Groebner Bases in Boolean Rings

for Model Checking and

Applications in Bioinformatics*

Quoc-Nam Tran, Ph.D.
Professor of Computer Science
Lamar University

*Invited Talk at CMU on October 8, 2010

Outline

- Groebner bases in a general setting
  - Applications
  - Complexity
- Groebner bases in Boolean rings
  - Naive approach
  - Problems
  - Our solutions
- Applications in bioinformatics
- Conclusion & new developments + open problems

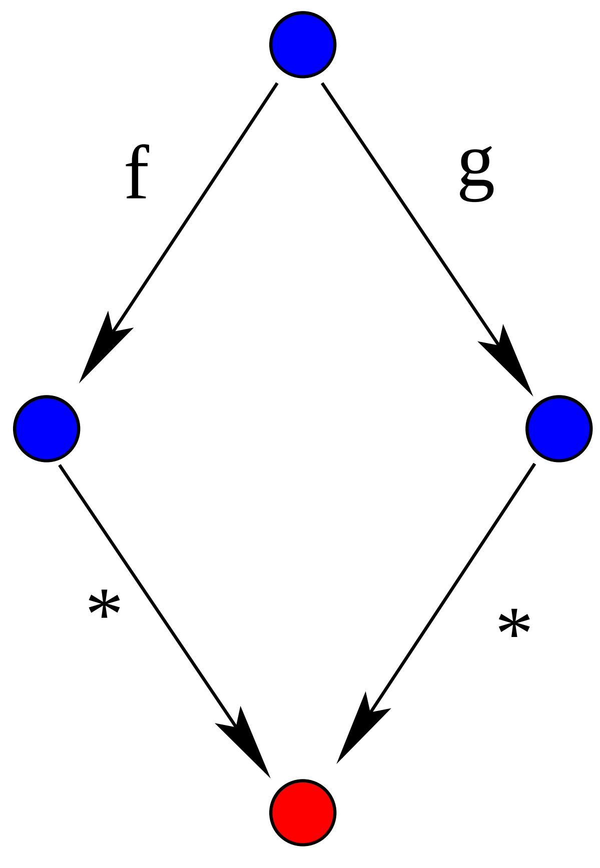
The Groebner bases method in a general setting

- Invented by B. Buchberger in 1965/1985 for polynomials over a computational field.

- Church-Rosser Term Rewriting System (G. Huet's procedure)

$$F = \begin{cases} \underline{2x^3} - 16x^2 + 48x - 44 + xyz - 2xz - 5xy - 2yz + 4z + 10y, \\ \underline{x^3} - 8x^2 + 10x + 33 + yz^2x - 4xyz + 3xy - 3xz^2 + 12xz - \\ \quad 5yz^2 + 20yz - 15y + 15z^2 - 60z, \\ \underline{yz^2x} - xyz - 12xy - 3xz^2 + 6xz + 21x - 5yz^2 + 14yz + \\ \quad 15y + 15z^2 - 48z - 15 \end{cases}$$

lcm(lpp(f),lpp(g))



A Gröbner basis of this system w.r.t. an elimination term order (lexicographic term order where $x \succ y \succ z$)

