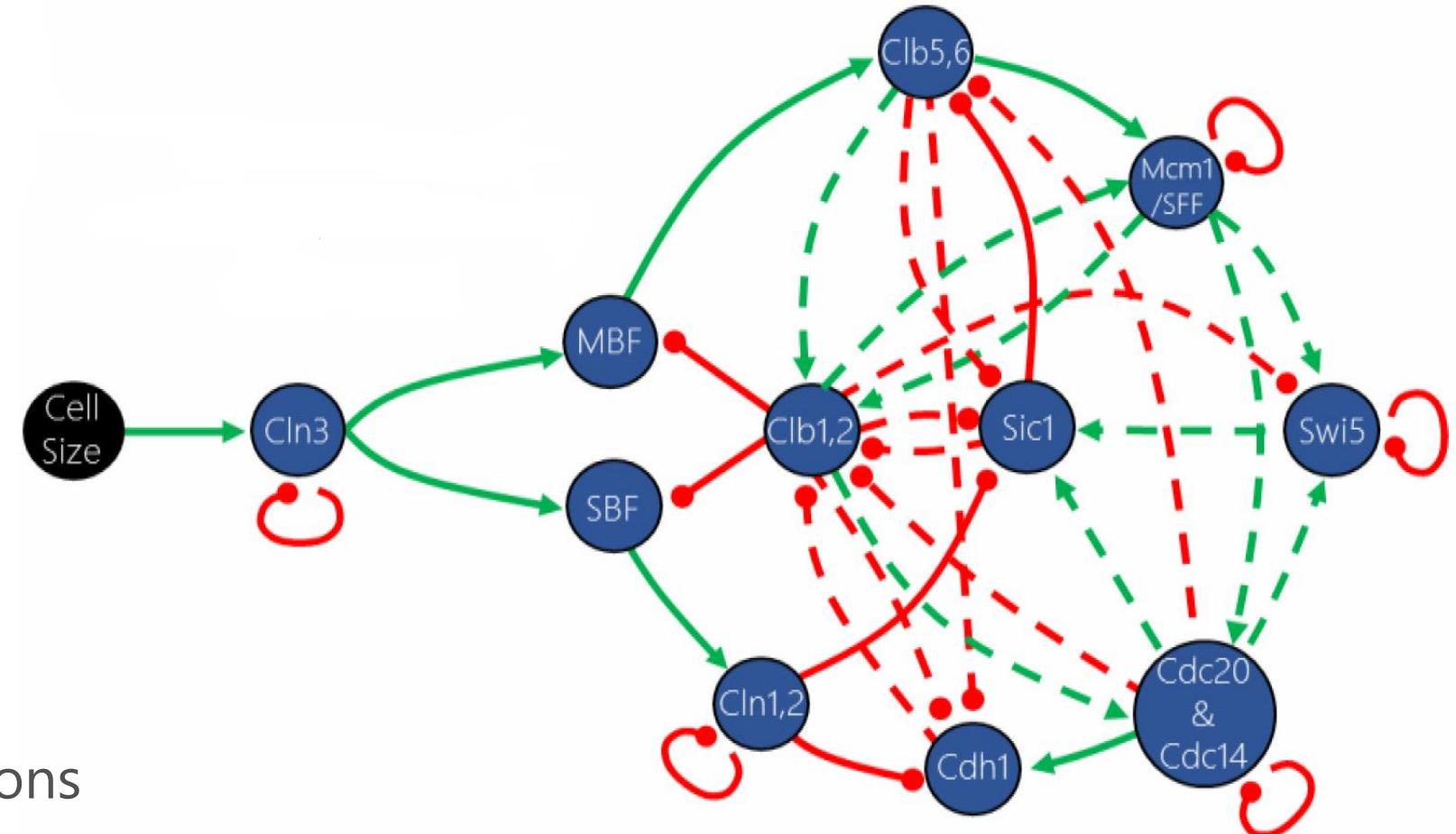


Algebraic Polynomial-based Synthesis for Abstract Boolean Network Analysis

Peter Backeman, Christoph M. Wintersteiger,
Boyan Yordanov, and Sara-Jane Dunn

Abstract Boolean Networks

- Genes
- Interactions
 - Positive, negative
 - Known
 - Suspected
- Observations
- ABN
 - Reproduces observations
 - (And other constraints)



