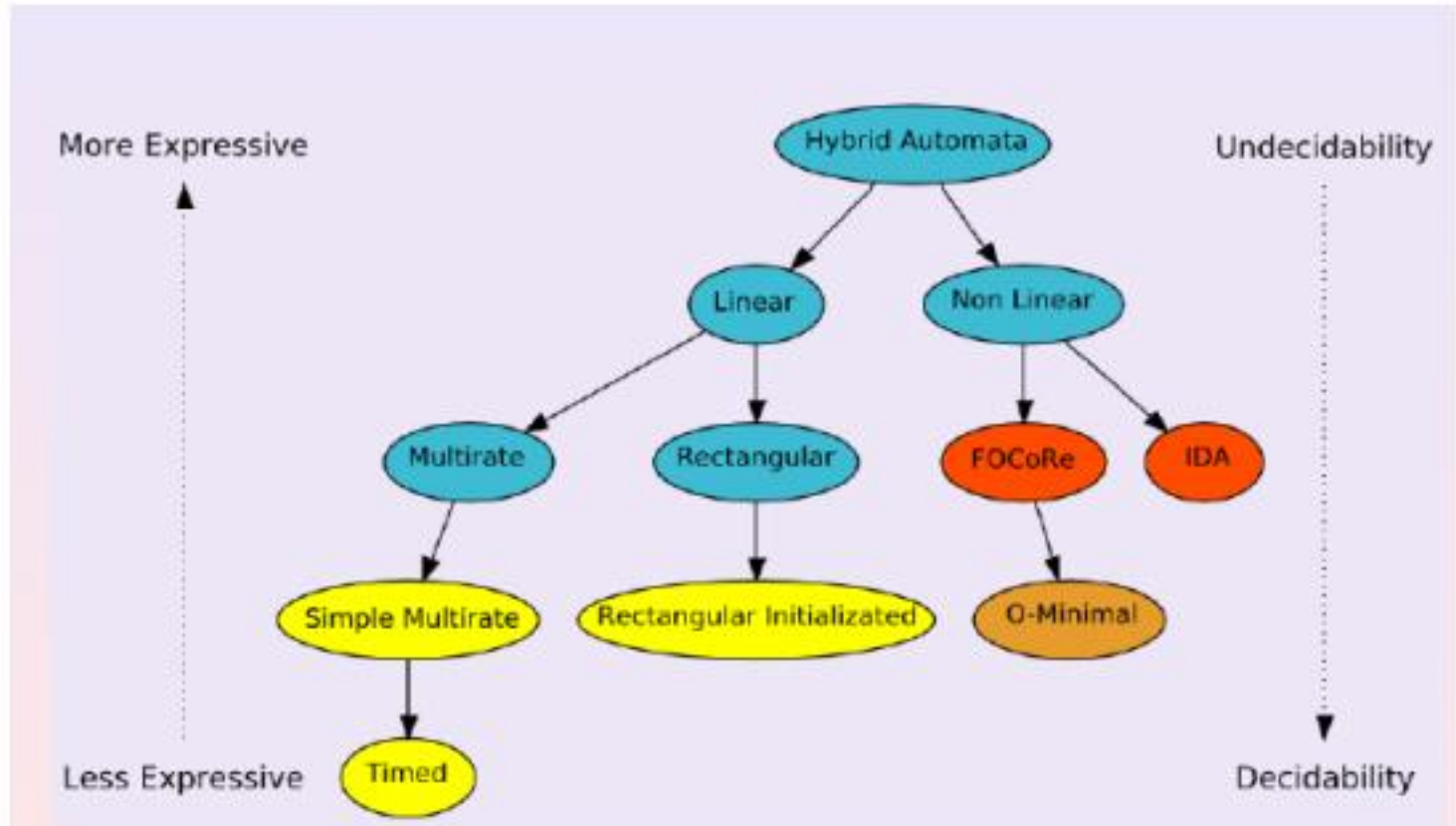
FOCoRe (**F**irst **O**rders **C**onstant **R**eset hybrid automata) are first-order hybrid automata:

- in Michael's form
- with **constant resets** (i.e., $\text{Reset}(e)[X, X']$ does not depend on X)

IDA (**I**ndependent **D**ynamics hybrid **A**utomata) allows **identity resets** between locations whose dynamic does not change

We can reduce reachability problem for either FOCoRe or IDA \mathcal{T} -automata to a satisfiability problem for formulæ of \mathcal{T}

Hybrid Hierarchy



Formal Understanding of Genetics

- **Problem 2:** How do genotypes determine phenotypes?
 - Take the view that the model (e.g., structure of the hybrid systems and their parameters) encodes the genotypes, where as the temporal logic formulæ satisfied by the model encode the phenotypes?
 - How does biology relate them?
 - Are there engineering principles governing them?



