

Counterexample-Guided Repair in Boolean Functional Synthesis

Supratik Chakraborty

Indian Institute of Technology Bombay

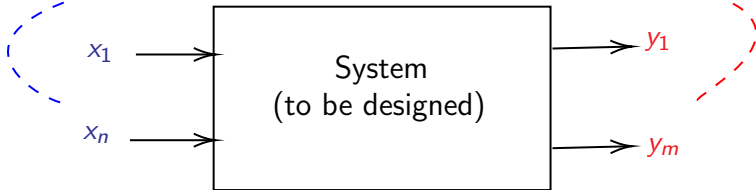
Joint work with S. Akshay, Ajith John, Shetal Shah

Workshop on Synthesis of Models & Systems, April 5, 2021
Simons Institute for the Theory of Computing

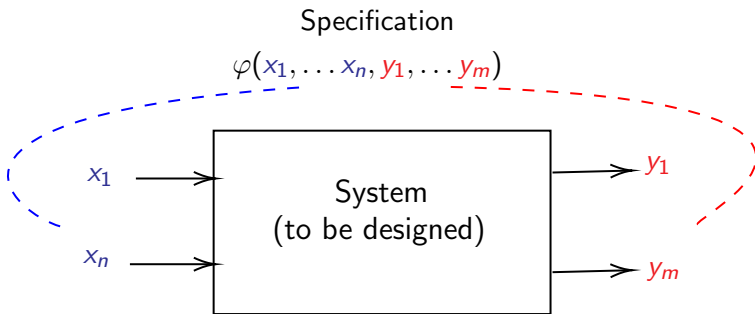
Synthesis: A Generic View

Specification

$$\varphi(x_1, \dots, x_n, y_1, \dots, y_m)$$

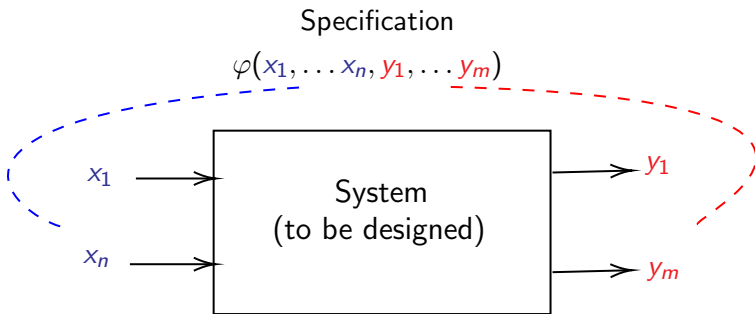


Synthesis: A Generic View



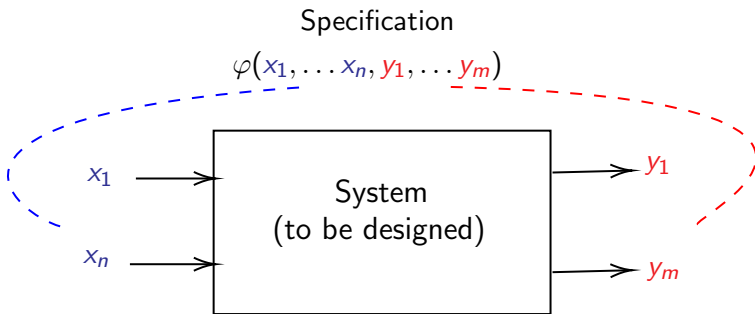
- Goal: Automatically synthesize system s.t. it satisfies $\varphi(x_1, \dots, x_n, y_1, \dots, y_m)$
 - x_i *input* variables (vector X)
 - y_j *output* variables (vector Y)

Synthesis: A Generic View



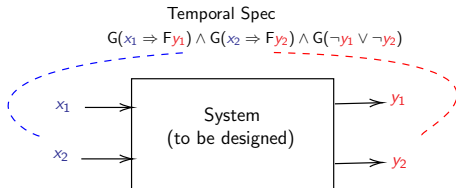
- Goal: Automatically synthesize system s.t. it satisfies $\varphi(x_1, \dots, x_n, y_1, \dots, y_m)$ **whenever possible**.
 - x_i *input* variables (vector X)
 - y_j *output* variables (vector Y)

Synthesis: A Generic View



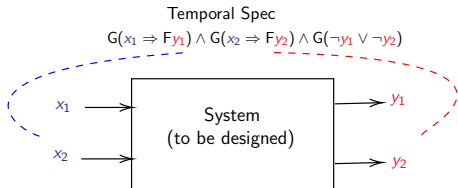
- Goal: Automatically synthesize system s.t. it satisfies $\varphi(x_1, \dots, x_n, y_1, \dots, y_m)$ **whenever possible**.
 - x_i input variables (vector X)
 - y_j output variables (vector Y)
- Need Y as functions F of
 - “History” of X and Y , “State” of system, ...such that $\varphi(X, F)$ is satisfied.

Synthesis: Some Examples

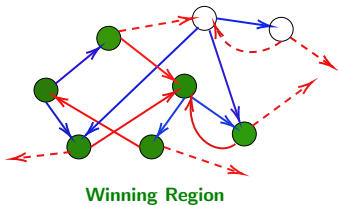


- Synthesize Y as function of
 - State (summarizing “history” of X and Y)

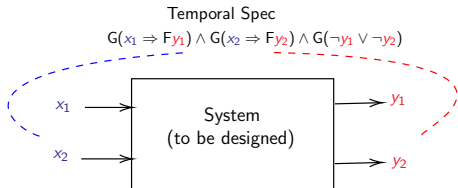
Synthesis: Some Examples



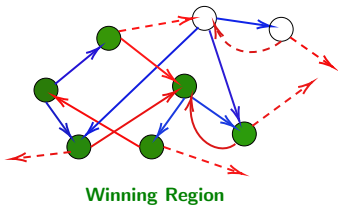
- Synthesize Y as function of
 - State (summarizing “history” of X and Y)



Synthesis: Some Examples

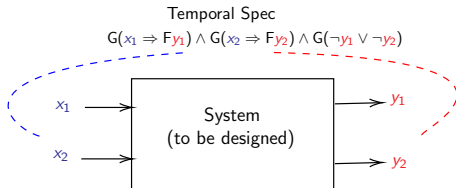


- Synthesize Y as function of
 - State (summarizing “history” of X and Y)

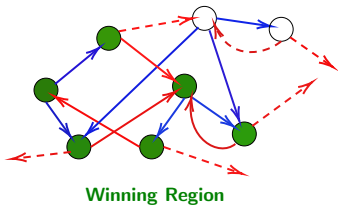


- Synthesize *winning strategy* to stay within *winning region*

Synthesis: Some Examples

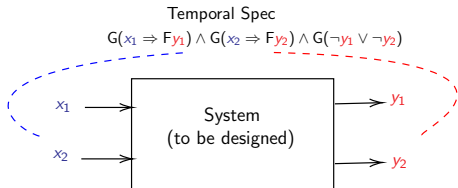


- Synthesize Y as function of
 - State (summarizing “history” of X and Y)

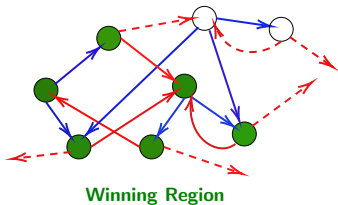


- Synthesize *winning strategy* to stay within *winning region*
 - $\text{WinRgn}(\text{NxtSt}(\text{state}, Y)) = 1$

Synthesis: Some Examples

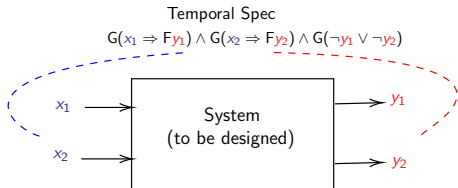


- Synthesize Y as function of
 - State (summarizing “history” of X and Y)

