It's a Bitwise World

Daniel Kroening

Joint work with I. Zinovik (ETH) and Y. Chebiryak (ETH)

Oxford University, Computing Laboratory

Outline

Literature

- *An Algebraic Algorithm for the Identification of Glass Networks with Periodic Orbits along Cyclic Attractors*, Algebraic Biology 2007 (tech report available with more detail)
- *Computing Binary Combinatorial Gray Codes via Exhaustive Search with SAT-Solvers*, IEEE Information Theory 2008 (tech report available with more detail)
- *Classification of Hamiltonian Cycles in the 6-Cube*, Journal of Satisfiability 2007

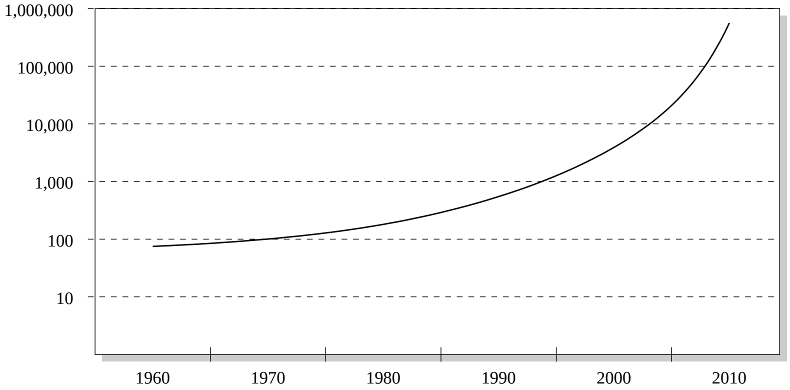
Vision

- Models of biological systems are *hybrid*
  - Continuous variables: concentrations, ...
  - Amenable to numerical simulation
  - Systems Biology: there is significant switching behavior

Vision

- Models of biological systems are hybrid
- Idea: Leverage enormous performance of propositional satisfiability (SAT) solvers

SAT


1,000,000
100,000
10,000
1,000
100
10
1960
1970
1980
1990
2000
2010

Vision

- Models of biological systems are hybrid
- Idea: Leverage enormous performance of propositional satisfiability (SAT) solvers
- Driven by annual competitions
- Main selling point: robustness

Satisfiability Modulo Theories

Idea: Extend the DPLL algorithm with solvers for first-order theories

- Equality logic
- Linear arithmetic over reals/integers
- Arrays
- Lists
- Quantified logics
- and combinations!

Vision

- Motivation:

  assist biologists in construction of dynamic models
  that behave consistently with observations
  of cell regulatory networks

Simulate biologists! Not biology!
–Bud Mishra

Overview

- Modeling approach:
  multi-dimensional piecewise linear differential equations
  approximating kinetics of biochemical reactions

- Analysis method:
  algebraic analysis of the return map
  based on a corollary of the Perron-Frobenius theorem

Multistationarity

Cell differentiation:

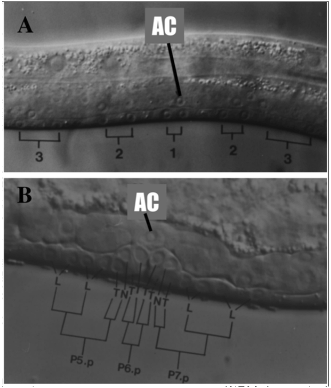
"In multicellular organisms, cell lines differ, not by which genes are present, but by which genes are on and which are off."

Multi steady-state models:

"Multistationarity is the property of systems whose structure is such that they can display two or more distinct, isolated steady states."

R. Thomas. Chaos, 2001

The development of the vulva of C. Elegans



Nomarski images of developing vulva. Left lateral views of larval hermaphrodites (anterior to the left). AC, anchor cell. 3, 3 VPCs (P4.p and P8.p shown here); 2, 2 VPCs (P5.p and P7.p); 1, 1 VPC. (A) mid-L3 stage. *Wormbook.org*

A Multistationary Model for Dimerisation of bHLH Proteins

$$\frac{\mathrm{d}x_1}{\mathrm{d}t} = -x_1 + \frac{\sigma x_1^c}{1+\sum_{i=1}^n x_i^c} + \alpha$$

