Microsoft Research

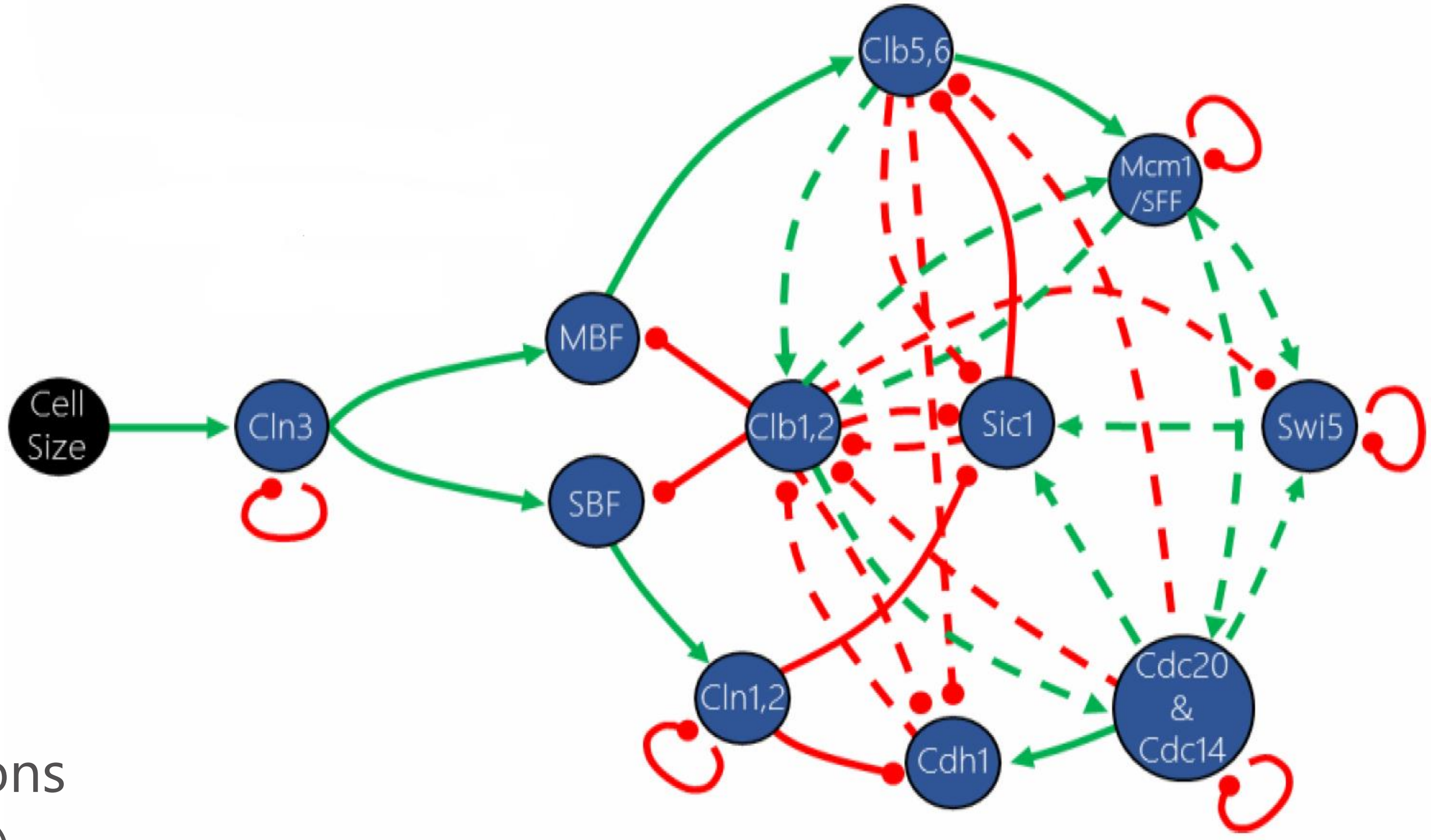
Microsoft

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

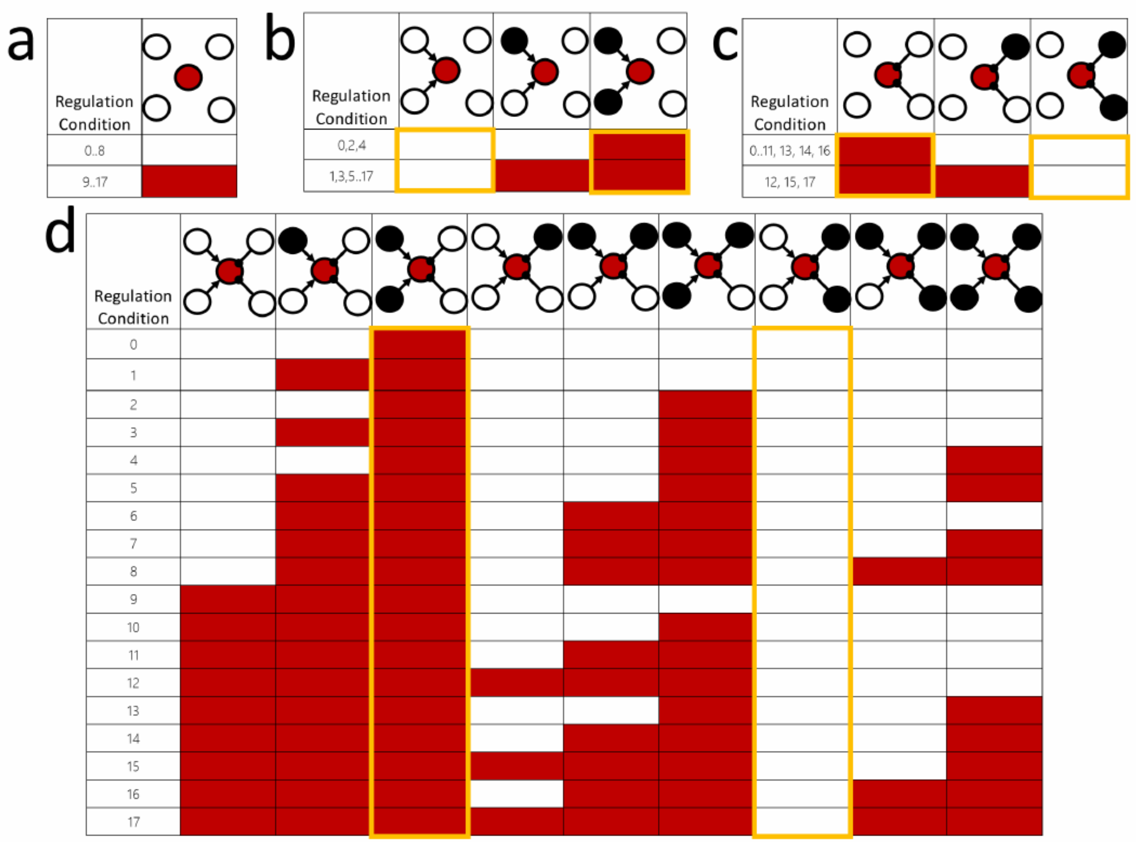