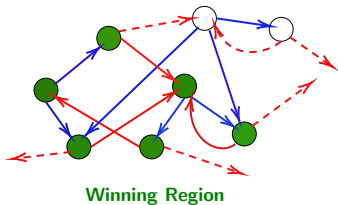


- Synthesize *winning strategy* to stay within *winning region*
 - $\text{WinRgn}(\text{NxtSt}(\text{state}, Y)) = 1$
 - No temporal operators

Synthesis: Some Examples

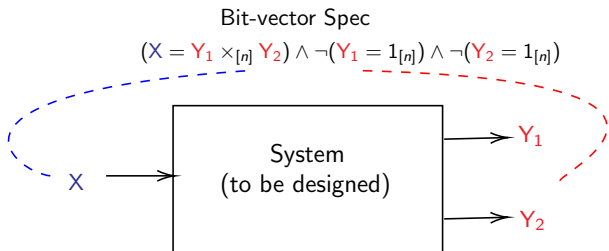


- Synthesize Y as function of
 - State (summarizing “history” of X and Y)



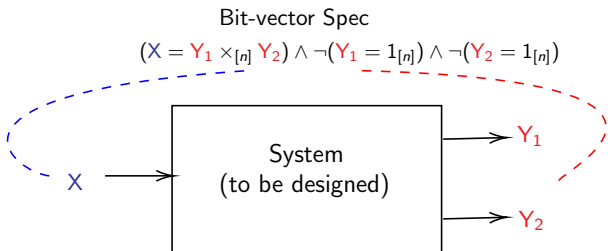
- Synthesize *winning strategy* to stay within *winning region*
 - $\text{WinRgn}(\text{NxtSt}(\text{state}, Y)) = 1$
 - No temporal operators
 - Not always satisfiable

Synthesis: Some Examples



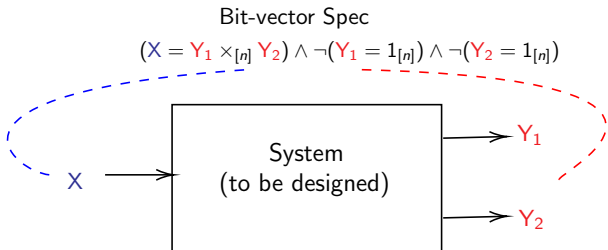
- Synthesize Y_1, Y_2 as functions of X

Synthesis: Some Examples



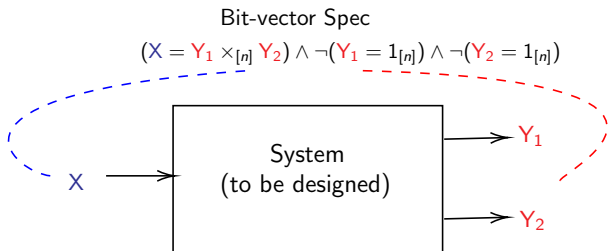
- Synthesize Y_1, Y_2 as functions of X
 - Spec has no temporal operators

Synthesis: Some Examples



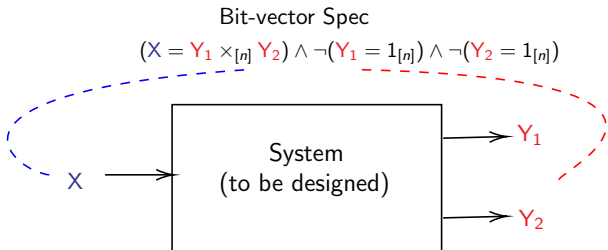
- Synthesize Y_1, Y_2 as functions of X
 - Spec has no temporal operators
 - Y_1, Y_2 must be non-trivial factors of X

Synthesis: Some Examples



- Synthesize Y_1, Y_2 as functions of X
 - Spec has no temporal operators
 - Y_1, Y_2 must be non-trivial factors of X
 - Not always satisfiable (if X is prime)

Synthesis: Some Examples



- Synthesize Y_1, Y_2 as functions of X
 - Spec has no temporal operators
 - Y_1, Y_2 must be non-trivial factors of X
 - Not always satisfiable (if X is prime)
 - Efficient solution would break crypto systems

Formal definition

Given Boolean relation $\varphi(x_1, \dots, x_n, y_1, \dots, y_m)$

- x_i *input* variables (vector X)
- y_j *output* variables (vector Y)

Formal definition

Given Boolean relation $\varphi(x_1, \dots, x_n, y_1, \dots, y_m)$

- x_i *input* variables (vector X)
- y_j *output* variables (vector Y)

Synthesize Boolean functions $F_j(X)$ for each y_j s.t.

$$\forall X (\exists y_1 \dots y_m \varphi(X, y_1 \dots y_m) \Leftrightarrow \varphi(X, F_1(X), \dots, F_m(X)))$$

Formal definition

Given Boolean relation $\varphi(x_1, \dots, x_n, y_1, \dots, y_m)$

- x_i *input* variables (vector X)
- y_j *output* variables (vector Y)

Synthesize Boolean functions $F_j(X)$ for each y_j s.t.

$$\forall X (\exists y_1 \dots y_m \varphi(X, y_1 \dots y_m) \Leftrightarrow \varphi(X, F_1(X), \dots, F_m(X)))$$

$F_j(X)$ is also called a *Skolem function* for y_j in φ .

Formal definition

Given Boolean relation $\varphi(x_1, \dots, x_n, y_1, \dots, y_m)$

- x_i input variables (vector X)
- y_j output variables (vector Y)

Synthesize Boolean functions $F_j(X)$ for each y_j s.t.

$$\forall X (\exists y_1 \dots y_m \varphi(X, y_1 \dots y_m) \Leftrightarrow \varphi(X, F_1(X), \dots, F_m(X)))$$

$F_j(X)$ is also called a *Skolem function* for y_j in φ .

- What if $\forall X \exists Y \varphi(X, Y) = 0$?

Formal definition

Given Boolean relation $\varphi(x_1, \dots, x_n, y_1, \dots, y_m)$

- x_i input variables (vector X)
- y_j output variables (vector Y)

Synthesize Boolean functions $F_j(X)$ for each y_j s.t.

$$\forall X (\exists y_1 \dots y_m \varphi(X, y_1 \dots y_m) \Leftrightarrow \varphi(X, F_1(X), \dots, F_m(X)))$$

$F_j(X)$ is also called a *Skolem function* for y_j in φ .

- What if $\forall X \exists Y \varphi(X, Y) = 0$?
 - Interesting as long as $\exists X \exists Y \varphi(X, Y) = 1$

Formal definition

Given Boolean relation $\varphi(x_1, \dots, x_n, y_1, \dots, y_m)$

- x_i input variables (vector X)
- y_j output variables (vector Y)

Synthesize Boolean functions $F_j(X)$ for each y_j s.t.

$$\forall X (\exists y_1 \dots y_m \varphi(X, y_1 \dots y_m) \Leftrightarrow \varphi(X, F_1(X), \dots, F_m(X)))$$

$F_j(X)$ is also called a *Skolem function* for y_j in φ .

- What if $\forall X \exists Y \varphi(X, Y) = 0$?
 - Interesting as long as $\exists X \exists Y \varphi(X, Y) = 1$
 - $F(X)$ must give right value of Y for all X s.t. $\exists Y \varphi(X, Y) = 1$
 - $F(X)$ inconsequential for other X

Formal definition

Given Boolean relation $\varphi(x_1, \dots, x_n, y_1, \dots, y_m)$

- x_i input variables (vector X)
- y_j output variables (vector Y)

Synthesize Boolean functions $F_j(X)$ for each y_j s.t.

$$\forall X (\exists y_1 \dots y_m \varphi(X, y_1 \dots y_m) \Leftrightarrow \varphi(X, F_1(X), \dots, F_m(X)))$$

$F_j(X)$ is also called a *Skolem function* for y_j in φ .