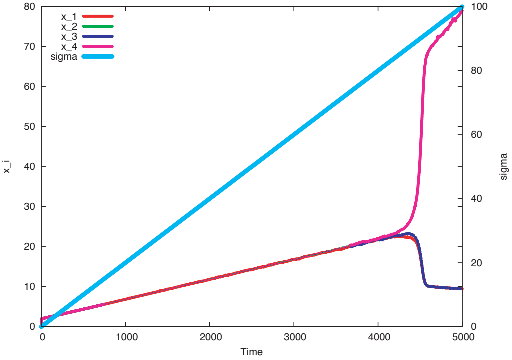
$$\vdots$$

$$\frac{\mathrm{d}x_n}{\mathrm{d}t} = -x_n + \frac{\sigma x_n^c}{1+\sum_{i=1}^n x_i^c} + \alpha$$

The four elements are initially coexpressed at an identical level, which increases with the transcriptional strength sigma;

when sigma reaches a threshold level, one element is upregulated, and others are downregulated.

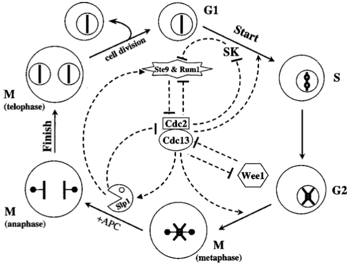
(O. Cinquin, J. Theor. Bio., 2005)



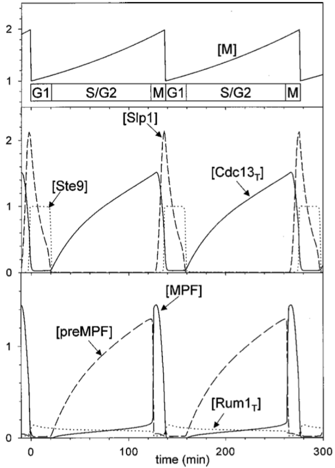
Time evolution of the concentrations of four switch elements

A model of the Cell Division Cycle of Fission Yeast

A model of the cell division cycle of fission yeast includes 12 non-linear differential and algebraic equations



The eukaryotic cell division cycle. The outside circle shows the major steps of DNA synthesis and mitosis. The inner diagram shows the relationships among the principal molecular components of the cell cycle engine



(B. Novak, Chaos, 2001)

Genes and Their Products

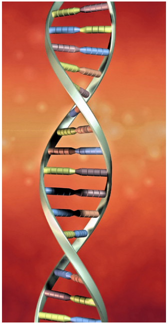
- We consider n genes, where each gene has a product
- Let x_i denote the concentration of the product of gene i
- Dynamics:

$$\dot{x}_i = -g_i(x_1, \ldots, x_n) - \gamma_i x_i \quad \text{for } 1 \leq i \leq n,$$

where

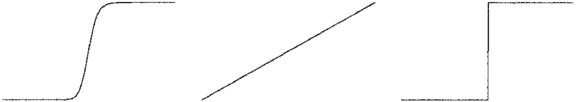
$\gamma_i > 0$: degradation rate of x_i

$g_i : \mathbb{R}^n_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$: coupling



Approximating Biochemical Kinetics

Nonlinearities often approximate switching behavior



$$g_i(x_1,\ldots,x_n) = \sum_{l\in L} k_{il} b_{il}(x_1,\ldots,x_n)$$

where

- $k_{il} \geq 0$: rate parameter
- L: set of indexes
- $b_{il} : \mathbb{R}^n_{\geq 0} \to \{0,1\}$: composition of step functions with steps located at $x_i = \theta_{il}$

Coupling is usually very localized: $|L|$ is small

Modeling with Glass Networks

- θ_{il}: l-th threshold concentration of the protein encoded by gene i
- Induces a partitioning of the phase space into a set of n-dimensional boxes
- In each box, the protein concentrations are described by ODEs with a constant production term μ_i and a rate parameter γ_i:

$$\dot{x}_i = \mu_i - \gamma_i x_i \quad \text{for } 1 \leq i \leq n$$

Modeling with Glass Networks

- θ_{il}: l-th threshold concentration of the protein encoded by gene i
- Induces a partitioning of the phase space into a set of n-dimensional boxes
- In each box, the protein concentrations are described by ODEs with a constant production term μ_i and a rate parameter γ_i:

$$\dot{x}_i = \mu_i - \gamma_i x_i \quad \text{for } 1 \leq i \leq n$$

- Further restrictions:
  - Gene activity is on/off only
  - decay rates are identical for all reactions

