Finding Lean Induced Cycles in Binary Hypercubes

Yury Chebiryak[1] Thomas Wahl[2]
Daniel Kroening[2] Leopold Haller[2]

[1]Computer Systems Institute, ETH Zurich, Switzerland
[2]Computing Laboratory, Oxford University, UK

SAT 2009 - Twelfth International Conference on Theory and Applications of Satisfiability Testing
June 30 - July 3, 2009, Swansea, Wales, United Kingdom.

This research was supported in part by an award from IBM Research and by ETH Research Grant TH-19 06-3.

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

UNIVERSITY OF OXFORD

Chinese Ring Puzzle

aka. The Devils Needle Puzzle aka. Cardan's rings

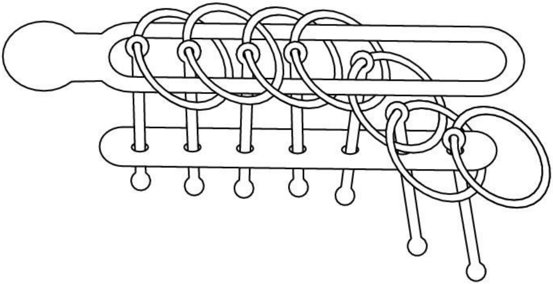


Figure adopted from [Knuth'05]

Empty the bar by

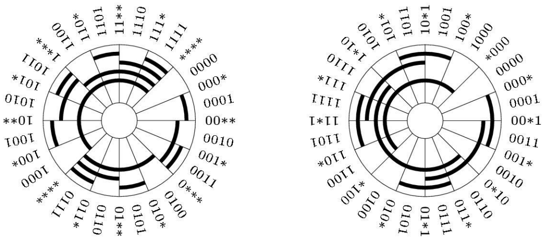
1. removing or replacing rightmost ring or
2. moving any ring if right ring is on bar and rest are off

Binary Reflected Gray Code

Gray codes applications:

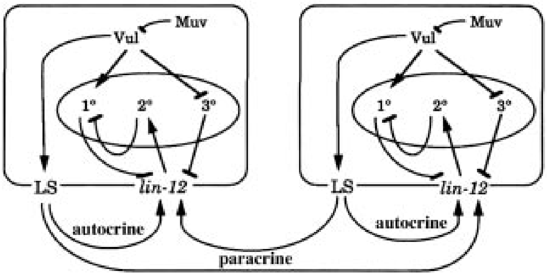
- analog to digital information conversion
- error correction
- circuit testing
- signal encoding
- data compression
- diagnosis of multiprocessors
- computational biology (e.g. [Glass'77,Zinovik+07])

Figure adopted from [Knuth'05]



Gene Regulatory Networks

Gene interaction using wiring diagrams:



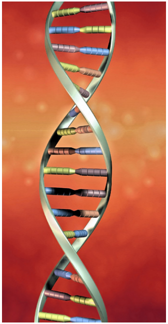
atopted from [Fisher+05]

Ordinary Differential Equations (ODE)

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



Glass Model

Gene activity is *on/off* only
The general form of a Glass network is

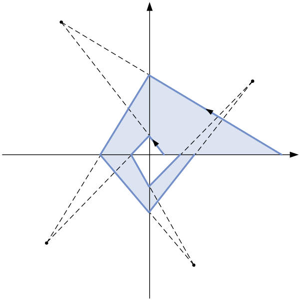
$$\dot{x}_i = G_i(\tilde{x}_1, \ldots, \tilde{x}_n) - \alpha x_i \quad \text{for } 1 \leq i \leq n$$

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 $a < b$

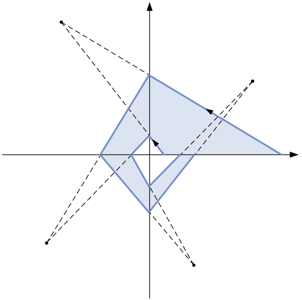
[Glass'77]