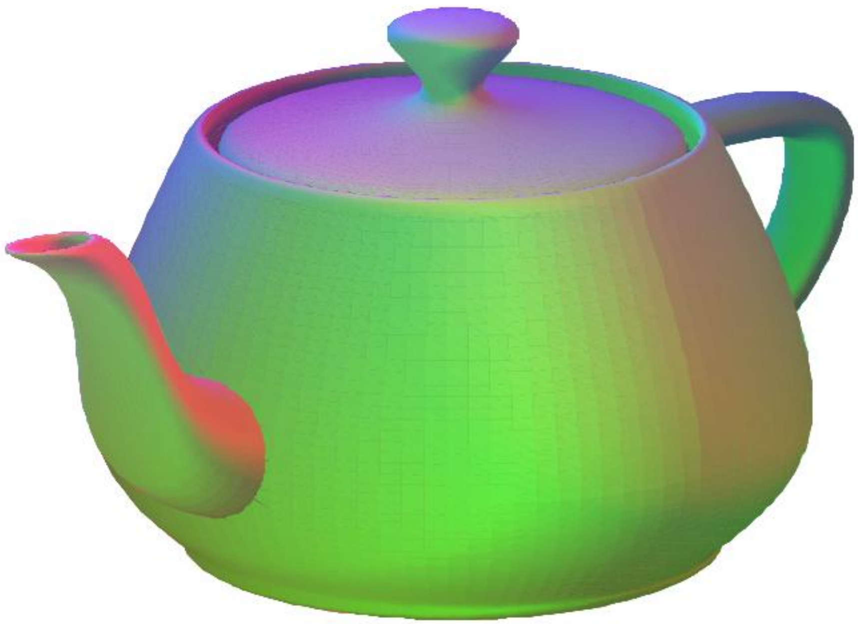
$$G = \begin{cases} z^3 - 9z^2 + 23z - 15, \\ -z^2 + 8y + 4z - 19, \\ x^3 - 8x^2 + 19x - 12, \end{cases}$$

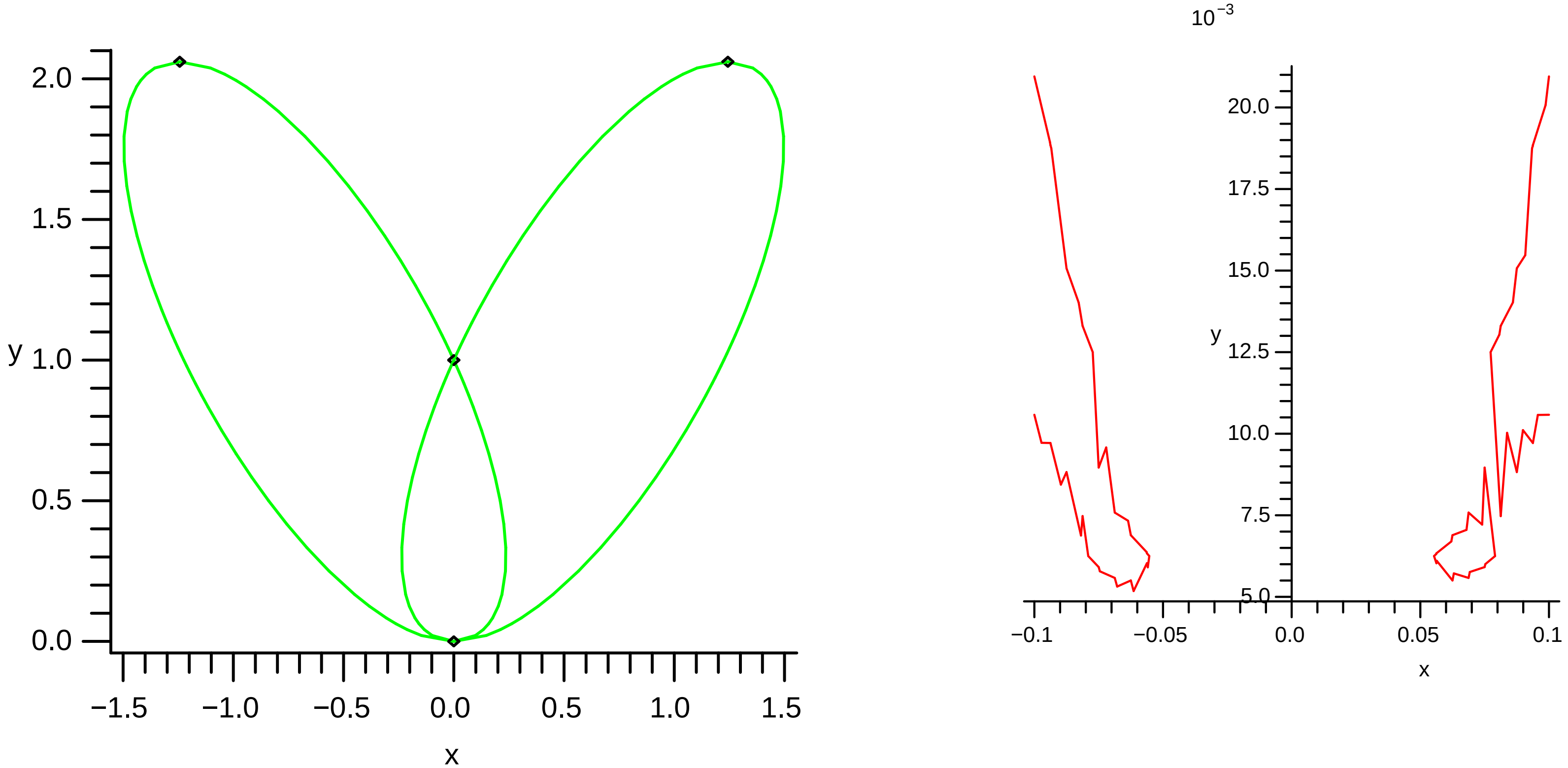
$$V(F) = V(G), \langle F \rangle = \langle G \rangle, \langle in_{\succ}(\langle G \rangle) \rangle = \langle in_{\succ}(G) \rangle$$