$\longrightarrow$ *Glass model*

Modeling with Glass Networks

The general form of a Glass network is

$$\dot{x}_i = G_i(\tilde{x}_1, \ldots, \tilde{x}_n) - \alpha x_i \quad \text{for } 1 \leq i \leq n$$

where

- $\alpha > 0$,
- G_i: interaction functions,
- $\tilde{x}_i = \begin{cases} a & : \text{ if } x_i < \theta_i \\ b & : \text{ if } x_i > \theta_i \end{cases}$
- with real constants $a < b$

Glass networks may exhibit aperiodic and chaotic behavior!

Modeling with Glass Networks

Apply scaling of the variables and obtain:

$$\dot{y}_i = F_i(\tilde{y}_1, \ldots, \tilde{y}_n) - y_i \quad \text{for } 1 \leq i \leq n$$

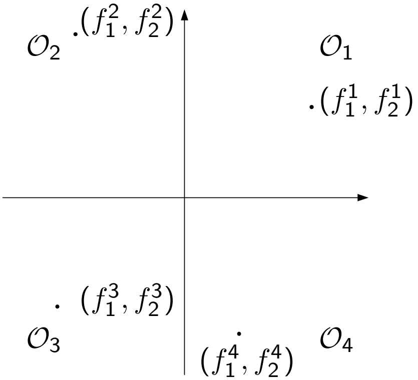
where

$$\tilde{y}_i = \begin{cases} 0 & : & \text{if } y_i < 0 \\ 1 & : & \text{if } y_i > 0 \end{cases}$$

Consequence:

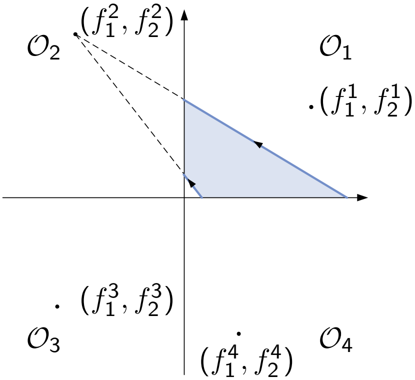
- All thresholds equal 0, unit decay rate
- Integration is easy
- The trajectories are straight lines in every orthant $\mathcal{O}_k$, $k \in \{1, 2, 3, \ldots, 2^n\}$

Phase Flow Diagrams



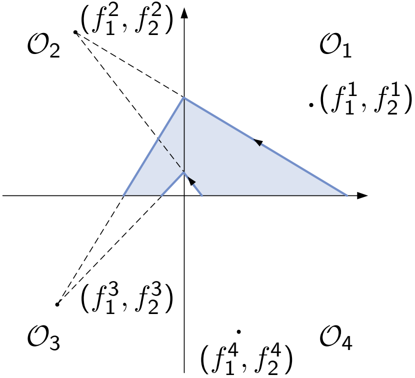
Flow in each orthant $\mathcal{O}_k$ is defined by its *focal point* $\vec{f}^k = (f_1^k, f_2^k, \ldots, f_n^k) \in \mathbb{R}^n$

Phase Flow Diagrams



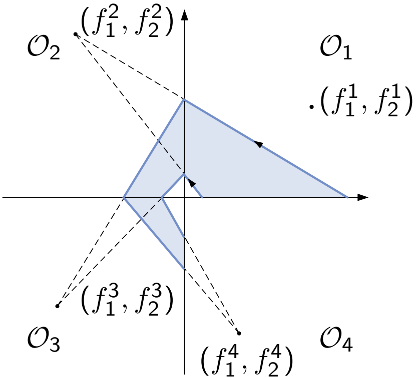
Flow in each orthant $\mathcal{O}_k$ is defined by its *focal point* $\vec{f}^k = (f_1^k, f_2^k, \ldots, f_n^k) \in \mathbb{R}^n$