From phase flow to hypercubes

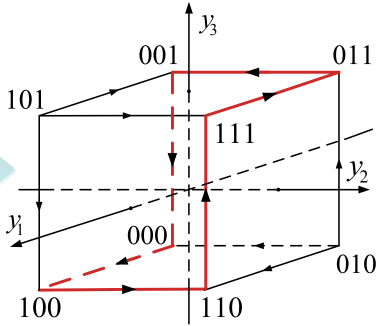


Phase Flow

From phase flow to hypercubes



Phase Flow



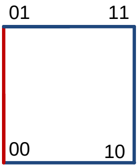
Transition Diagram

Hypercube

$$Q_1 = K_2, \quad Q_n = K_2 \times Q_{n-1}$$

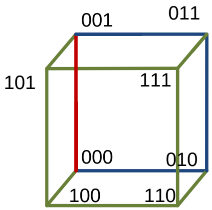
Hypercube

$$Q_1 = K_2, \quad Q_n = K_2 \times Q_{n-1}$$



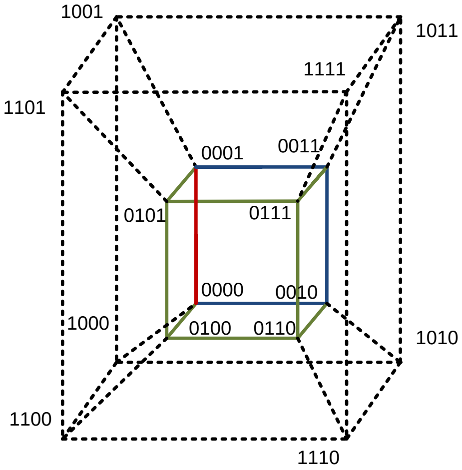
Hypercube

$$Q_1 = K_2, \quad Q_n = K_2 \times Q_{n-1}$$



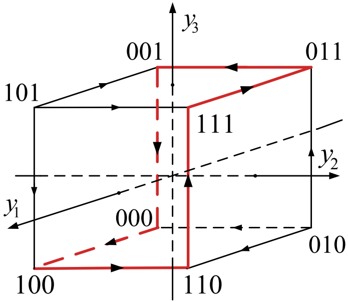
Hypercube

$$Q_1 = K_2, \quad Q_n = K_2 \times Q_{n-1}$$



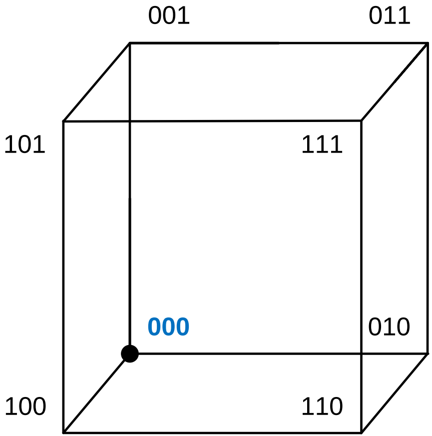
Induced cycle aka Cyclic snake

Periodic trajectories of Glass networks $\sim$ cycles in hypercubes
A *cyclic attractor* = induced cycle, edges oriented toward