Some Applications of Groebner Bases Computation

- Solving systems of non-linear equations
- Computer Aided Geometric Design & Solid Modeling
- Automated Theorem Proving
- Applied Mathematics
- Automated Verification of Hardware & Software (Model Checking)

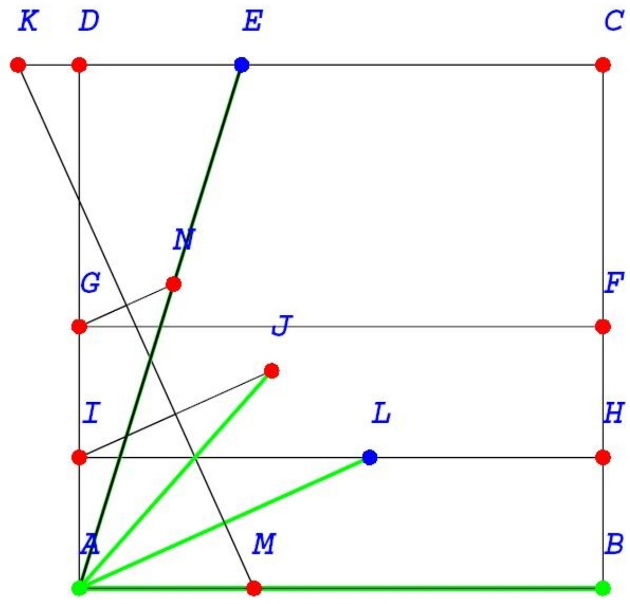
Computer Aided Geometric Design





2.0
1.5
y
1.0
0.5
0.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
x
10−3
20.0
17.5
15.0
y
12.5
10.0
7.5
5.0
−0.1
−0.05
0.0
0.05
0.1
x

Trisecting an Angle by Hand



$t \leftarrow 0$;

```
process p0                              process p1
 s0 ←nc;                                 s1 ←nc;
 while 1                                 while 1
  t' ←(t = 0∧s0 =c?¬t: t)                 t' ←(t = 1∧s1 =c?¬t: t)
  s0' ←(case                              s1' ←(case
     s0=nc: {r, nc},                         s1=nc: {r, nc},
     s0=r ∧ s1=nc: c,                        s1=r ∧ s0=nc: c,
     s0=r∧s1=r∧t = 0: c,                     s1=r∧s0=r∧t = 0: c,
     s0=c: {c, nc}                           s1=c: {c, nc}
     default: s0);                           default: s1);
  t ← t';                                 t ← t';
  s0 ← s0';                               s1 ← s1';
```

- Use one variable x_1 for t, two variables x_2, x_3 for s_0, two variables x_4, x_5 for s_1, and one variable x_6 for keeping track of the running process. We encode the enumerated variables s_0 and s_1 by setting the corresponding pair of bits to $(0,0)$ for nc, $(0,1)$ for r, and $(1,0)$ for c.