- What if $\forall X \exists Y \varphi(X, Y) = 0$?
 - Interesting as long as $\exists X \exists Y \varphi(X, Y) = 1$
 - $F(X)$ must give right value of Y for all X s.t. $\exists Y \varphi(X, Y) = 1$
 - $F(X)$ inconsequential for other X
 - Given X , $F(X)$, easy to check if $\exists Y \varphi(X, Y) = \varphi(X, F(X)) = 0$

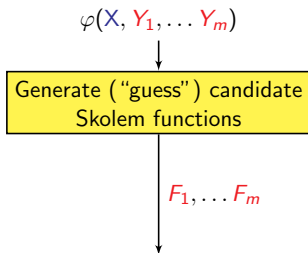
Applications of Boolean Functional Synthesis

1. Cryptanalysis: Interesting but hard for synthesis!
2. Disjunctive decomposition of symbolic transition relations
[Trivedi et al'02]
3. Quantifier elimination, of course!
 - $\exists Y \varphi(X, Y) \equiv \varphi(X, F(X))$
4. Certifying QBF-SAT solvers
 - Nice survey of applications by Shukla et al'19
5. Reactive controller synthesis
 - Synthesizing moves to stay within winning region
6. Program synthesis
 - Combinatorial sketching [Solar-Lezama et al'06, Srivastava et al'13]
 - Complete functional synthesis [Kuncak et al'10]
7. Repair/partial synthesis of circuits [Fujita et al'13]

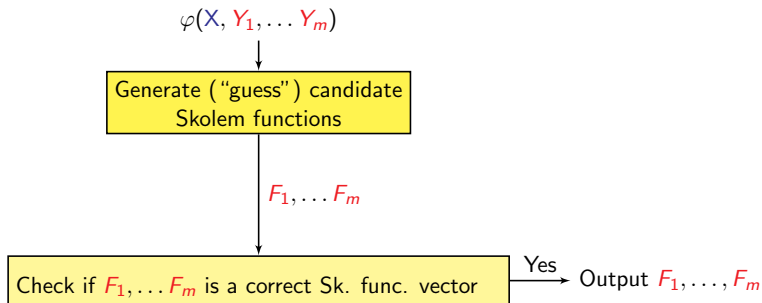
Existing Approaches

1. Closely related to most general Boolean unifiers
 - Boole'1847, Lowenheim'1908, Macii'98
2. Extract Sk. functions from proof of validity of $\forall X \exists Y \varphi(X, Y)$
 - Bendetti'05, Jussilla et al'07, Balabanov et al'12, Heule et al'14
3. Using templates: Solar-Lezama et al'06, Srivastava et al'13
4. Self-substitution + function composition: Jiang'09, Trivedi'03
5. Synthesis from special normal form representation of specification
 - From ROBDDs: Kukula et al'00, Kuncak et al'10, Fried et al'16, Tabajara et al'17
 - From SynNNF: Akshay et al'09
6. Incremental determinization: Rabe et al'17,'18
7. Quantifier instantiation techniques in SMT solvers
 - Barrett et al'15, Bierre et al'17
8. Input/output component separation: C. et al'18
9. Guess/learn Skolem function candidate + check + **repair**
 - John et al'15, Akshay et al'17,'18,'20, Golia et al'20

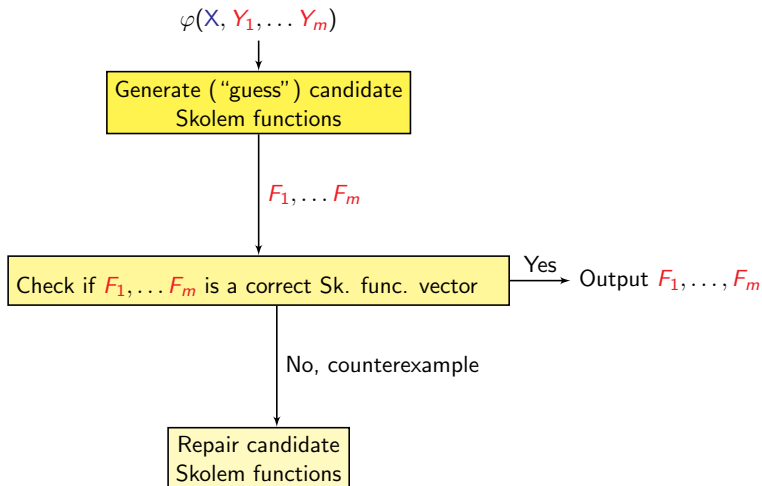
A Practically Effective Solution: Guess-Check-Repair



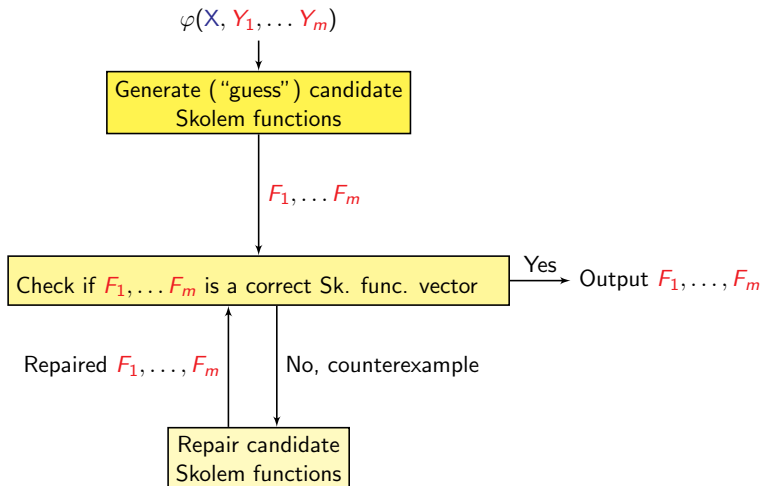
A Practically Effective Solution: Guess-Check-Repair



A Practically Effective Solution: Guess-Check-Repair



A Practically Effective Solution: Guess-Check-Repair



A Useful “Guess”-ing Trick

Fix a linear ordering of outputs: $y_1 \prec y_2 \prec \dots \prec y_m$,

A Useful “Guess”-ing Trick

Fix a linear ordering of outputs: $y_1 \prec y_2 \prec \dots \prec y_m$, and compute

- y_m as $G_m(X, y_1, \dots, y_{m-1})$ from spec $\varphi(X, y_1, \dots, y_{m-1}, y_m)$

A Useful “Guess”-ing Trick

Fix a linear ordering of outputs: $y_1 \prec y_2 \prec \dots \prec y_m$, and compute

- y_m as $G_m(X, y_1, \dots, y_{m-1})$ from spec $\varphi(X, y_1, \dots, y_{m-1}, y_m)$
- y_{m-1} as $G_{m-1}(X, y_1, \dots, y_{m-2})$ from $\exists y_m \varphi(X, y_1, \dots, y_{m-2}, y_{m-1}, y_m)$
- \vdots

A Useful “Guess”-ing Trick

Fix a linear ordering of outputs: $y_1 \prec y_2 \prec \dots \prec y_m$, and compute

- y_m as $G_m(X, y_1, \dots, y_{m-1})$ from spec $\varphi(X, y_1, \dots, y_{m-1}, y_m)$
- y_{m-1} as $G_{m-1}(X, y_1, \dots, y_{m-2})$ from $\exists y_m \varphi(X, y_1, \dots, y_{m-2}, y_{m-1}, y_m)$
- \vdots
- y_1 as $G_1(X)$ from $\exists y_2 \dots \exists y_m \varphi(X, y_1, y_2 \dots y_m)$