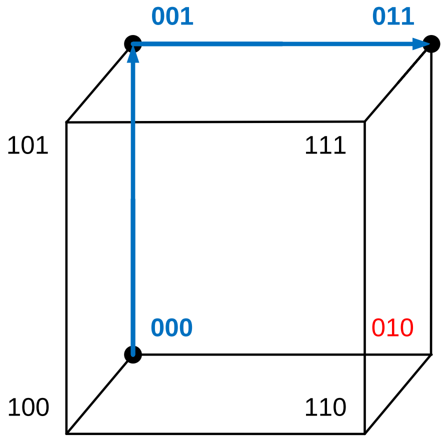
Induced cycle: Example

Initial node: 000,



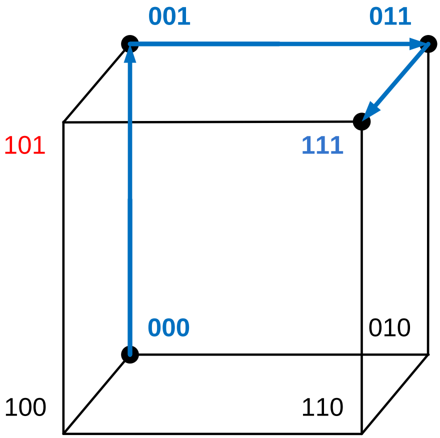
Induced cycle: Example

Initial node: 000, Transition sequence: 1, 2



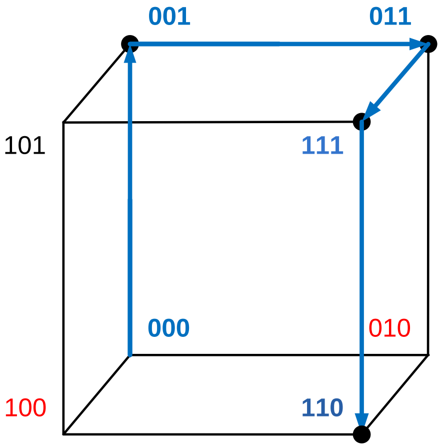
Induced cycle: Example

Initial node: 000, Transition sequence: 1, 2, 3



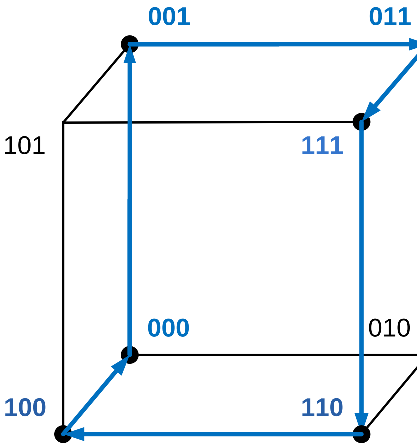
Induced cycle: Example

Initial node: 000, Transition sequence: 1, 2, 3, 1



Induced cycle: Example

Initial node: 000, Transition sequence: 1, 2, 3, 1, 2, 3



Hamming Distance, Paths on Hypercube

For two nodes W_k and W_l of n-cube Hamming distance

$$d_H(W_k, W_l) = \left|\{\ i \mid W_k[i] \neq W_l[i]\}\right|\ ,$$

Path $P = W_0, W_1, \ldots, W_{L-1}$ of length L,

$$\forall j \in \{0, 1, \ldots, L-2\}.\ \ d_H(W_j, W_{j+1}) = 1\ .$$

Cycle

$$\forall j \in \{0, 1, \ldots, L-1\}.\ \ d_H(W_j, W_{j+1 \bmod L}) = 1\ .$$

Cyclic distance

Definition

The *cyclic distance* $d_C(W_j, W_k)$ of two nodes W_j and W_k of a cycle of length L in an n-cube is

$$d_C(W_j, W_k) = \min\left\{|k-j|,\ L-|j-k|\right\}\ .$$

[Suparta06]

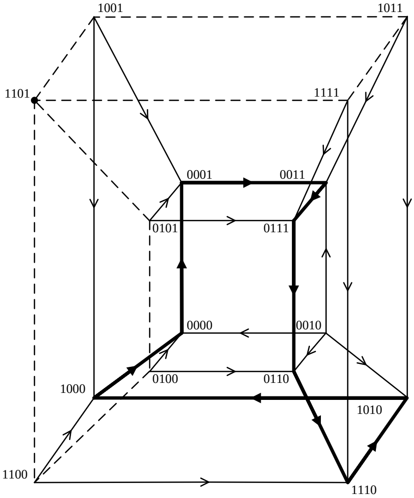
Induced cycle: Definition

Definition

An *induced cycle* $I_0, I_1, \ldots, I_{L-1}$ in an n-cube is a cycle such that any two nodes on the cycle are adjacent in the n-cube only if they are neighbours on the cycle:

$$\forall j, k \in \{0, 1, \ldots, L-1\}.$$
$$d_H(I_j, I_k) < 2 \implies d_C(I_j, I_k) < 2$$

Lean induced cycles: Example in 4D



Node 1101 is *shunned*

Lean induced cycles: shunned nodes

Definition

A node W is shunned if and only if it is not adjacent to any node of a cycle:

$$\forall i \in \{0, \ldots, L-1\}.\ d_H(W, I_i) > 1$$

Goal: maximize number of shunned nodes

Solution: use SAT solver

Propositional SAT: Advantages

- definite answer
- provably correct (satisfying assignment or proof)
- easy to modify encoding
- easy to add new constraints
- can benefit from theoretical results
- enumeration/classification (ALL-SAT)

Encoding induced cycles:

Input	dimension n, length L
Coordinates	$(n \cdot L)$ boolean variables $I_j[k]$, where $0 \leq j < L$ and $0 \leq k < n$
Transition seq	$(n \cdot L)$ XOR gates $xor^{k,k+1}[l]$
Cycle	$\bigwedge_{j=0}^{L-1} \quad d_H(I_j, I_{j+1 \text{ mod } L}) = 1$
Chordless	$\bigwedge_{j=0}^{L-3} \bigwedge_{k=j+2}^{L-1} \quad d_H(I_j, I_k) > 1$

How to encode d_H efficiently?