- The transition relation T can be constructed based on the assignments made by processes p_0 and p_1.

- The property we want to check is $\mathbf{EF}f$, where $f \equiv (s_0 = \text{c}) \wedge (s_1 = \text{c})$.

- The temporal formula can be translated into the least fixed point of $\mu y.f \vee \mathbf{EX}y$ where f can be represented by $I_f = \langle x_2(x_3+1)+1, x_4(x_5+1)+1\rangle$.

- Groebner basis computation found the least fixed point of $\lambda y.f \vee \mathbf{EX}y$ as $I_{\mathbf{EF}f} = \langle x_2+1, x_3, x_4+1, x_5\rangle$. The initial condition can be represented by $I_{init} = \langle x_2, x_3, x_4, x_5, x_6\rangle$.

- Another simple Groebner basis computation for $V(I_{\mathbf{EF}f}) \cap V(I_{init})$ show that the constant polynomial 1 is the Groebner basis.

- That means $\mathbf{EF}f$ is false in the initial states.

Complexity

- Exponential Space

- P$\subseteq$NP$\subseteq$P-SPACE=NP-SPACE$\subseteq$EXP-TIME$\subseteq$EXP-SPACE

Focus

- Improving the practical efficiency.

- Better complexity for specific domains.

- Hybrid symbolic-numerical methods.

The Groebner bases method in Boolen rings

A ring $\mathbf{R} = \langle R, +, \cdot, 0, 1 \rangle$ is Boolean if $\mathbf{R}$ satisfies $x^2 \approx x, \forall x \in R$.

If $\mathbf{R}$ is a Boolean ring, then $\mathbf{R}$ is commutative and $x + x \approx 0$.

Boolean algebra $(R, \wedge, \vee)$ gives rise to a ring $(R, +, \cdot)$ and vice versa

$a + b = (a \wedge \neg b) \vee (b \wedge \neg a)$ and $a \cdot b = a \wedge b$.

$x \vee y = x + y + x \cdot y$, $x \wedge y = x \cdot y$ and $\neg x = x + 1$.

Naive Approach

- $F \subset \mathbf{R}[X]$

- $F' = F \cup \{x_1^2 + x_1, x_2^2 + x_2, \ldots x_n^2 + x_n\}$

Problems

- Theoretical point of view: EXP-SPACE

- Practical point of view: Blow-up in degree and number of terms

- Parallelism: very hard to parallelize Buchberger's algorithm

Our Solutions

$$p - \mathsf{nf}(p) = \sum_{i=1}^{s} f_i \cdot h_i$$

$$\begin{aligned} p &= \textstyle\sum_{x\in[X],\mathsf{deg}(x)\leq n} r_x \cdot x + \\ &\quad \textstyle\sum_{i=1}^{s} (\sum_{x\in[X],\mathsf{deg}(x)\leq n} f_{i,x} \cdot x) \cdot \\ &\quad (\textstyle\sum_{x\in[X],\mathsf{deg}(x)\leq n} h_{i,x} \cdot x) \\ &= \textstyle\sum_{x\in[X],\mathsf{deg}(x)\leq n} (r_x + \\ &\quad \textstyle\sum_{i=1}^{s} \sum_{u,v\in[X],u\cdot v=x} f_{i,u} \cdot h_{i,v}) \cdot x \\ &= M.b \end{aligned} \tag{1}$$

Given a set of polynomials F, a term order $\prec$ and a polynomial p.