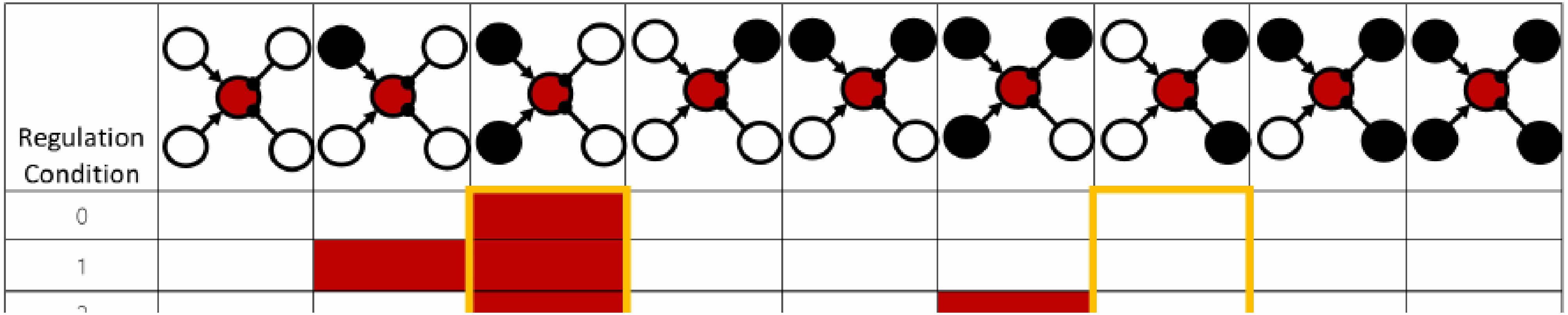


Figure 2: **The set of 18 regulation conditions (excluding the two threshold rules).**

we define the following regulation condition templates:

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

ABN Regulation Conditions



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

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

Observations

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

```
$TwoiPlusLifCultureConditions :=
{
 LIF = 1 and
 CH = 1 and
 PD = 1
};
```

```
$TwoiPlusLif:=
{
 MEKERK = 0 and
 Oct4=1 and
 Sox2=1 and
 Nanog=1 and
 Esrrb=1 and
 Klf2=1 and
 Tfcp2l1=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

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

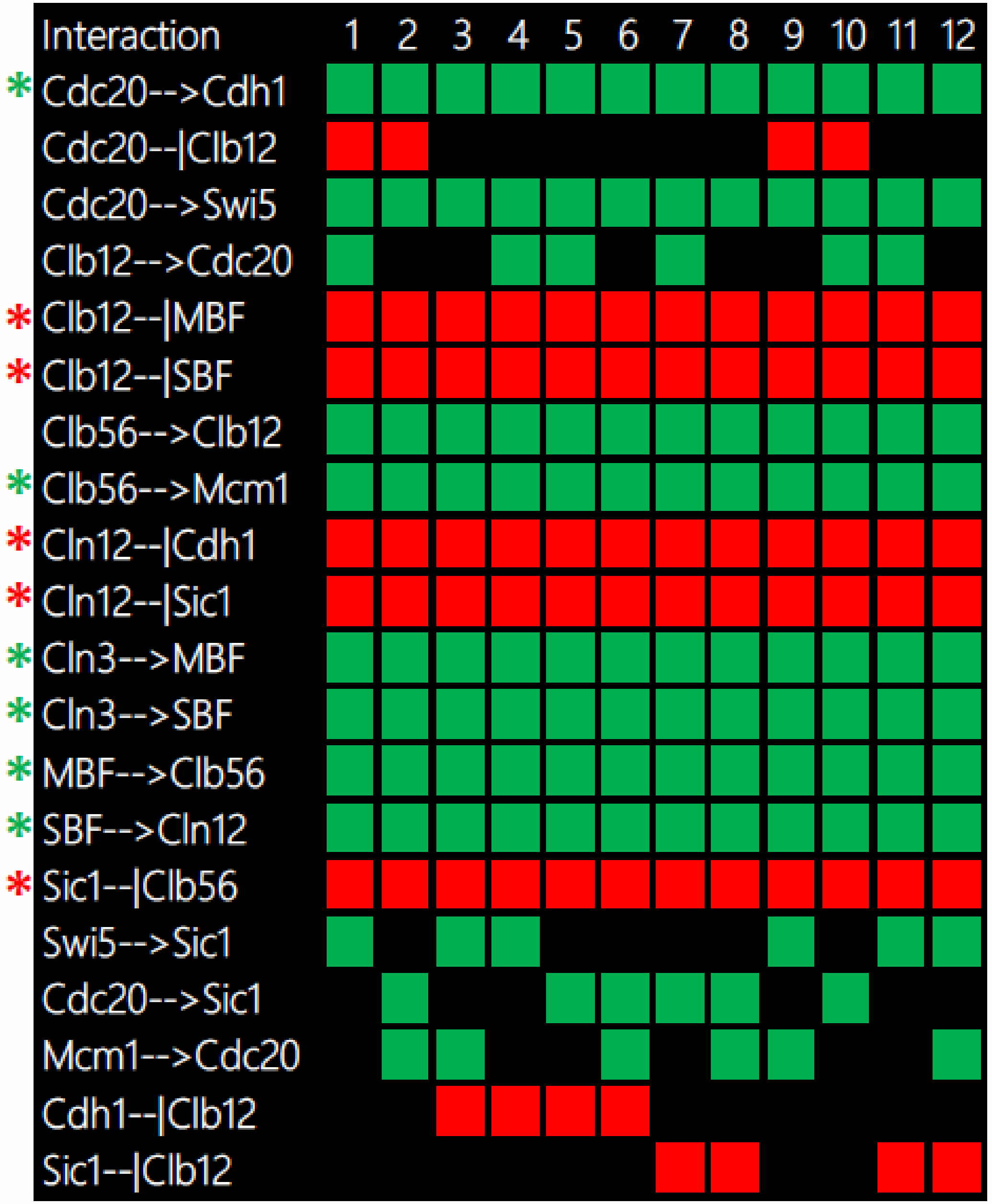
State of the art (Bio)

- RE:IN
- Check

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

- Record a list of (I,R)

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



SMT Quantifiers

- Z3 comes with
  - *E-matching* for unsatisfiable problems
  - *Model based quantifier instantiation* (MBQI) for satisfiable problems

- MBQI
  - Skolemize

$$\forall u \exists e \,.\, \phi(u, e) \;\rightarrow\; \forall u \sout{\exists e \,.}\, \phi(u, \sout{e} f_e(u))$$

  - 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 u

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

- Find out what would have worked

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

- Fix the broken model

$$f_e'(x) = ITE(u, e, f_e(x))$$