Phase Flow Diagrams



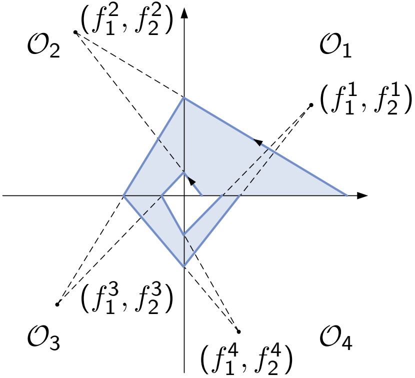
Flow in each orthant $\mathcal{O}_k$ is defined by its *focal point* $\vec{f}^k = (f_1^k, f_2^k, \ldots, f_n^k) \in \mathbb{R}^n$

Phase Flow Diagrams



Flow in each orthant $\mathcal{O}_k$ is defined by its *focal point* $\vec{f}^k = (f_1^k, f_2^k, \ldots, f_n^k) \in \mathbb{R}^n$

Phase Flow Diagrams



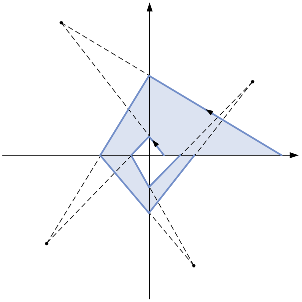
Flow in each orthant $\mathcal{O}_k$ is defined by its *focal point* $\vec{f}^k = (f_1^k, f_2^k, \ldots, f_n^k) \in \mathbb{R}^n$

Objective

- Identify networks with cyclic attractors that exhibit a flow given by a set of focal points

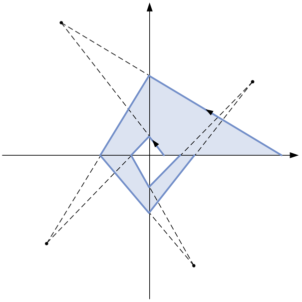
- Existing algorithms:
  - rely on numerical solutions to eigenvalue problems
  - typically $n \approx 4$

Transition Diagrams

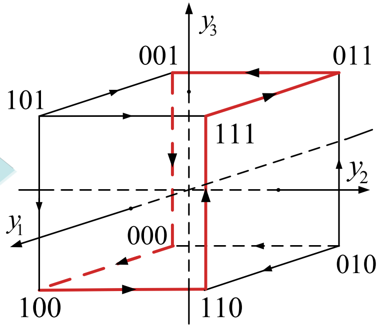


Phase Flow

Transition Diagrams



Phase Flow



Transition Diagram

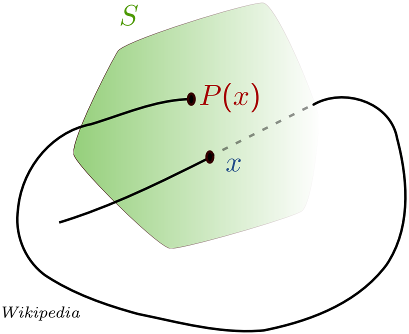
Cyclic Attractors

A cycle in the transition diagram is called a cyclic attractor if

1. it is a chord-free simple cycle in the n-cube, and
2. all edges adjacent to the cycle are directed towards the nodes of the cycle

Poincaré Return Maps

Given some section S, the Poincaré map P tells you where a given point x will first return to S



Henri Poincaré

The Poincaré Return Map for Glass Networks

Theorem: The flow along a cyclic attractor either

- converges to the origin, or
- has a unique stable 1-period orbit

The Poincaré Return Map for Glass Networks

Theorem: The flow along a cyclic attractor either

- converges to the origin, or
- has a unique stable 1-period orbit

Distinguish using the Poincare map:

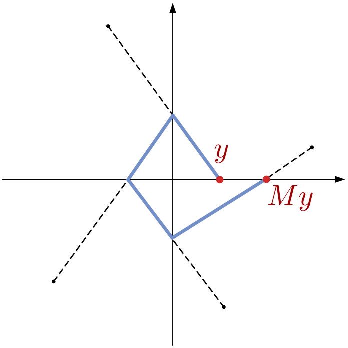
The flow has a stable orbit $\iff$
The Poincaré map has a stable fixed point