Find the normal form nf(p) of p with respect to $I = \langle F \rangle$ and $\prec$.

Step_1 M and b on fly.

Step_2 Find a full row rank sub-matrix

Step_3 Find a full column rank sub-matrix
Add corresponding elements of vector b into vector b'

Return the solution of $p = M'.b'$

Remark:

Let s be the number of polynomials in F and S be the biggest number of monomials in all polynomials of F. Finding the value of any element in M requires $O(s \cdot S \cdot n)$ memory space.

$F = \{(x_1 + 1).(x_2 + 1) \ldots (x_n + 1), x_1x_2 + x_3\}$??

Given a set of polynomials F and a term order $\prec$.

Find the reduced Groebner basis of $I = \langle F \rangle$ with respect to $\prec$.

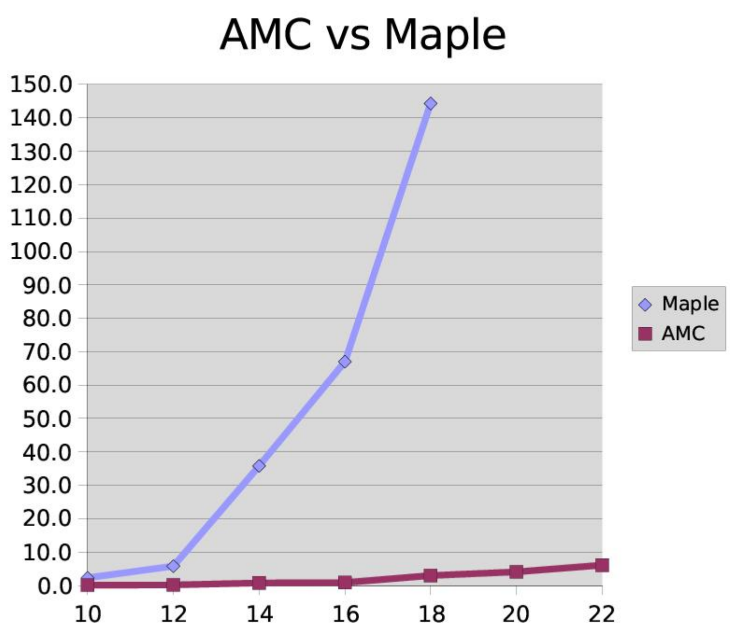
Step_1 Set $G' = \emptyset$; Matrix M and vector b on the fly

Step_2 For all monomial m, $1 \not\prec m \prec x_1 \cdot x_2 \cdots x_n$ do
If $1 = m + \mathrm{nf}(m)$ then stop and return $\{1\}$;
Add $m + \mathrm{nf}(m)$ into G' when m is minimal reducible.

Step_3 return G'.