ABN Regulation Conditions

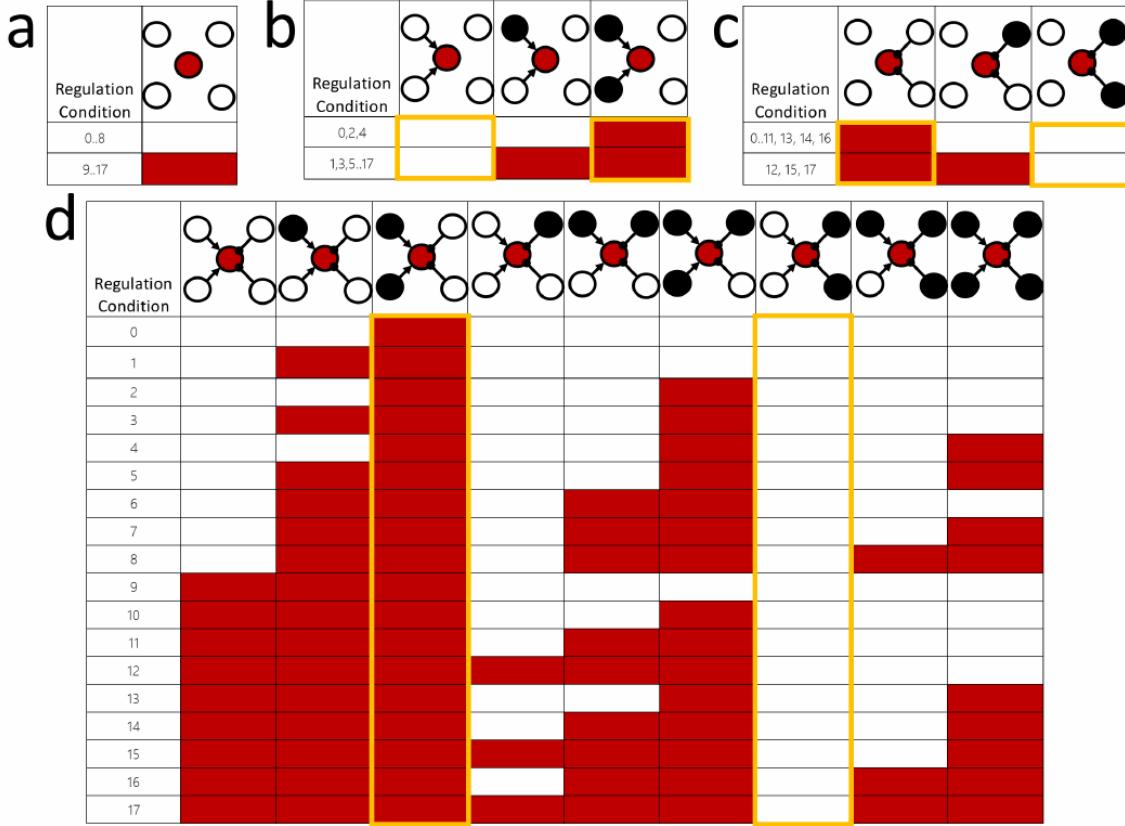


Figure 2: **The set of 18 regulation conditions (excluding the two threshold rules).**

we define the following regulation condition templates:

$$\begin{aligned}
 R'_0(c, q) &\triangleq \text{AllActivators}(c, q) \wedge \text{NoRepressors}(c, q) \\
 R'_1(c, q) &\triangleq \neg \text{NoActivators}(c, q) \wedge \text{NoRepressors}(c, q) \\
 R'_2(c, q) &\triangleq \text{AllActivators}(c, q) \wedge \neg \text{AllRepressors}(c, q) \\
 R'_3(c, q) &\triangleq (\text{NoRepressors}(c, q) \wedge \neg \text{NoActivators}(c, q)) \vee \\
 &\quad (\neg \text{AllRepressors}(c, q) \wedge \text{AllActivators}(c, q)) \\
 R'_4(c, q) &\triangleq \text{AllActivators}(c, q) \\
 R'_5(c, q) &\triangleq \text{AllActivators}(c, q) \vee (\text{NoRepressors}(c, q) \wedge \neg \text{NoActivators}(c, q)) \\
 R'_6(c, q) &\triangleq \neg \text{NoActivators}(c, q) \wedge \neg \text{AllRepressors}(c, q) \\
 R'_7(c, q) &\triangleq (\neg \text{NoActivators}(c, q) \wedge \neg \text{AllRepressors}(c, q)) \vee \text{AllActivators}(c, q) \\
 R'_8(c, q) &\triangleq \neg \text{NoActivators}(c, q) \\
 R'_9(c, q) &\triangleq \text{NoRepressors}(c, q) \\
 R'_{10}(c, q) &\triangleq \text{NoRepressors}(c, q) \vee (\neg \text{AllRepressors}(c, q) \wedge \text{AllActivators}(c, q)) \\
 R'_{11}(c, q) &\triangleq \text{NoRepressors}(c, q) \vee (\neg \text{NoActivators}(c, q) \wedge \neg \text{AllRepressors}(c, q)) \\
 R'_{12}(c, q) &\triangleq \neg \text{AllRepressors}(c, q) \\
 R'_{13}(c, q) &\triangleq \text{NoRepressors}(c, q) \vee \text{AllActivators}(c, q) \\
 R'_{14}(c, q) &\triangleq (\text{NoRepressors}(c, q) \vee \text{AllActivators}(c, q)) \vee \\
 &\quad (\neg \text{AllRepressors}(c, q) \wedge \neg \text{NoActivators}(c, q)) \\
 R'_{15}(c, q) &\triangleq \neg \text{AllRepressors}(c, q) \vee \text{AllActivators}(c, q) \\
 R'_{16}(c, q) &\triangleq \text{NoRepressors}(c, q) \vee \neg \text{NoActivators}(c, q) \\
 R'_{17}(c, q) &\triangleq \neg \text{AllRepressors}(c, q) \vee \neg \text{NoActivators}(c, q)
 \end{aligned}$$

ABN Regulation Conditions



$$R'_0(c, q) \triangleq \text{AllActivators}(c, q) \wedge \text{NoRepressors}(c, q)$$

$$R'_1(c, q) \triangleq \neg \text{NoActivators}(c, q) \wedge \text{NoRepressors}(c, q)$$

$$R'_2(c, q) \triangleq \neg \text{AllActivators}(c, q) \wedge \neg \text{NoRepressors}(c, q)$$

Observations

```
9 // Experiment Two from 2i to 2i plus LIF
10 #ExperimentTwo[0] |= $Twoi "Exp2 initial expression pattern";
11 #ExperimentTwo[0] |= $TwoiPlusLifCultureConditions "Exp2 culture conditions";
12 #ExperimentTwo[0] |= $NoKnockDowns "Exp2 no knockdowns";
13 #ExperimentTwo[0] |= $NoOverExpression "Exp2 no overexpression";
14 #ExperimentTwo[18] |= $TwoiPlusLif "Exp2 penultimate state";
15 #ExperimentTwo[19] |= $TwoiPlusLif "Exp2 final state";
```

```
196 $TwoiPlusLifCultureConditions :=  
197 {  
198   LIF = 1 and  
199   CH = 1 and  
200   PD = 1  
201 };
```

```
315 $TwoiPlusLif:=  
316 {  
317 MEKERK = 0 and  
318 Oct4=1 and  
319 Sox2=1 and  
320 Nanog=1 and  
321 Esrrb=1 and  
322 Klf2=1 and  
323 Tfcpl11=1 and
```