A Useful “Guess”-ing Trick

Fix a linear ordering of outputs: $y_1 \prec y_2 \prec \dots \prec y_m$, and compute

- y_m as $G_m(X, y_1, \dots, y_{m-1})$ from spec $\varphi(X, y_1, \dots, y_{m-1}, y_m)$
- y_{m-1} as $G_{m-1}(X, y_1, \dots, y_{m-2})$ from $\exists y_m \varphi(X, y_1, \dots, y_{m-2}, y_{m-1}, y_m)$
- \vdots
- y_1 as $G_1(X)$ from $\exists y_2 \dots \exists y_m \varphi(X, y_1, y_2 \dots y_m)$

- Generate Skolem functions for only 1-output specs
- Need to compute (approximations of) $\exists y_i \dots y_m \varphi(X, Y)$

A Useful “Guess”-ing Trick

Fix a linear ordering of outputs: $y_1 \prec y_2 \prec \dots \prec y_m$, and compute

- y_m as $G_m(X, y_1, \dots, y_{m-1})$ from spec $\varphi(X, y_1, \dots, y_{m-1}, y_m)$
- y_{m-1} as $G_{m-1}(X, y_1, \dots, y_{m-2})$ from $\exists y_m \varphi(X, y_1, \dots, y_{m-2}, y_{m-1}, y_m)$
- \vdots
- y_1 as $G_1(X)$ from $\exists y_2 \dots \exists y_m \varphi(X, y_1, y_2 \dots y_m)$

- Generate Skolem functions for only 1-output specs
- Need to compute (approximations of) $\exists y_i \dots y_m \varphi(X, Y)$

A $|X|$ -input, $|Y|$ -output circuit computing the desired Skolem function vector (F_1, \dots, F_m) can be constructed with

- $\#gates \leq \sum_{i=1}^m \#gates(G_i) + 2m$
- $\#wires \leq \sum_{i=1}^m \#wires(G_i) + \frac{m(m-1)}{2}$

A Useful “Guess”-ing Trick

Fix a linear ordering of outputs: $y_1 \prec y_2 \prec \dots \prec y_m$, and compute

- y_m as $G_m(X, y_1, \dots, y_{m-1})$ from spec $\varphi(X, y_1, \dots, y_{m-1}, y_m)$
- y_{m-1} as $G_{m-1}(X, y_1, \dots, y_{m-2})$ from $\exists y_m \varphi(X, y_1, \dots, y_{m-2}, y_{m-1}, y_m)$
- \vdots
- y_1 as $G_1(X)$ from $\exists y_2 \dots \exists y_m \varphi(X, y_1, y_2 \dots y_m)$

- Generate Skolem functions for only 1-output specs
- Need to compute (approximations of) $\exists y_i \dots y_m \varphi(X, Y)$

A $|X|$ -input, $|Y|$ -output circuit computing the desired Skolem function vector (F_1, \dots, F_m) can be constructed with

- #gates $\leq \sum_{i=1}^m \text{\#gates}(G_i) + 2m$
- #wires $\leq \sum_{i=1}^m \text{\#wires}(G_i) + \frac{m(m-1)}{2}$

Sufficient to compute the G_i functions

The “Guess”ing Game

Single output: $\varphi(X, y)$

- Both $\varphi(X, 1)$ and $\neg\varphi(X, 0)$ are Skolem functions
- No guess-work!

The “Guess”ing Game

Single output: $\varphi(X, y)$

- Both $\varphi(X, 1)$ and $\neg\varphi(X, 0)$ are Skolem functions
- No guess-work!

Multiple outputs: $\varphi(X, y_1, \dots, y_m)$

- Need to compute $\exists y_i \dots \exists y_m \varphi(X, y_1, \dots, y_{i-2}, y_{i-1}, y_i, \dots, y_m)$
 - Hard in general
- Approximations easily obtained from NNF of φ

The “Guess”ing Game

Single output: $\varphi(X, y)$

- Both $\varphi(X, 1)$ and $\neg\varphi(X, 0)$ are Skolem functions
- No guess-work!

Multiple outputs: $\varphi(X, y_1, \dots, y_m)$

- Need to compute $\exists y_1 \dots \exists y_m \varphi(X, y_1, \dots, y_{i-2}, y_{i-1}, y_i, \dots, y_m)$
 - Hard in general
- Approximations easily obtained from NNF of φ
 - Over-approx: Set y_i, \dots, y_m and $\neg y_i, \dots, \neg y_m$ leaves to 1

The “Guess”ing Game

Single output: $\varphi(X, y)$

- Both $\varphi(X, 1)$ and $\neg\varphi(X, 0)$ are Skolem functions
- No guess-work!

Multiple outputs: $\varphi(X, y_1, \dots, y_m)$

- Need to compute $\exists y_i \dots \exists y_m \varphi(X, y_1, \dots, y_{i-2}, y_{i-1}, y_i, \dots, y_m)$
 - Hard in general
- Approximations easily obtained from NNF of φ
 - Over-approx: Set y_i, \dots, y_m and $\neg y_i, \dots, \neg y_m$ leaves to 1
 - Under-approx: Set y_i, \dots, y_m and $\neg y_i, \dots, \neg y_m$ leaves to 0

The “Guess”ing Game

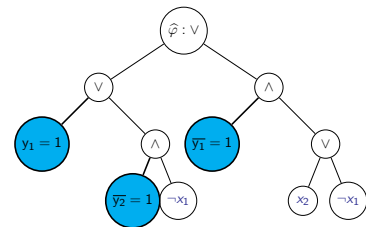
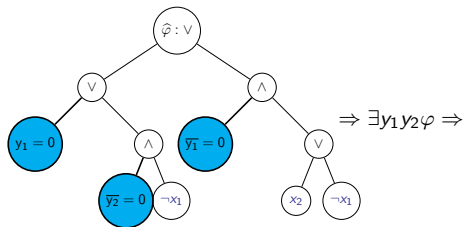
Single output: $\varphi(X, y)$

- Both $\varphi(X, 1)$ and $\neg\varphi(X, 0)$ are Skolem functions
- No guess-work!

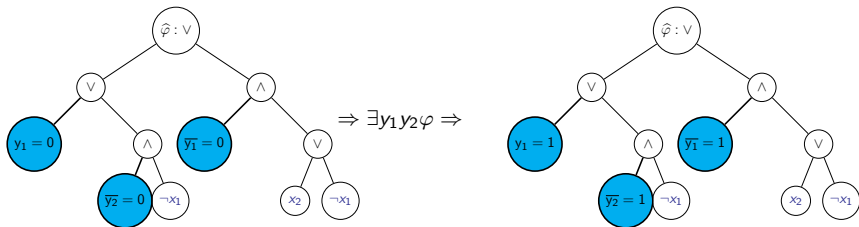
Multiple outputs: $\varphi(X, y_1, \dots, y_m)$

- Need to compute $\exists y_i \dots \exists y_m \varphi(X, y_1, \dots, y_{i-2}, y_{i-1}, y_i, \dots, y_m)$
 - Hard in general
- Approximations easily obtained from NNF of φ
 - Over-approx: Set y_i, \dots, y_m and $\neg y_i, \dots, \neg y_m$ leaves to 1
 - Under-approx: Set y_i, \dots, y_m and $\neg y_i, \dots, \neg y_m$ leaves to 0
 - Use approximations to “guess” candidate functions G_i
 - Guarantees over-/under-approximation of guessed Skolem functions

Illustrating Approximations



Illustrating Approximations



- $\hat{\varphi}(x_1 \dots x_n, \overbrace{0 \dots 0}^i, y_{i+1} \dots y_m, \overbrace{0 \dots 0}^i, \neg y_{i+1} \dots \neg y_m) \Rightarrow \exists y_1 \dots y_i \varphi(\dots)$
- $\hat{\varphi}(x_1 \dots x_n, \overbrace{1 \dots 1}^i, y_{i+1} \dots y_m, \overbrace{1 \dots 1}^i, \neg y_{i+1} \dots \neg y_m) \Leftarrow \exists y_1 \dots y_i \varphi(\dots)$

Checking correctness of “guess”-ed Skolem functions

Given candidate Skolem functions $F_1, \dots, F_m,$

$$\text{Is } \forall X (\exists Y \varphi(X, Y) \Leftrightarrow \varphi(X, F(X))) ?$$

Checking correctness of “guess”-ed Skolem functions

Given candidate Skolem functions $F_1, \dots, F_m,$

$$\text{Is } \forall X (\exists Y \varphi(X, Y) \Leftrightarrow \varphi(X, F(X))) ?$$

Can we avoid using a QBF solver?