- Theoretical point of view: P-SPACE

- Practical point of view:
  - No blow-up in degree

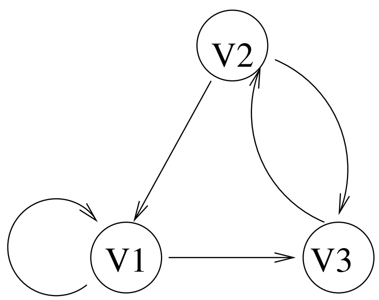


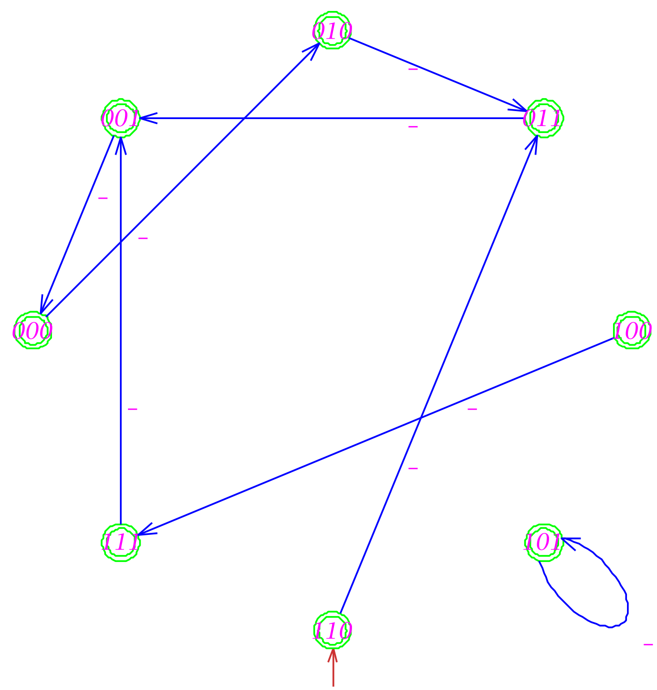
- Parallelism: multi-core GPUs

Boolean Networks

One of the extensively studied topics for BN is to identify the attractors, the directed cycles in the state transition diagram.

time t			time t+1		
v_1	v_2	v_3	v_1	v_2	v_3
0	0	0	0	1	0
0	0	1	0	0	0
0	1	0	0	1	1
0	1	1	0	0	1
1	0	0	1	1	1
1	0	1	1	0	1
1	1	0	0	1	1
1	1	1	0	0	1



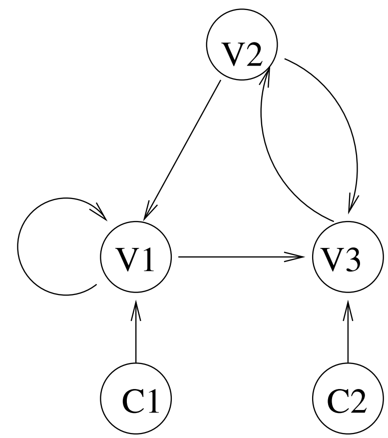


Boolean network with $v_1(t+1) = v_1(t) \wedge \neg v_2(t)$, $v_2(t+1) = \neg v_3(t)$, $v_3(t+1) = v_1(t) \vee v_2(t)$

Finding a singleton attractor is NP-hard. Can be easily translated into an LTL model checking problem. Fore example, to find an attractor of length 4 of the BN in Figure 1, one can use the following LTL formula: LTLSPEC !F(((X X X X(v1) <-> v1) & (X X X X(v2) <-> v2) & (X X X X(v3) <-> v3)) & !((X(v1) <-> v1) & (X(v2) <-> v2) & (X(v3) <-> v3))).

Finding control strategies for a network

Internal			Control	
v_1	v_2	v_3	c_1	c_2
1	1	0	0	0
0	1	1	0	1
0	0	0	0	0
0	1	0		



$v_1(t+1) = v_1(t) \wedge \neg v_2(t) \wedge c_1,\ v_2(t+1) = \neg v_3(t),\ v_3(t+1) = (v_1(t) \vee v_2(t)) \wedge \neg c_2.$