Encoding induced cycles: Once-twice predicates

- $once^{A,B}$ - at least one of $xor^{A,B}[i]$ is enabled
- $twice^{A,B}$ - at least two ...
- $d_H(A,B) = 1$ is encoded as

$$once^{A,B} \wedge \neg twice^{A,B}$$

- $d_H(A,B) > 1$ as

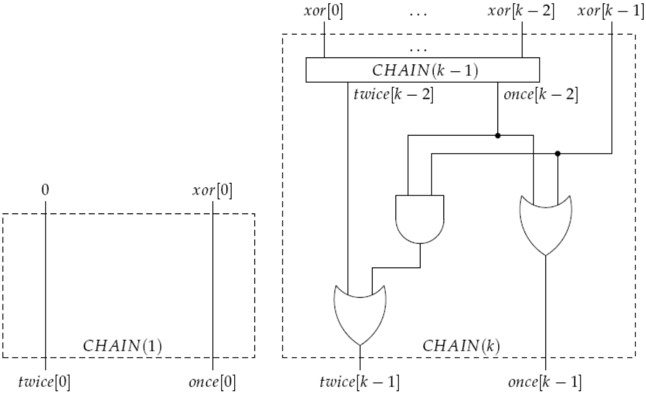
$$once^{A,B} \wedge twice^{A,B}$$

- and $d_H(A,B) \geq 1$ as

$$once^{A,B}$$

variety of encodings: OR-tree, long clause, etc.

Encoding induced cycles: Hamming distance



(a) base case (b) inductive definition

[Chebiryak&Kroening, JSAT'2008]

Encoding shunned nodes

Lean induced cycle $I_0, \ldots, I_{L-1}$ with at least S shunned nodes

Coordinates	S Boolean vectors $s_0, \ldots s_{S-1}$ of length n
Disjoint	$\bigwedge_{0 \leq i < j \leq S-1} \quad d_H(s_i, s_j) \geq 1$
Shunned	$\bigwedge_{i=0}^{S-1} \bigwedge_{j=0}^{L-1} \quad d_H(s_i, I_j) > 1$

Equivalence relation

Two cycles equivalent = transition sequences identical by

- permutation of axes
- reflection
- rotation

Example: 1, 2, 3, 1, 2, 3, $\sim$ 1, 3, 2, 1, 3, 2

$$\begin{aligned}
1,2,3,1,2,3 &\sim 2,3,1,2,3,1 && \text{(left rotation by 1)}\\
&\sim 1,3,2,1,3,2 && \text{(reflection)}\\
1,2,3,1,2,3 &\sim 1,3,2,1,3,2 && \pi(1,2,3) = (1,3,2)
\end{aligned}$$

Classification: ALL-SAT with blocking clauses for every equivalent cycle

Equivalence classes computation using SAT

Use ALL-SAT to compute $|IC(n, L, \geq U)|$, then

$$|IC(n, L, k)| = |IC(n, L, \geq k)| - |IC(n, L, \geq k+1)|$$

Scalability issues

- Too many blocking clauses per class:

$$(2L \cdot n!) = 25920 \quad \text{for } n = 6, L = 18$$

- Multiplied by number of classes (1228)
  = 30 millions blocking clauses.
- ALL-SAT is done when UNSAT is reached.

$\Rightarrow$ reduce number of blocking clauses per class

Symmetric Gray codes

Symmetric transition sequence:

$$t_1, t_2, \ldots, t_m, t_1, t_2, \ldots, t_m$$

$\Rightarrow$ rotations by $\frac{L}{2}$ positions and more are ineffective

Prefix Filtering

1. Fix first three transitions to $1, 2, 3$
   - $1, 2$ - w.l.o.g.
   - next transition can only be $3, 4, \ldots, n$
   - w.l.o.g. restrict to *canonical* one (i.e. dimension 3)
2. no need to add blocking clauses for equivalent cycles not starting with $1, 2, 3$
   (*trivially satisfied*)

Example: 1,2,3,1,2,3 $\sim$ 1, 3, 2, 1, 3, 2
blocking clause

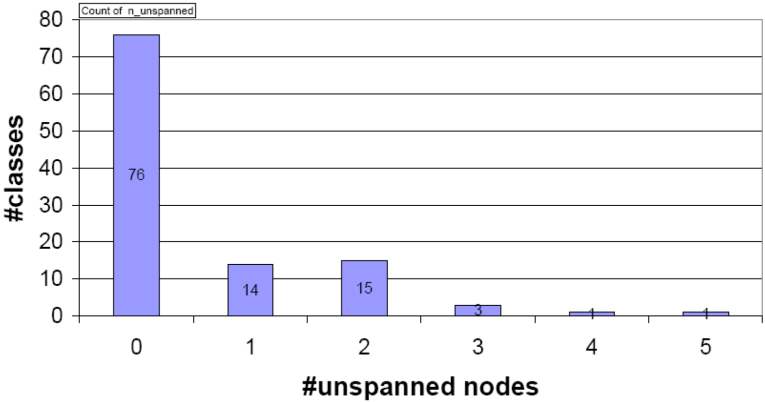
$$\neg xor^{0,1}[1] \vee \neg xor^{1,2}[3] \vee \neg xor^{2,3}[2] \vee \neg xor^{3,4}[1] \vee \neg xor^{4,5}[3] \vee \neg xor^{5,0}[2]$$