ABN Encoding

- Components
 - (Boolean) genes G
 - (Constrained) interactions I
 - (Abstract) regulation conditions R
 - Initial and final valuation of all genes
- Implicit
 - States S
 - Transition relation $T(s, s')$
 - Property p
 - T unwound for fixed number of steps (here 20)
 - Synchronous

$$\phi(I, R)$$

Problems

- Find good interaction graphs

$$\exists I \exists R . \phi(I, R)$$

- Find bad interaction graphs

$$\exists I \forall R . \neg \phi(I, R)$$

- Minimal/essential interaction graphs

$$\exists I, R. \phi(I, R) \Rightarrow \neg (\exists I', R'. \phi(I', R') \wedge (\#I' < \#I))$$

- Find a “simple” f such that

$$\forall I . f(I) = (\exists R . \phi(I, R))$$

State of the art (Bio)

- RE:IN
- Check

$$\exists I \exists R . \phi(I, R)$$

- Record a list of (I, R)

$$f(I, R) = ITE(1, 2, \\ITE(2, 3,\\4))$$

Interaction	1	2	3	4	5	6	7	8	9	10	11	12
* Cdc20-->Cdh1	green											
Cdc20-- Clb12	red	red								red	red	
Cdc20-->Swi5	green											
Clb12-->Cdc20	green		green									
* Clb12-- MBF	red											
* Clb12-- SBF	red											
Clb56-->Clb12	green											
* Clb56-->Mcm1	green											
* Cln12-- Cdh1	red											
* Cln12-- Sic1	red											
* Cln3-->MBF	green											
* Cln3-->SBF	green											
* MBF-->Clb56	green											
* SBF-->Cln12	green											
* Sic1-- Clb56	red											
Swi5-->Sic1	green		green									
Cdc20-->Sic1	green											
Mcm1-->Cdc20	green											
Cdh1-- Clb12		red										
Sic1-- Clb12								red	red	red	red	red

SMT Quantifiers

- Z3 comes with
 - *E-matching* for unsatisfiable problems
 - *Model based quantifier instantiation* (MBQI) for satisfiable problems
- MBQI
 - Skolemize
$$\forall u \exists e . \phi(u, e) \rightarrow \forall u \exists e . \phi(u, e f_e(u))$$
 - Guess a model, e.g.,
$$\{ f_e(x) = \neg x \}$$
 - Check for counter-examples u s.t.
$$\neg \forall u . \phi(u, \neg u) = \exists u . \neg \phi(u, \neg u)$$
 - Remember u

MBQI: fixing the model

- The model f_e is broken at $\textcolor{orange}{u}$

$$\neg \phi(\textcolor{orange}{u}, f_e(\textcolor{orange}{u}))$$

- Find out what would have worked

$$\exists e . \phi(\textcolor{orange}{u}, \textcolor{teal}{e})$$

- Fix the broken model

$$f'_e(x) = ITE(\textcolor{orange}{u}, \textcolor{teal}{e}, f_e(x))$$

State of the art (SMT)

- MBQI
 - Check
 - $\exists u. \neg \phi(u, m)$
 - $\exists e. \phi(u, e)$
 - Record a list of counter-examples & fixes (u, e)
 - $f(\dots) = ITE(1, 2,$
 $ITE(2, 3,$
 $4)$

Heatmap showing interactions between various proteins across 12 conditions. The y-axis lists interactions, and the x-axis shows conditions 1 through 12. Green indicates positive interaction, red indicates negative interaction.