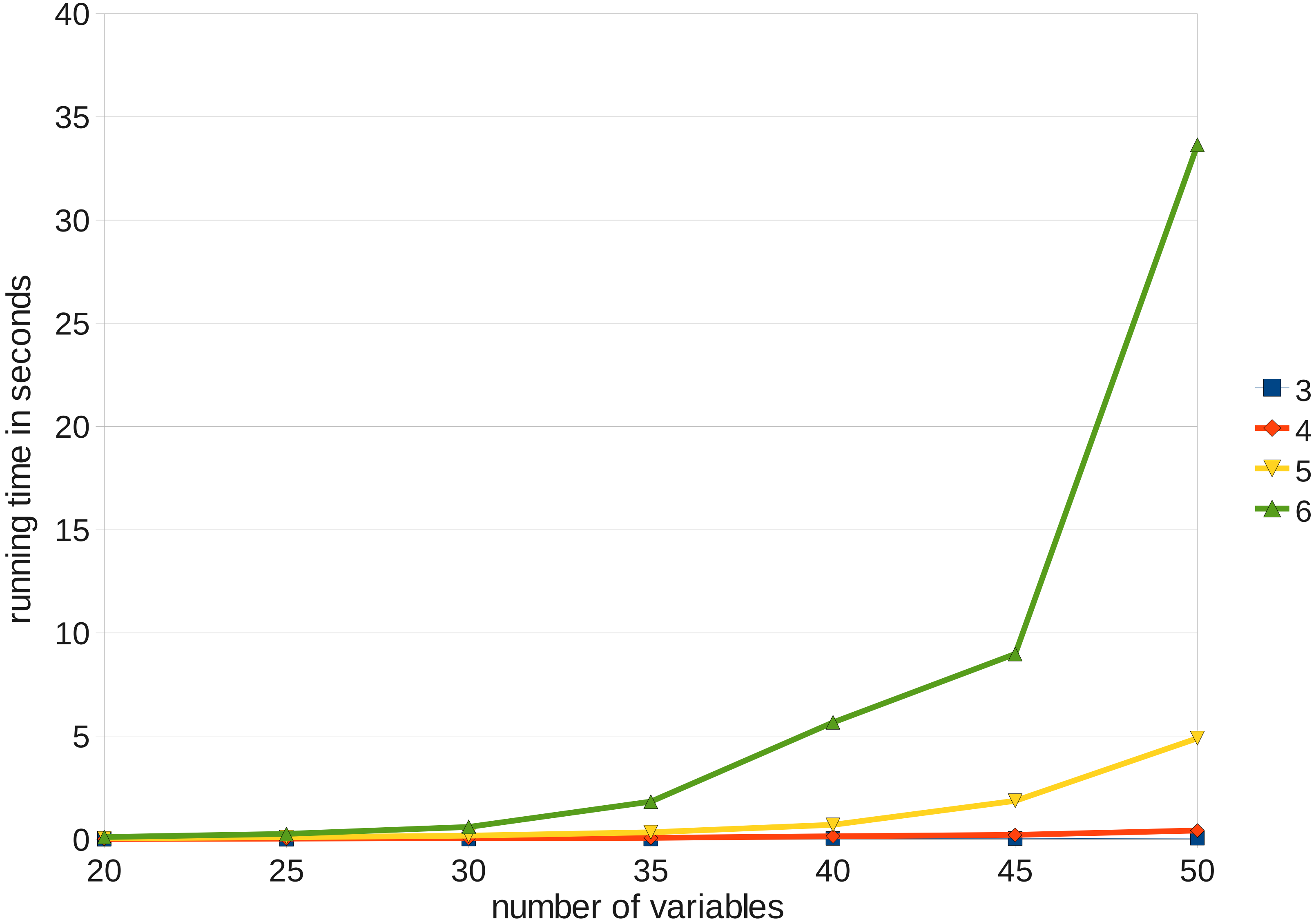
To find a control strategy for this BN with the initial state of $[1,1,0]$ and desired state of $[0,1,0]$, one can use the following LTL formula: LTLSPEC !F((v1 <-> 0) & (v2 <-> 1) & (v3 <-> 0)).

We generated 3,600 random networks with 25, 30, 35, 40, 45 and 50 internal nodes; in-degree of 3, 4, 5, and 6; cycle length of 1, 2, 3, 4 and 5.

For experimenting with the problem of finding the control strategies for BN, we generated 2,160 random networks with 25, 30, 35, 40, 45 and 50 internal nodes; in-degree of 3, 4, 5, 6, 7 and 8; and 5, 8 and 10 control nodes.

When the symbolic model checking using BDD approach is used, for almost all of the problems, the BDDs were blown up very fast and the system crashed very soon, especially for BNs with more than 30 nodes and in-degree of more than 3.