The Poincaré Return Map for Glass Networks

$$M\vec{y} = A\vec{y}/(1 + \langle \phi, \vec{y} \rangle)$$

where

- $\psi^{(k)}$: $-e_j/f_j^k$
- $B^{(k)} = I - (\vec{f}^k \vec{e}_j^T)/f_j^k$
- $A = ||a_{mp}||$ is the $(n-1) \times (n-1)$ matrix obtained by deleting the i-th column and row of $B^{(L)} \dots B^{(1)}$
- ϕ: the same reduction of $\psi^{(L)} \dots \psi^{(1)}$



R. Edwards, Physica D, 2000

An Algebraic Criterion for Flow Identification

Theorem (L. Glass, 1978)

the flow admits a stable periodic orbit $\iff$
the dominant eigenvalue r *of the positive matrix* A *is greater than one*

An Algebraic Criterion for Flow Identification

Theorem (L. Glass, 1978)

the flow admits a stable periodic orbit $\iff$
the dominant eigenvalue r *of the positive matrix* A *is greater than one*

Theorem (Gantmacher, 1974)

A real number λ *is greater than the maximal characteristic value* r *of a (non-negative) matrix* A *if and only if for this* λ *all the successive principal minors of the characteristic matrix* $\lambda I - A$ *are positive*

(Corollary of the P-F theorem)

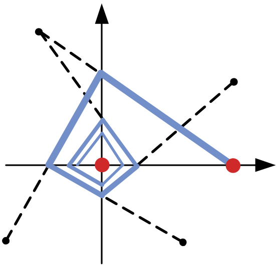
An Algebraic Criterion for Flow Identification

A new algebraic criterion for converging flow:

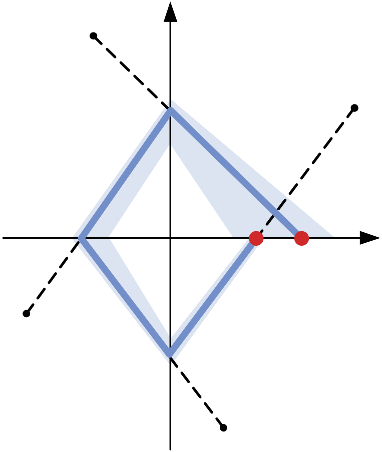
$$\Psi^{con} = \begin{array}{l} (det(A - I)^{(1)} < 0) \wedge \\ (det(A - I)^{(2)} > 0) \wedge \\ (det(A - I)^{(3)} < 0) \wedge \ldots \wedge \\ (det(A - I)^{(n-1)} \gtrless 0) \end{array}$$

where $det(A - I)^{(k)}$ is the determinant of the upper-left $k \times k$ matrix

An Algebraic Criterion for Flow Identification



Ψ^{con} true



Ψ^{con} false

The Search for Cyclic Attractors

The search for attractors is computationally very demanding:

- Glass used exhaustive enumeration for up to 5 variables
- The length of longest attractor is not known for cubes with $n > 7$
- The matrix A for the Poincare map can be calculated in $2\,n!\,N$ ways that specify equivalent cycles up to cube symmetry transformations

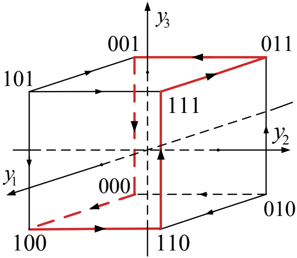
Computing Cyclic Attractors using SAT

First do an encoding of the desired paths through the n-cube

$$\Psi^{cycle} = \bigwedge_i \bigwedge_j [H^1_{i,j} \Leftrightarrow (H^0_{i-1,j} \vee H^0_{i+1,j})]$$