Checking correctness of “guess”-ed Skolem functions

Given candidate Skolem functions F_1, \dots, F_m ,

$$\text{Is } \forall X (\exists Y \varphi(X, Y) \Leftrightarrow \varphi(X, F(X))) ?$$

Can we avoid using a QBF solver?

Yes, we can! [ACGKS'15]

- Propositional error formula $\varepsilon(X, Y, Y')$:

$$(\varphi(X, Y') \wedge \bigwedge_{j=1}^m (Y_j \Leftrightarrow F_j) \wedge \neg \varphi(X, Y))$$

Checking correctness of “guess”-ed Skolem functions

Given candidate Skolem functions F_1, \dots, F_m ,

$$\text{Is } \forall X (\exists Y \varphi(X, Y) \Leftrightarrow \varphi(X, F(X))) ?$$

Can we avoid using a QBF solver?

Yes, we can! [ACGKS'15]

- Propositional error formula $\varepsilon(X, Y, Y')$:

$$(\varphi(X, Y') \wedge \bigwedge_{j=1}^m (Y_j \Leftrightarrow F_j) \wedge \neg \varphi(X, Y))$$

- ε unsatisfiable iff F_1, \dots, F_m is correct Skolem function vector

Checking correctness of “guess”-ed Skolem functions

Given candidate Skolem functions F_1, \dots, F_m ,

$$\text{Is } \forall X (\exists Y \varphi(X, Y) \Leftrightarrow \varphi(X, F(X))) ?$$

Can we avoid using a QBF solver?

Yes, we can! [ACGKS'15]

- Propositional error formula $\varepsilon(X, Y, Y')$:

$$(\varphi(X, Y') \wedge \bigwedge_{j=1}^m (Y_j \Leftrightarrow F_j) \wedge \neg \varphi(X, Y))$$

- ε unsatisfiable iff F_1, \dots, F_m is correct Skolem function vector
- Suppose σ : satisfying assignment of ε

Checking correctness of “guess”-ed Skolem functions

Given candidate Skolem functions F_1, \dots, F_m ,

$$\text{Is } \forall X (\exists Y \varphi(X, Y) \Leftrightarrow \varphi(X, F(X))) ?$$

Can we avoid using a QBF solver?

Yes, we can! [ACGKS'15]

- Propositional error formula $\varepsilon(X, Y, Y')$:

$$(\varphi(X, Y') \wedge \bigwedge_{j=1}^m (Y_j \Leftrightarrow F_j) \wedge \neg \varphi(X, Y))$$

- ε unsatisfiable iff F_1, \dots, F_m is correct Skolem function vector
- Suppose σ : satisfying assignment of ε
 - $\varphi(\sigma[X], \sigma[Y']) = 1, \quad \sigma[Y] = F(\sigma[X]), \quad \varphi(\sigma[X], \sigma[Y]) = 0$

Checking correctness of “guess”-ed Skolem functions

Given candidate Skolem functions F_1, \dots, F_m ,

$$\text{Is } \forall X (\exists Y \varphi(X, Y) \Leftrightarrow \varphi(X, F(X))) ?$$

Can we avoid using a QBF solver?

Yes, we can! [ACGKS'15]

- Propositional error formula $\varepsilon(X, Y, Y')$:

$$\left(\varphi(X, Y') \wedge \bigwedge_{j=1}^m (Y_j \Leftrightarrow F_j) \wedge \neg \varphi(X, Y) \right)$$

- ε unsatisfiable iff F_1, \dots, F_m is correct Skolem function vector
- Suppose σ : satisfying assignment of ε
 - $\varphi(\sigma[X], \sigma[Y']) = 1$, $\sigma[Y] = F(\sigma[X])$, $\varphi(\sigma[X], \sigma[Y]) = 0$
 - σ is **counterexample** to the claim that F_1, \dots, F_m is a correct Skolem function vector

Counterexample Generalization

Recall: Skolem functions guessed from **approximations** of $\exists y_{i+1} \dots \exists y_m \varphi(X, y_1, \dots, y_{i-1}, y_i, y_{i+1}, \dots, y_m)$

Counterexample Generalization

Recall: Skolem functions guessed from approximations of

$$\exists y_{i+1} \dots \exists y_m \varphi(X, y_1, \dots, y_{i-1}, y_i, y_{i+1}, \dots, y_m)$$

- Let $\exists y_{i+1} \dots \exists y_m \varphi(X, Y) \Rightarrow \Theta_i(X, y_1, \dots, y_{i-1}, y_i)$ Over-approx

Recall: Skolem functions guessed from **approximations** of

$\exists y_{i+1} \dots \exists y_m \varphi(X, y_1, \dots, y_{i-1}, y_i, y_{i+1}, \dots, y_m)$

• Let $\exists y_{i+1} \dots \exists y_m \varphi(X, Y) \Rightarrow \Theta_i(X, y_1, \dots, y_{i-1}, y_i)$

• Let $\delta_i = \neg \Theta_i|_{y_i=0}$; $\gamma_i = \neg \Theta_i|_{y_i=1}$ Under-approximations

Recall: Skolem functions guessed from **approximations** of

$\exists y_{i+1} \dots \exists y_m \varphi(X, y_1, \dots, y_{i-1}, y_i, y_{i+1}, \dots, y_m)$

- Let $\exists y_{i+1} \dots \exists y_m \varphi(X, Y) \Rightarrow \Theta_i(X, y_1, \dots, y_{i-1}, y_i)$
- Let $\delta_i = \neg \Theta_i|_{y_i=0}$; $\gamma_i = \neg \Theta_i|_{y_i=1}$
- Initial guess $G_i(X, y_1, \dots, y_{i-1}) \in \{\delta_i, \neg \gamma_i\}$

Counterexample Generalization

Recall: Skolem functions guessed from **approximations** of

$$\exists y_{i+1} \dots \exists y_m \varphi(X, y_1, \dots, y_{i-1}, y_i, y_{i+1}, \dots, y_m)$$

- Let $\exists y_{i+1} \dots \exists y_m \varphi(X, Y) \Rightarrow \Theta_i(X, y_1, \dots, y_{i-1}, y_i)$
- Let $\delta_i = \neg \Theta_i|_{y_i=0}$; $\gamma_i = \neg \Theta_i|_{y_i=1}$
- Initial guess $G_i(X, y_1, \dots, y_{i-1}) \in \{\delta_i, \neg \gamma_i\}$... **1-sided error**
 - $G_i = \delta_i$ **cannot err** if it evaluates to 1
 - $G_i = \neg \gamma_i$ **cannot err** if it evaluates to 0

Counterexample Generalization

Recall: Skolem functions guessed from **approximations** of

$\exists y_{i+1} \dots \exists y_m \varphi(X, y_1, \dots, y_{i-1}, y_i, y_{i+1}, \dots, y_m)$