When bounded model checking is used with zChaff SAT solver, for almost all of the problems, memory use was reasonable but NuSMV failed to find a counter-example after 10,000 seconds with a bound of at most 30. Notice that in real-world BNs, one may have thousands of nodes.


40
35
30
25
20
15
10
5
0
running time in seconds
20
25
30
35
40
45
50
number of variables
3
4
5
6

Conclusion

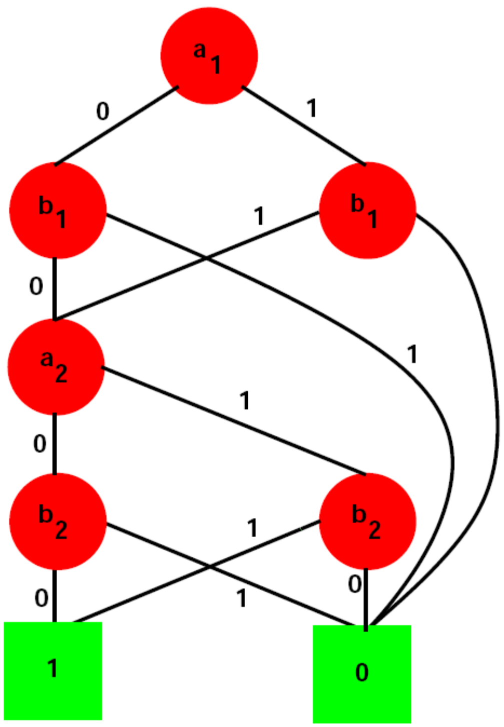
- Lowest bound for Groebner bases computation P-SPACE vs EXP-SPACE

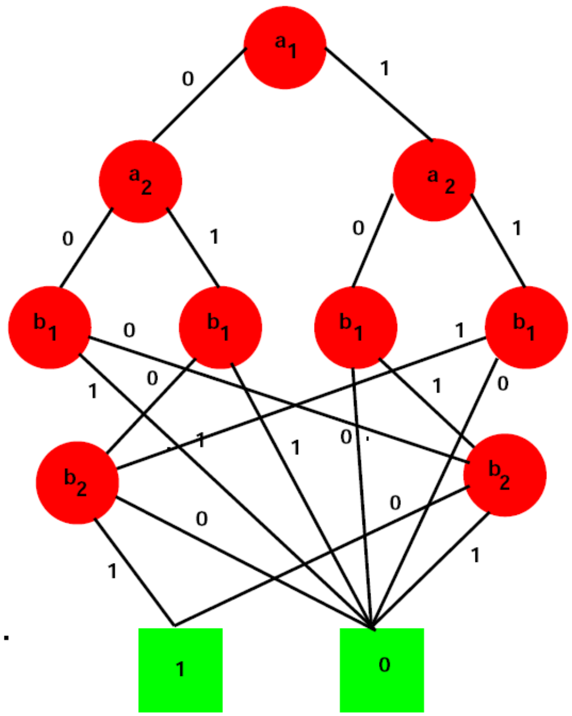
- Same complexity for generalized Boolean rings, e.g. $F_4[X]$

- Parallelism

- Bases conversions over generalized Boolean rings
  - Conjecture: P

$$a_1 \Leftrightarrow b_1 \wedge a_2 \Leftrightarrow b_2$$





- $I = \langle\{f_1, f_2, \ldots, f_k\}\rangle \lhd K[x_1, \ldots, x_n, x_1^{'}, \ldots, x_n^{'}]$, we need $I \cap K[x_1^{'}, \ldots, x_n^{'}]$. If G_I is a Groebner basis of I w.r.t. an elimination term order, where $x_1 \succ \ldots \succ x_n \succ x_1^{'} \succ \ldots \succ x_n^{'}$. Return $G \cap K[x_1^{'}, \ldots, x_n^{'}]$.

THANKS!

- Questions or suggestions?