- What properties are important?
- Are some parameters more "flexible," "facilitated" or "robust" than others?
- Are there symmetries? Scaling laws? Laws governing compositionality? Laws governing modularity and hierarchy?
- How does evolution control them?
- What structural changes can be accomplished by nature?
- What selection forces act on them? What invariants are important to biology? Are there utility functions that are being optimized? What are they?

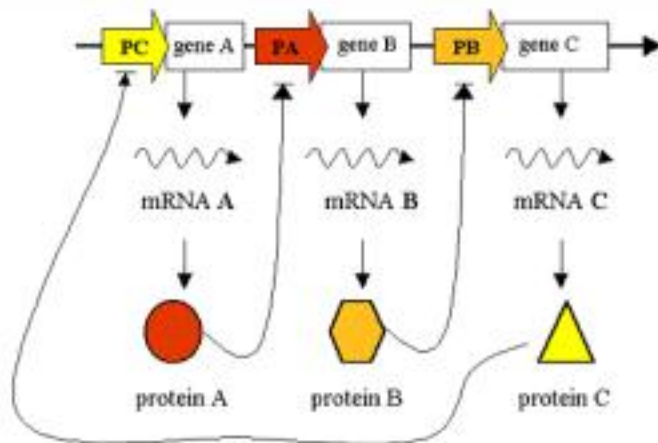
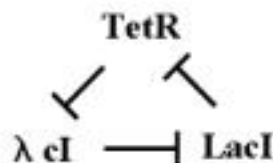
Systems Biology & Synthetic Biology

- **Problem 3:** How can our understanding of systems biology lead to designing useful artificial biological systems?
 - Can we create ab initio a complete artificial organism?
 - Can we perturb the properties of an existing cell by introducing synthesized biological circuits?

An Artificial Clock

The Repressilator:

a cyclic, three-repressor, transcriptional network



- Three proteins:

- LacI, tetR & λ cI
- Arranged in a cyclic manner (logically, not necessarily physically) so that the protein product of one gene is repressor for the next gene.

$\text{LacI} \rightarrow \neg \text{tetR}; \text{tetR} \rightarrow \text{TetR}$

$\text{TetR} \rightarrow \neg \lambda cI; \lambda cI \rightarrow \lambda cI$

$\lambda cI \rightarrow \neg \text{lacI}; \text{lacI} \rightarrow \text{LacI}$

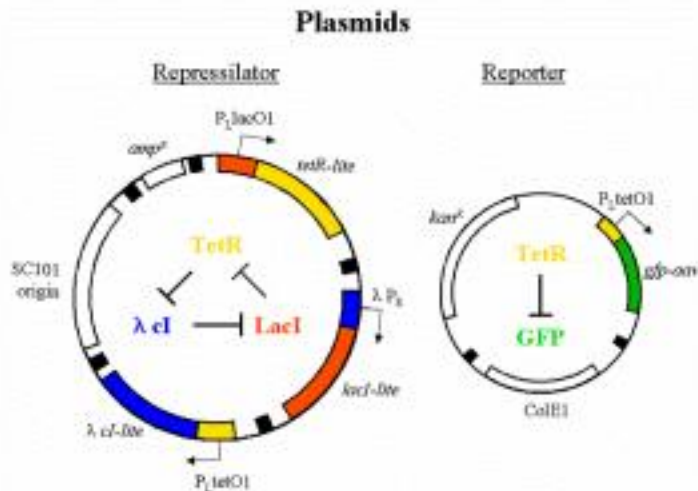
Cycles of Repression

- The first repressor protein, LacI from *E. coli* inhibits the transcription of the second repressor gene, tetR from the tetracycline-resistance transposon Tn10, whose protein product in turn inhibits the expression of a third gene, cI from λ phage.
- Finally, CI inhibits lacI expression,
- completing the cycle.