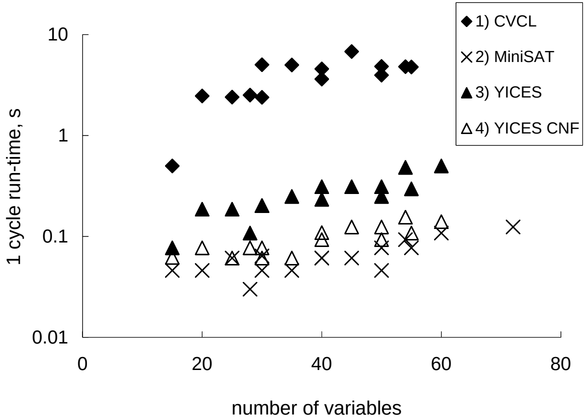
$$\Psi^{gray} = \bigwedge_{i=1}^{N-1} H^1_{i,i+1} \wedge H^1_{1,N}$$

$$\Psi^{ind} = \Psi^{gray} \wedge \Psi^{cycle}$$



H^k_{ij} denotes that the *Hamming distance* between the bit vectors i and j is k

Experimental Results



Run-time for the search for one cycle

Some Data on Induced Cycles

Dimension, n	Length, N	#induced cycles	#equivalence-classes
5	10	126	**9**
	12	42	5
	14	70	3
6	16	92436	385
	18	247806	1066
	20	220440	981
	22	121572	465
	24	23232	103
	26	780	4

Lower bounds for the number of induced cycles
Dimension 5: compare to results from Glass

Symmetry Breaking

There are a number of ways to deal with the symmetries:

1. Eliminate symmetries upfront
2. Find a solution, then block all the equivalent ones (∼all-SAT)
3. (New) Compute an over-approximation, then filter (see JSAT paper)

Computing Cyclic Attractors using SMT

Now add constraints for return map

$$\Psi^{con} = \begin{array}{l} (det(A-I)^{(1)} < 0)\wedge \\ (det(A-I)^{(2)} > 0)\wedge \\ (det(A-I)^{(3)} < 0) \wedge \ldots \wedge \\ (det(A-I)^{(n-1)} \gtrless 0) \end{array}$$

(Convergence of flow)

Computing Cyclic Attractors using SMT

Now add constraints for return map

$$\Psi^{con} = \begin{array}{l} (det(A-I)^{(1)} < 0) \wedge \\ (det(A-I)^{(2)} > 0) \wedge \\ (det(A-I)^{(3)} < 0) \wedge \ldots \wedge \\ (det(A-I)^{(n-1)} \gtrless 0) \end{array}$$

(Convergence of flow)

$$\Psi^{pos} = \bigwedge_m \bigwedge_p (a_{mp} > 0)$$

(Matrix A positive)

Computing Cyclic Attractors using SMT

Now add constraints for return map

$$\Psi^{con} = \begin{array}{l} (det(A-I)^{(1)} < 0) \wedge \\ (det(A-I)^{(2)} > 0) \wedge \\ (det(A-I)^{(3)} < 0) \wedge \ldots \wedge \\ (det(A-I)^{(n-1)} \gtrless 0) \end{array}$$

(Convergence of flow)

$$\Psi^{pos} = \bigwedge_m \bigwedge_p (a_{mp} > 0)$$

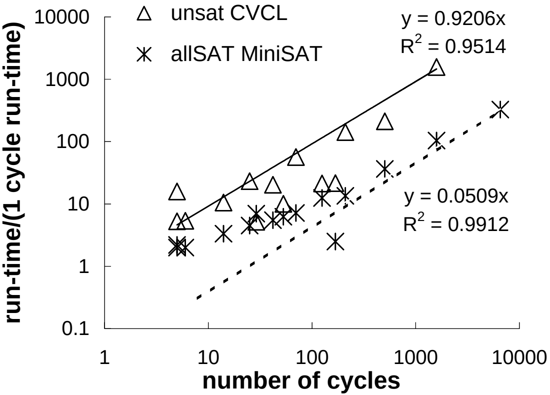
(Matrix A positive)

$$\Psi^{ind} = \Psi^{gray} \wedge \Psi^{cycle}$$

(propositional part)

Experimental Results

Check if $\Psi^{ind} \wedge \Psi^{pos} \wedge (\neg\Psi^{con})$ is satisfiable



Conclusions

- We have shown a method to reason about some ODEs with propositional logic
- The core of many problems over continuous variables is a bit-vector problem

Conclusions and Future Work

- Now able to produce cyclic attractors with custom properties efficiently
  (we are currently leading the search for Gray codes)

- Example application: Bifurcation analysis, stability of phase flow (in the AB paper)

- Current work:
  hack cylindrical decomposition into a Model Checker