Interaction	1	2	3	4	5	6	7	8	9	10	11	12
Cdc20-->Cdh1	Green											
Cdc20-- Clb12	Red	Red							Red	Red		
Cdc20-->Swi5	Green											
Clb12-->Cdc20	Green			Green	Green	Green	Green			Green	Green	Green
* Clb12-- MBF	Red											
* Clb12-- SBF	Red											
Clb56-->Clb12	Green											
* Clb56-->Mcm1	Green											
* Cln12-- Cdh1	Red											
* Cln12-- Sic1	Red											
* Cln3-->MBF	Green											
* Cln3-->SBF	Green											
* MBF-->Clb56	Green											
* SBF-->Cln12	Green											
* Sic1-- Clb56	Red											
Swi5-->Sic1	Green		Green	Green					Green		Green	Green
Cdc20-->Sic1		Green			Green							
Mcm1-->Cdc20		Green	Green			Green	Green	Green	Green	Green		Green
Cdh1-- Clb12			Red	Red	Red	Red						
Sic1-- Clb12								Red	Red			

Skolem Function Templates

- Skolemize into

$$\forall u \exists e . \phi(u, e f_e(u))$$

- But also enforce

$$\exists a_0, a_1 \forall x . f_e(x) = a_0 \oplus a_1 x$$

- Now we have

$$\exists a_0, a_1 \forall u . \phi(u, a_0 \oplus a_1 u)$$

Algebraic Normal Form

- Every function over n Booleans
 - Has equivalent ANF

$$\begin{aligned} f(x_1, \dots, x_n) = & a_0 \oplus \\ & a_1 x_1 \oplus a_2 x_2 \oplus \dots \oplus \\ & a_{1,2} x_1 x_2 \oplus \dots \oplus \\ & \dots \oplus \\ & a_{1,2,\dots,n} x_1 \dots x_n \end{aligned}$$

- Such that the coefficients $a_i \in \{0,1\}$ fully describe f .
- E.g., $x_1 \wedge x_2 = 1 \oplus x_1 \oplus x_2 \oplus x_1 x_2$

Instantiation Templates

- When looking for a counter-example

$$\neg \forall u . \phi(u, f_e(u))$$

- Constrain u to a template too

$$\neg \forall u . c = f_u(u) \Rightarrow \phi(c, f_e(u))$$

- Now enforce a template on f_u

$$\exists a_0, a_1, \dots$$

$$\exists e_0, e_1, \dots$$

$$\neg \forall u . \phi(a_0 \oplus a_1 u_1 \dots, e_0 \oplus e_1 u_1 \dots)$$

Templated Branches

- Instead of lists of counter-examples

$$\begin{aligned} f(\dots) = & \textit{ITE}(a_0 \oplus a_1 x_1 \dots, e_0 \oplus e_1 x_1 \dots, \\ & \textit{ITE}(a'_0 \oplus a'_1 x_1 \dots, e'_0 \oplus e'_1 x_1 \dots, \\ & \quad \quad \quad \cdots \\ & \quad \quad \quad 0) \end{aligned}$$

- E.g., non-linear functions become piecewise linear

Template Refinement

- When there are no solutions
- Increase template complexity

$$\begin{aligned} f(x_1, \dots, x_n) = & a_0 \oplus \\ & a_1 \textcolor{teal}{x}_1 \oplus a_2 \textcolor{teal}{x}_2 \oplus \dots \oplus \\ & a_{1,2} \textcolor{orange}{x}_1 x_2 \oplus \dots \oplus \\ & \dots \oplus \\ & a_{1,2,\dots,n} \textcolor{magenta}{x}_1 \dots x_n \end{aligned}$$

Scale Example

- 16 genes, 8 optional interactions, 2 experiments
- RE:IN enumerated 96 solutions
- $((\text{Nanog} \rightarrow \text{Sox2}) \text{ AND } (\text{Klf2} \rightarrow \text{Oct4})) \text{ OR }$
 $((\text{Sall4} \rightarrow \text{Sox2}) \text{ AND } (\text{Klf2} \rightarrow \text{Oct4}))$

Future work

- Smarter template refinement
- Alternatives for ANF polynomials
 - New (relatively complete) classes of parametrizable functions
 - E.g., graphs & properties
- Extensions to other theories
 - Esp. integers, bit-vectors, arrays
- Lots of applications
 - Complexity management
 - Tailored function templates



© 2013 Microsoft Corporation. All rights reserved. Microsoft, Windows and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries.
The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.