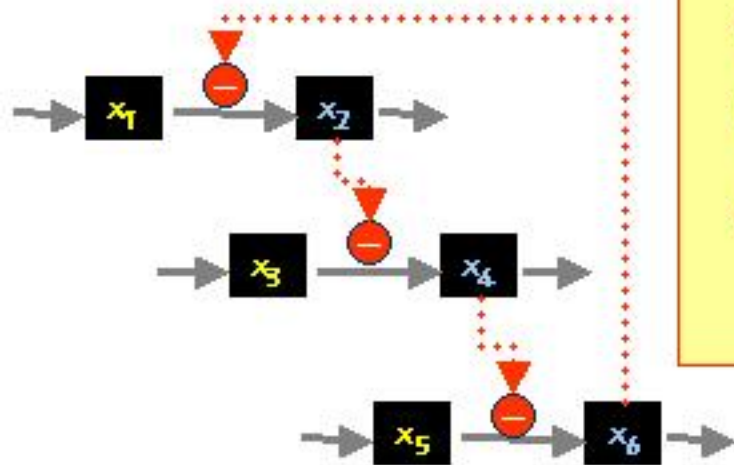
Biological Model

- Standard molecular biology: Construct

- A low-copy plasmid encoding the repressilator and
- A compatible higher-copy reporter plasmid containing the tet-repressible promoter PLtet01 fused to an intermediate stability variant of gfp.



Cascade Model: Repressilator?



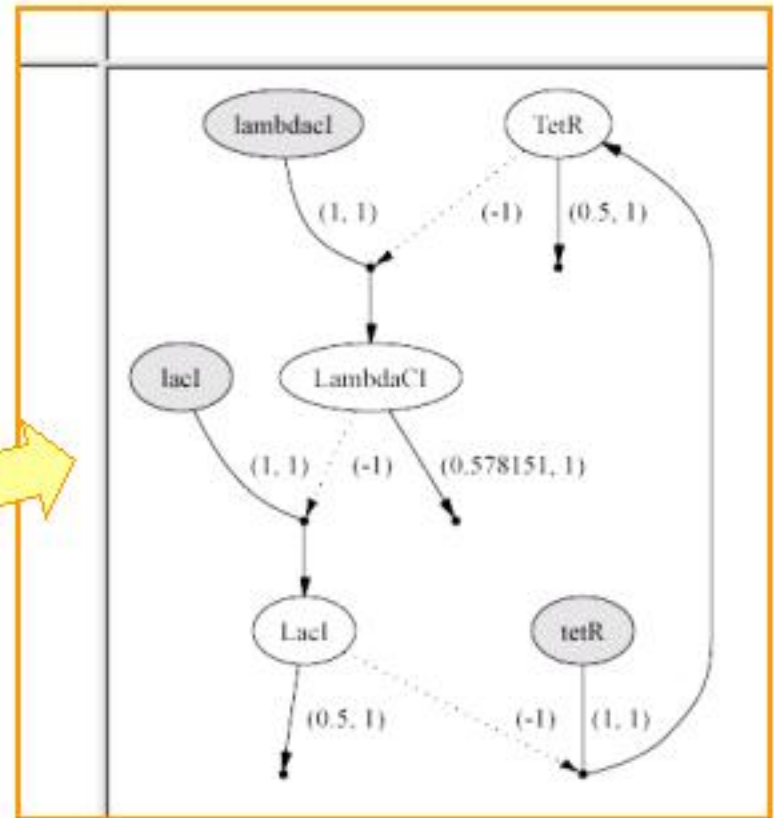
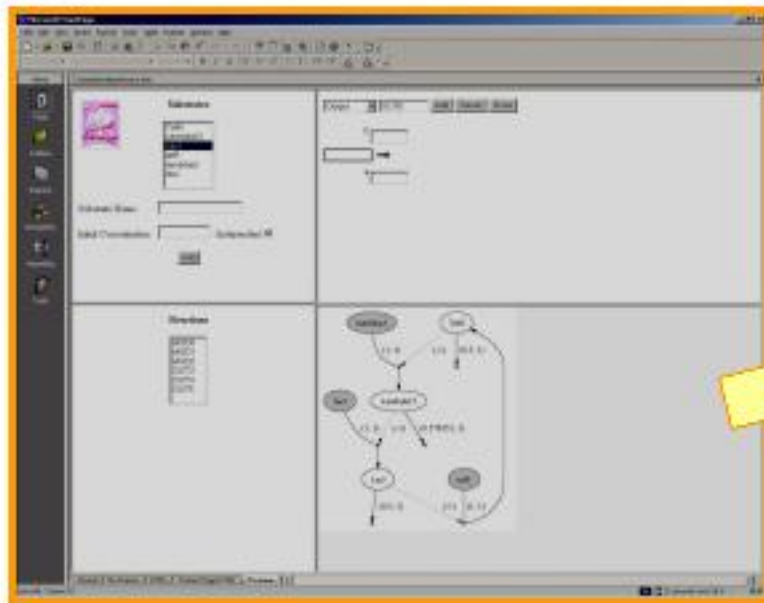
$$\frac{dx_2}{dt} = \alpha_2 X_6^{g_{26}} x_1^{g_{21}} - \beta_2 x_2^{h_{22}}$$