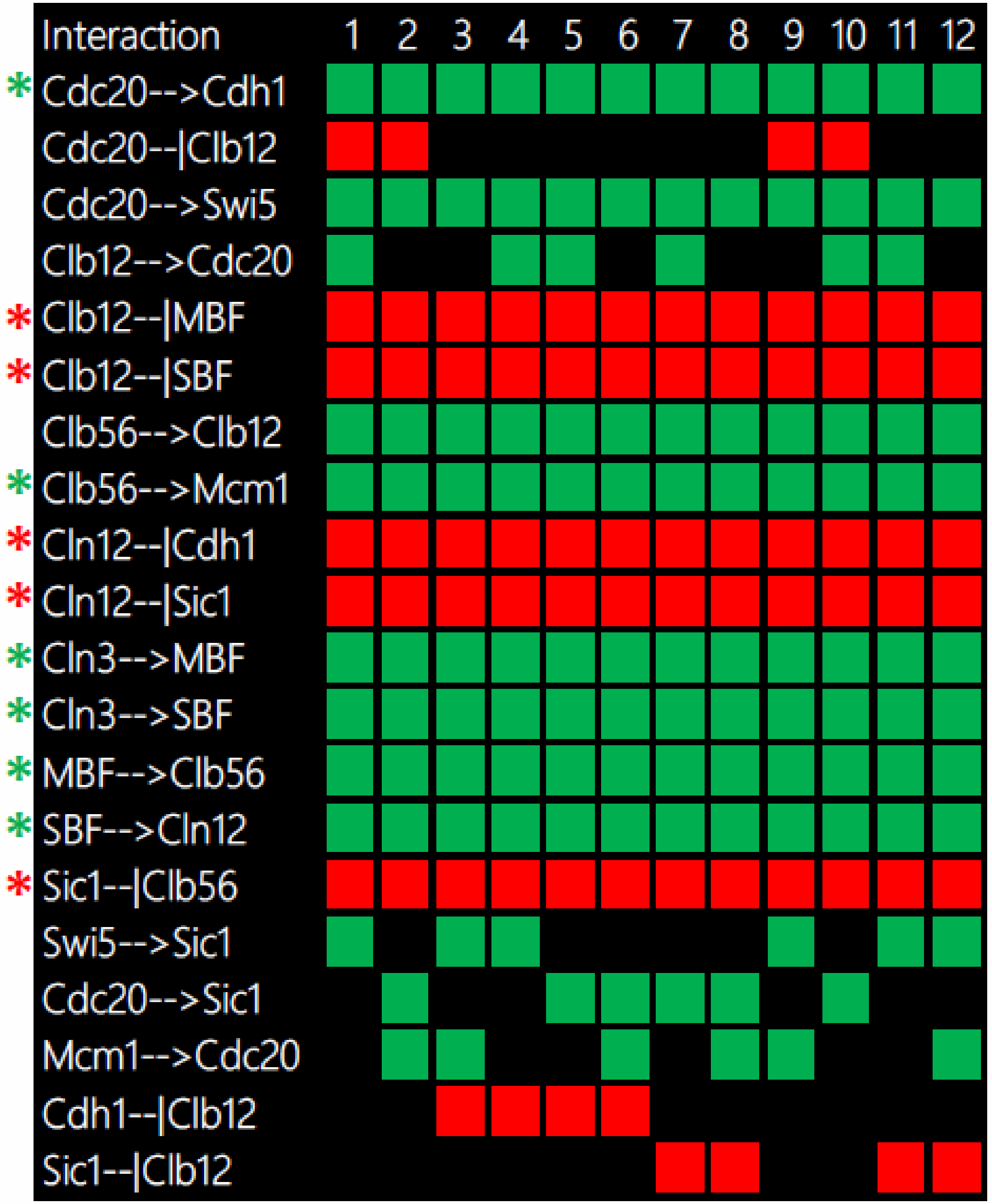
State of the art (SMT)

- MBQI
- Check

$$\exists u.\ \neg\phi(u, m)$$
$$\exists e.\quad \phi(u, e)$$

- Record a list of counter-examples & fixes (u, e)

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



Skolem Function Templates

- Skolemize into

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

- 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 $\boldsymbol{n}$ Booleans
  - Has equivalent ANF

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

- Such that the coefficients $a_i \in \{0,1\}$ fully describe f.
- E.g., $\mathrm{x}_1 \wedge x_2 = 1 \oplus x_1 \oplus x_2 \oplus x_1 x_2$

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 $\mathrm{f_u}$

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

Templated Branches

- Instead of lists of counter-examples

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

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

Template Refinement

- When there are no solutions
- Increase template complexity

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

Scale Example

- 16 genes, 8 optional interactions, 2 experiments
- RE:IN enumerated 96 solutions

- ((Nanog --> Sox2) AND (Klf2 --> Oct4)) OR ((Sall4 --> Sox2) AND (Klf2 --> 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

Microsoft