- Let $\exists y_{i+1} \dots \exists y_m \varphi(X, Y) \Rightarrow \Theta_i(X, y_1, \dots, y_{i-1}, y_i)$
- Let $\delta_i = \neg \Theta_i|_{y_i=0}$; $\gamma_i = \neg \Theta_i|_{y_i=1}$
- Initial guess $G_i(X, y_1, \dots, y_{i-1}) \in \{\delta_i, \neg \gamma_i\}$... **1-sided error**
 - $G_i = \delta_i$ **cannot err** if it evaluates to 1
 - $G_i = \neg \gamma_i$ **cannot err** if it evaluates to 0

Generalized counterexample

Given $\sigma \models \varepsilon(X, Y, Y')$ and δ_i, γ_i for $1 \leq i \leq m$

Counterexample Generalization

Recall: Skolem functions guessed from **approximations** of

$\exists y_{i+1} \dots \exists y_m \varphi(X, y_1, \dots, y_{i-1}, y_i, y_{i+1}, \dots, y_m)$

- Let $\exists y_{i+1} \dots \exists y_m \varphi(X, Y) \Rightarrow \Theta_i(X, y_1, \dots, y_{i-1}, y_i)$
- Let $\delta_i = \neg \Theta_i|_{y_i=0}$; $\gamma_i = \neg \Theta_i|_{y_i=1}$
- Initial guess $G_i(X, y_1, \dots, y_{i-1}) \in \{\delta_i, \neg \gamma_i\}$... **1-sided error**
 - $G_i = \delta_i$ **cannot err** if it evaluates to 1
 - $G_i = \neg \gamma_i$ **cannot err** if it evaluates to 0

Generalized counterexample

Given $\sigma \models \varepsilon(X, Y, Y')$ and δ_i, γ_i for $1 \leq i \leq m$

Find function $\mu(X, y_1, \dots, y_{j-1})$ for some $j \in \{1, \dots, m\}$ s.t.

Counterexample Generalization

Recall: Skolem functions guessed from **approximations** of

$\exists y_{i+1} \dots \exists y_m \varphi(X, y_1, \dots, y_{i-1}, y_i, y_{i+1}, \dots, y_m)$

- Let $\exists y_{i+1} \dots \exists y_m \varphi(X, Y) \Rightarrow \Theta_i(X, y_1, \dots, y_{i-1}, y_i)$
- Let $\delta_i = \neg \Theta_i|_{y_i=0}$; $\gamma_i = \neg \Theta_i|_{y_i=1}$
- Initial guess $G_i(X, y_1, \dots, y_{i-1}) \in \{\delta_i, \neg \gamma_i\}$... **1-sided error**
 - $G_i = \delta_i$ **cannot err** if it evaluates to 1
 - $G_i = \neg \gamma_i$ **cannot err** if it evaluates to 0

Generalized counterexample

Given $\sigma \models \varepsilon(X, Y, Y')$ and δ_i, γ_i for $1 \leq i \leq m$

Find function $\mu(X, y_1, \dots, y_{j-1})$ for some $j \in \{1, \dots, m\}$ s.t.

- $\sigma \models \mu$... μ **generalizes** σ

Counterexample Generalization

Recall: Skolem functions guessed from **approximations** of

$\exists y_{i+1} \dots \exists y_m \varphi(X, y_1, \dots, y_{i-1}, y_i, y_{i+1}, \dots, y_m)$

- Let $\exists y_{i+1} \dots \exists y_m \varphi(X, Y) \Rightarrow \Theta_i(X, y_1, \dots, y_{i-1}, y_i)$
- Let $\delta_i = \neg \Theta_i|_{y_i=0}$; $\gamma_i = \neg \Theta_i|_{y_i=1}$
- Initial guess $G_i(X, y_1, \dots, y_{i-1}) \in \{\delta_i, \neg \gamma_i\}$... **1-sided error**
 - $G_i = \delta_i$ **cannot err** if it evaluates to 1
 - $G_i = \neg \gamma_i$ **cannot err** if it evaluates to 0

Generalized counterexample

Given $\sigma \models \varepsilon(X, Y, Y')$ and δ_i, γ_i for $1 \leq i \leq m$

Find function $\mu(X, y_1, \dots, y_{j-1})$ for some $j \in \{1, \dots, m\}$ s.t.

- $\sigma \models \mu$... μ **generalizes** σ
- $\mu \Rightarrow \gamma_j \wedge \delta_j$

Counterexample Generalization

Recall: Skolem functions guessed from **approximations** of

$\exists y_{i+1} \dots \exists y_m \varphi(X, y_1, \dots, y_{i-1}, y_i, y_{i+1}, \dots, y_m)$

- Let $\exists y_{i+1} \dots \exists y_m \varphi(X, Y) \Rightarrow \Theta_i(X, y_1, \dots, y_{i-1}, y_i)$
- Let $\delta_i = \neg \Theta_i|_{y_i=0}$; $\gamma_i = \neg \Theta_i|_{y_i=1}$
- Initial guess $G_i(X, y_1, \dots, y_{i-1}) \in \{\delta_i, \neg \gamma_i\}$... **1-sided error**
 - $G_i = \delta_i$ **cannot err** if it evaluates to 1
 - $G_i = \neg \gamma_i$ **cannot err** if it evaluates to 0

Generalized counterexample

Given $\sigma \models \varepsilon(X, Y, Y')$ and δ_i, γ_i for $1 \leq i \leq m$

Find function $\mu(X, y_1, \dots, y_{j-1})$ for some $j \in \{1, \dots, m\}$ s.t.

- $\sigma \models \mu$... μ **generalizes** σ
- $\mu \Rightarrow \gamma_j \wedge \delta_j$
 - $\Rightarrow \forall y_j \dots \forall y_m \neg \varphi(X, y_1, \dots, y_{j-1}, y_j, y_{j+1}, \dots, y_m)$

Counterexample Generalization

Recall: Skolem functions guessed from **approximations** of

$\exists y_{i+1} \dots \exists y_m \varphi(X, y_1, \dots, y_{i-1}, y_i, y_{i+1}, \dots, y_m)$

- Let $\exists y_{i+1} \dots \exists y_m \varphi(X, Y) \Rightarrow \Theta_i(X, y_1, \dots, y_{i-1}, y_i)$
- Let $\delta_i = \neg \Theta_i|_{y_i=0}$; $\gamma_i = \neg \Theta_i|_{y_i=1}$
- Initial guess $G_i(X, y_1, \dots, y_{i-1}) \in \{\delta_i, \neg \gamma_i\}$... **1-sided error**
 - $G_i = \delta_i$ **cannot err** if it evaluates to 1
 - $G_i = \neg \gamma_i$ **cannot err** if it evaluates to 0

Generalized counterexample

Given $\sigma \models \varepsilon(X, Y, Y')$ and δ_i, γ_i for $1 \leq i \leq m$

Find function $\mu(X, y_1, \dots, y_{j-1})$ for some $j \in \{1, \dots, m\}$ s.t.