$$\frac{dx_4}{dt} = \alpha_4 X_2^{g_{42}} x_3^{g_{43}} - \beta_4 x_4^{h_{44}}$$

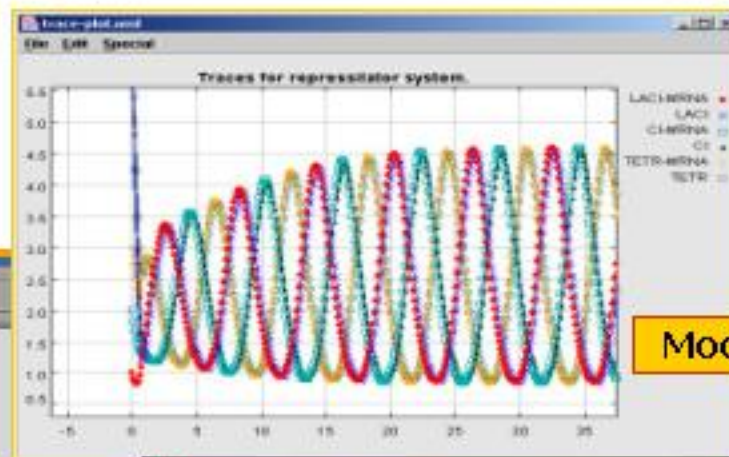
$$\frac{dx_6}{dt} = \alpha_6 X_4^{g_{64}} x_5^{g_{65}} - \beta_6 x_6^{h_{66}}$$

$$X_1, X_3, X_5 = \text{const}$$

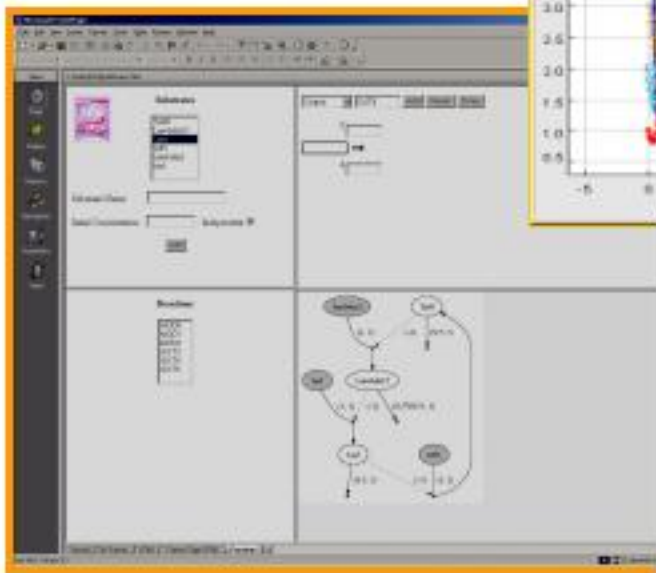
SimPathica System



Simpathica System



Model Simulation



XS-systems Trace Analysis

XSYS File Tools Help

default

Intersection Trace Details System Messages

```
7 steady_state().  
The formula  
STEADY_STATE()  
is always true over the trace.  
7 Always( not (PRPP > 40)  
or (steady_state() and  
Eventually(IMP > 100)) and  
Eventually(HX < 10) and  
Eventually( Always((IMP > 97) and (IMP < 99))) and  
Eventually( Always((HX > 9) and (HX < 11))))).  
The formula  
ALWAYS(NOT((PRPP > 40)  
or AND(STEADY_STATE(),  
EVENTUALLY((IMP > 100)),  
EVENTUALLY((HX < 10)),  
EVENTUALLY(ALWAYS((IMP > 97) and (IMP < 99))),  
EVENTUALLY(ALWAYS((HX > 9) and (HX < 11))))))  
is true over the trace.  
7 □
```

ted trace: "default"

Model Checking

Model Building

- **Problem 4:** How do we build models?
What measurements are important in this process?
 - Causal Models vs. Phenomenological Models
 - Structural properties of models. Are they important?

Applications

- **Problem 5:** How can we apply our understanding of systems biology to solve important biomedical problem?
 - Diagnostics and prognostics. If a particular aspects of a model change, what properties are affected?
 - Drug discovery. Vaccine design. Better elucidation of ADME/Tox profiles, as well as optimizing clinical trial efficiency.
 - Genetically modified food

The End