is satisfied: $xor^{0,1}[1] \wedge xor^{1,2}[2] \wedge xor^{2,3}[3]$

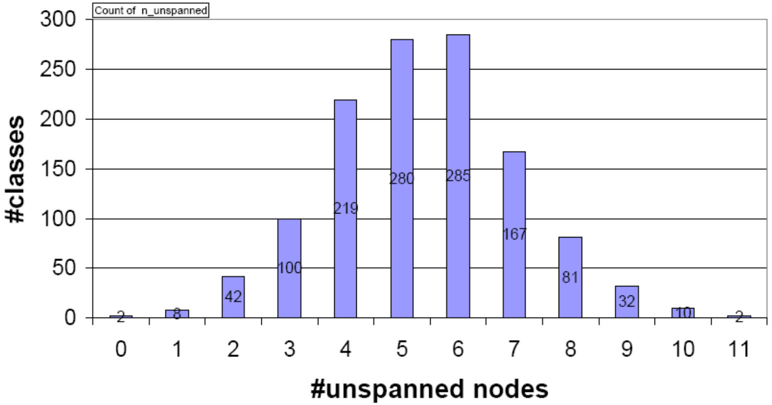
Longest lean induced cycles

n	L	U	Result	Time (sec)
3	6	0	SAT	<0.001
		1	UNSAT	<0.001
4	8	0	SAT	0.007
		1	SAT	0.009
		2	UNSAT	0.015
5	14	0	SAT	0.033
		1	UNSAT	3.012
6	26	0	SAT	0.420
		1	UNSAT	750.390
7	48	0	SAT	27568.000
		1	SAT	32175.000
		2	SAT	36936.000
		3	SAT	208304.000
		4	timeout	>60h

Classification: L=24



Classification: L=18



Conclusion

- New combinatorial problem with application in Systems Biology
- Solutions for dimensions up to 7
- Classification for dimensions up to 6

Challenges

- Higher dimensions
- Efficient algorithms
- Real world examples
- Complexity of the problem
- $n \longrightarrow \infty$

References

Blass+01 Blass, U.; Honkala, I.; Karpovsky, M. G. and Litsyn, S. *Short Dominating Paths and Cycles in the Binary Hypercube*, Annals of Combinatorics (5), pp. 51-59, 2001

Chebiryak+08 Chebiryak, Y. and Kroening, D. *Towards a classification of Hamiltonian cycles in the 6-cube*, Journal on Satisfiability, Boolean Modeling and Computation (4), pp. 57-74, 2008

Chebiryak+08 Chebiryak, Y. and Kroening, D. *An Efficient SAT Encoding of Circuit codes*, IEEE International Symposium on Information Theory and its Applications, Auckland, New Zealand, December, 2008

Fisher+05 Fisher, J.; Piterman, N.; Hubbard, E.; Stern, M. and Harel, D. *Computational insights into Caenorhabditis elegans vulval development*, Procs National Acad Sciences (102), pp. 1951-1956, 2005

Glass'77 Glass, L. *Combinatorial aspects of dynamics in biological systems*, Statistical mechanics and statistical methods in theory and applications, Plenum, pp. 585-611, 1977

Harary+88 Harary, F.; Hayes, J. P. and Wu, H. *A Survey of the Theory of Hypercube Graphs*. Comput. Math. Appl. (15), pp. 277-289, 1988

de Jong+08 de Jong, H. and Page, M. *Search for Steady States of Piecewise-Linear Differential Equation Models of Genetic Regulatory Networks*, IEEE/ACM Trans. Comput. Biology Bioinform. (5), pp. 208-222, 2008

Knuth05 Knuth, D. E. *The Art of Computer Programming, vol. 4, fascicle 2: Generating All Tuples and Permutations*, Addison-Wesley Professional, 2005

Suparta'06 Suparta, I. N. Counting sequences, *Gray codes and lexicodes*, Delft University of Technology, 2006

Zinovik+07 Zinovik, I.; Kroening, D. and Chebiryak, Y. *An Algebraic Algorithm for the Identification of Glass Networks with Periodic Orbits Along Cyclic Attractors*, Procs. Algebraic Biology, pp. 140-154, 2007

Zinovik+08 Zinovik, I.; Kroening, D and Chebiryak, Y. *Computing Binary Combinatorial Gray Codes via Exhaustive Search with SAT-solvers*, IEEE Transactions on Information Theory, 2008, vol. 54, pp. 1819-1823