- $\sigma \models \mu$... μ **generalizes** σ
- $\mu \Rightarrow \gamma_j \wedge \delta_j$
 - $\Rightarrow \forall y_j \dots \forall y_m \neg \varphi(X, y_1, \dots, y_{j-1}, y_j, y_{j+1}, \dots, y_m)$
 - If $\pi \models \mu$, no extension of π satisfies φ ... **counterexample**

Counterexample Generalization

Recall: Skolem functions guessed from **approximations** of

$\exists y_{i+1} \dots \exists y_m \varphi(X, y_1, \dots, y_{i-1}, y_i, y_{i+1}, \dots, y_m)$

- Let $\exists y_{i+1} \dots \exists y_m \varphi(X, Y) \Rightarrow \Theta_i(X, y_1, \dots, y_{i-1}, y_i)$
- Let $\delta_i = \neg \Theta_i|_{y_i=0}$; $\gamma_i = \neg \Theta_i|_{y_i=1}$
- Initial guess $G_i(X, y_1, \dots, y_{i-1}) \in \{\delta_i, \neg \gamma_i\}$... **1-sided error**
 - $G_i = \delta_i$ **cannot err** if it evaluates to 1
 - $G_i = \neg \gamma_i$ **cannot err** if it evaluates to 0

Generalized counterexample

Given $\sigma \models \varepsilon(X, Y, Y')$ and δ_i, γ_i for $1 \leq i \leq m$

Find function $\mu(X, y_1, \dots, y_{j-1})$ for some $j \in \{1, \dots, m\}$ s.t.

- $\sigma \models \mu$... μ **generalizes** σ
- $\mu \Rightarrow \gamma_j \wedge \delta_j$
 - $\Rightarrow \forall y_j \dots \forall y_m \neg \varphi(X, y_1, \dots, y_{j-1}, y_j, y_{j+1}, \dots, y_m)$
 - If $\pi \models \mu$, no extension of π satisfies φ ... **counterexample**

Must ensure that (X, G_1, \dots, G_{j-1}) never evaluates to π

Repairing “guess”-ed candidate Skolem functions

- Every model of $\mu(X, y_1, \dots, y_{j-1})$ gives a problematic combination of G_1, \dots, G_{j-1} values

Repairing “guess”-ed candidate Skolem functions

- Every model of $\mu(X, y_1, \dots, y_{j-1})$ gives a problematic combination of G_1, \dots, G_{j-1} values
- Flip G_{j-1} whenever μ holds

Repairing “guess”-ed candidate Skolem functions

- Every model of $\mu(X, y_1, \dots, y_{j-1})$ gives a problematic combination of G_1, \dots, G_{j-1} values
- Flip G_{j-1} whenever μ holds
 - Recall $G_{j-1} \in \{\neg\gamma_{j-1}, \delta_{j-1}\}$

Repairing “guess”-ed candidate Skolem functions

- Every model of $\mu(X, y_1, \dots, y_{j-1})$ gives a problematic combination of G_1, \dots, G_{j-1} values
- Flip G_{j-1} whenever μ holds
 - Recall $G_{j-1} \in \{\neg\gamma_{j-1}, \delta_{j-1}\}$
 - **Only** source of error: under-approximation of $\neg\exists y_j, \dots, \exists y_m \varphi(X, y_1, \dots, y_{j-2}, y_{j-1}, y_j, \dots, y_m)$

Repairing “guess”-ed candidate Skolem functions

- Every model of $\mu(X, y_1, \dots, y_{j-1})$ gives a problematic combination of G_1, \dots, G_{j-1} values
- Flip G_{j-1} whenever μ holds
 - Recall $G_{j-1} \in \{\neg\gamma_{j-1}, \delta_{j-1}\}$
 - **Only** source of error: under-approximation of $\neg\exists y_j, \dots, \exists y_m \varphi(X, y_1, \dots, y_{j-2}, y_{j-1}, y_j, \dots, y_m)$
 - Repair: **Expand under-approximation**
 - If G_{j-1} is $\neg\gamma_{j-1}$, $\gamma_{j-1} \leftarrow \gamma_{j-1} \vee \mu|_{\sigma[y_{j-1}]}$
 - If G_{j-1} is δ_{j-1} , $\delta_{j-1} \leftarrow \delta_{j-1} \vee \mu|_{\sigma[y_{j-1}]}$

Repairing “guess”-ed candidate Skolem functions

- Every model of $\mu(X, y_1, \dots, y_{j-1})$ gives a problematic combination of G_1, \dots, G_{j-1} values
- Flip G_{j-1} whenever μ holds
 - Recall $G_{j-1} \in \{\neg\gamma_{j-1}, \delta_{j-1}\}$
 - **Only** source of error: under-approximation of $\neg\exists y_j, \dots, \exists y_m \varphi(X, y_1, \dots, y_{j-2}, y_{j-1}, y_j, \dots, y_m)$
 - Repair: **Expand under-approximation**
 - If G_{j-1} is $\neg\gamma_{j-1}$, $\gamma_{j-1} \leftarrow \gamma_{j-1} \vee \mu|_{\sigma[y_{j-1}]}$
 - If G_{j-1} is δ_{j-1} , $\delta_{j-1} \leftarrow \delta_{j-1} \vee \mu|_{\sigma[y_{j-1}]}$

Counter-example guided repair by expanding δ_i 's and γ_i 's.

Repairing “guess”-ed candidate Skolem functions

- Every model of $\mu(X, y_1, \dots, y_{j-1})$ gives a problematic combination of G_1, \dots, G_{j-1} values
- Flip G_{j-1} whenever μ holds
 - Recall $G_{j-1} \in \{\neg\gamma_{j-1}, \delta_{j-1}\}$
 - **Only** source of error: under-approximation of $\neg\exists y_j, \dots, \exists y_m \varphi(X, y_1, \dots, y_{j-2}, y_{j-1}, y_j, \dots, y_m)$
 - Repair: **Expand under-approximation**
 - If G_{j-1} is $\neg\gamma_{j-1}$, $\gamma_{j-1} \leftarrow \gamma_{j-1} \vee \mu|_{\sigma[y_{j-1}]}$
 - If G_{j-1} is δ_{j-1} , $\delta_{j-1} \leftarrow \delta_{j-1} \vee \mu|_{\sigma[y_{j-1}]}$

Counter-example guided repair by expanding δ_i 's and γ_i 's.

Expansion-based repair

Repairing “guess”-ed candidate Skolem functions

- Every model of $\mu(X, y_1, \dots, y_{j-1})$ gives a problematic combination of G_1, \dots, G_{j-1} values
- Flip G_{j-1} whenever μ holds
 - Recall $G_{j-1} \in \{\neg\gamma_{j-1}, \delta_{j-1}\}$
 - **Only** source of error: under-approximation of $\neg\exists y_j, \dots, \exists y_m \varphi(X, y_1, \dots, y_{j-2}, y_{j-1}, y_j, \dots, y_m)$
 - Repair: **Expand under-approximation**
 - If G_{j-1} is $\neg\gamma_{j-1}$, $\gamma_{j-1} \leftarrow \gamma_{j-1} \vee \mu|_{\sigma[y_{j-1}]}$
 - If G_{j-1} is δ_{j-1} , $\delta_{j-1} \leftarrow \delta_{j-1} \vee \mu|_{\sigma[y_{j-1}]}$

Counter-example guided repair by expanding δ_i 's and γ_i 's.

Expansion-based repair

Simple argument for termination – expansions can't go on forever

Can we always find μ ?

Can we always find μ ?

Yes!!!

Can we always find μ ?

Yes!!!

- $\mu_{\text{worst-case}}$: $\text{ClausalForm}(\sigma[X, y_1, \dots, y_{m-1}])$
 - $\sigma \models \mu_{\text{worst-case}}$

Can we always find μ ?

Yes!!!

- $\mu_{\text{worst-case}}$: $\text{ClausalForm}(\sigma[X, y_1, \dots, y_{m-1}])$
 - $\sigma \models \mu_{\text{worst-case}}$
 - $\mu_{\text{worst-case}} \Rightarrow \gamma_m \wedge \delta_m$

Can we always find μ ?

Yes!!!

- $\mu_{\text{worst-case}}$: $\text{ClausalForm}(\sigma[X, y_1, \dots, y_{m-1}])$
 - $\sigma \models \mu_{\text{worst-case}}$
 - $\mu_{\text{worst-case}} \Rightarrow \gamma_m \wedge \delta_m$

... Why?

Can we always find μ ?

Yes!!!

- $\mu_{\text{worst-case}}$: $\text{ClausalForm}(\sigma[X, y_1, \dots, y_{m-1}])$
 - $\sigma \models \mu_{\text{worst-case}}$
 - $\mu_{\text{worst-case}} \Rightarrow \gamma_m \wedge \delta_m$... Why?
 - $G_m \in \{\varphi|_{y_m=1}, \neg\varphi|_{y_m=0}\}$ **always correct** Skolem function

Can we always find μ ?

Yes!!!

- $\mu_{\text{worst-case}}$: $\text{ClausalForm}(\sigma[X, y_1, \dots, y_{m-1}])$
 - $\sigma \models \mu_{\text{worst-case}}$
 - $\mu_{\text{worst-case}} \Rightarrow \gamma_m \wedge \delta_m$... Why?
 - $G_m \in \{\varphi|_{y_m=1}, \neg\varphi|_{y_m=0}\}$ **always correct** Skolem function
 - And yet, $\varphi(\sigma[X, y_1, \dots, y_{m-1}], G_m(\sigma[X, y_1, \dots, y_{m-1}])) = 0$

Can we always find μ ?

Yes!!!

- $\mu_{\text{worst-case}}$: $\text{ClausalForm}(\sigma[X, y_1, \dots, y_{m-1}])$
 - $\sigma \models \mu_{\text{worst-case}}$
 - $\mu_{\text{worst-case}} \Rightarrow \gamma_m \wedge \delta_m$... Why?
 - $G_m \in \{\varphi|_{y_m=1}, \neg\varphi|_{y_m=0}\}$ **always correct** Skolem function
 - And yet, $\varphi(\sigma[X, y_1, \dots, y_{m-1}], G_m(\sigma[X, y_1, \dots, y_{m-1}])) = 0$
- Choose smallest j s.t. $\sigma \models \delta_j \wedge \gamma_j$

Can we always find μ ?

Yes!!!

- $\mu_{\text{worst-case}}$: $\text{ClausalForm}(\sigma[X, y_1, \dots, y_{m-1}])$
 - $\sigma \models \mu_{\text{worst-case}}$
 - $\mu_{\text{worst-case}} \Rightarrow \gamma_m \wedge \delta_m$... Why?
 - $G_m \in \{\varphi|_{y_m=1}, \neg\varphi|_{y_m=0}\}$ **always correct** Skolem function
 - And yet, $\varphi(\sigma[X, y_1, \dots, y_{m-1}], G_m(\sigma[X, y_1, \dots, y_{m-1}])) = 0$
- Choose smallest j s.t. $\sigma \models \delta_j \wedge \gamma_j$

Repairing multiple Skolem functions

Can we always find μ ?

Yes!!!

- $\mu_{\text{worst-case}}$: $\text{ClausalForm}(\sigma[X, y_1, \dots, y_{m-1}])$
 - $\sigma \models \mu_{\text{worst-case}}$
 - $\mu_{\text{worst-case}} \Rightarrow \gamma_m \wedge \delta_m$... Why?
 - $G_m \in \{\varphi|_{y_m=1}, \neg\varphi|_{y_m=0}\}$ **always correct** Skolem function
 - And yet, $\varphi(\sigma[X, y_1, \dots, y_{m-1}], G_m(\sigma[X, y_1, \dots, y_{m-1}])) = 0$
- Choose smallest j s.t. $\sigma \models \delta_j \wedge \gamma_j$

Repairing multiple Skolem functions

Observation: $(\mu|_{y_{j-1}=0} \wedge \mu|_{y_{j-1}=1}) \Rightarrow \forall y_{j-1} \dots \forall y_m \neg\varphi$

Can we always find μ ?

Yes!!!

- $\mu_{\text{worst-case}}$: $\text{ClausalForm}(\sigma[X, y_1, \dots, y_{m-1}])$
 - $\sigma \models \mu_{\text{worst-case}}$
 - $\mu_{\text{worst-case}} \Rightarrow \gamma_m \wedge \delta_m$... Why?
 - $G_m \in \{\varphi|_{y_m=1}, \neg\varphi|_{y_m=0}\}$ **always correct** Skolem function
 - And yet, $\varphi(\sigma[X, y_1, \dots, y_{m-1}], G_m(\sigma[X, y_1, \dots, y_{m-1}])) = 0$
- Choose smallest j s.t. $\sigma \models \delta_j \wedge \gamma_j$

Repairing multiple Skolem functions

Observation: $(\mu|_{y_{j-1}=0} \wedge \mu|_{y_{j-1}=1}) \Rightarrow \forall y_{j-1} \dots \forall y_m \neg\varphi$

- Update μ to $\mu|_{y_{j-1}=0} \wedge \mu|_{y_{j-1}=1}$ and repeat with j decremented

Can we always find μ ?

Yes!!!

- $\mu_{\text{worst-case}}$: $\text{ClausalForm}(\sigma[X, y_1, \dots, y_{m-1}])$
 - $\sigma \models \mu_{\text{worst-case}}$
 - $\mu_{\text{worst-case}} \Rightarrow \gamma_m \wedge \delta_m$... Why?
 - $G_m \in \{\varphi|_{y_m=1}, \neg\varphi|_{y_m=0}\}$ **always correct** Skolem function
 - And yet, $\varphi(\sigma[X, y_1, \dots, y_{m-1}], G_m(\sigma[X, y_1, \dots, y_{m-1}])) = 0$
- Choose smallest j s.t. $\sigma \models \delta_j \wedge \gamma_j$

Repairing multiple Skolem functions

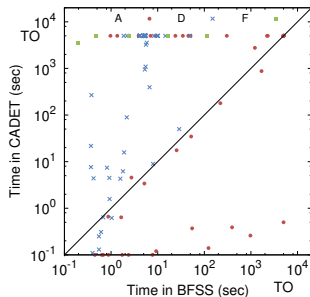
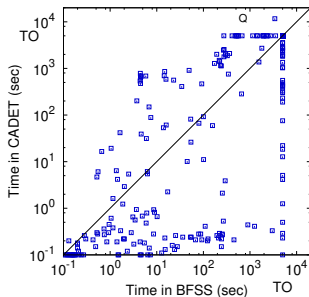
Observation: $(\mu|_{y_{j-1}=0} \wedge \mu|_{y_{j-1}=1}) \Rightarrow \forall y_{j-1} \dots \forall y_m \neg\varphi$

- Update μ to $\mu|_{y_{j-1}=0} \wedge \mu|_{y_{j-1}=1}$ and repeat with j decremented
- Can repair multiple Skolem functions starting from one counterexample

Experimental Comparison

BFSS (Cex-guided-repair) vis-a-vis CADET (incremental determinization) [Rabe & Seshia'16]

[Comparisons with other tools in FMSD 2020 paper]

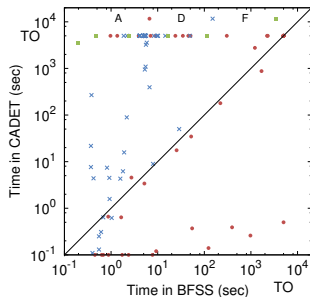
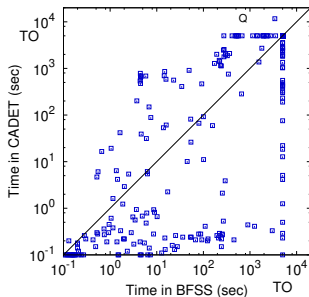


Q: QBFEval, A: Arithmetic, F: Factorization, D: Disjunctive Decomposition. TO: Timeout (3600 sec)

Experimental Comparison

BFSS (Cex-guided-repair) vis-a-vis CADET (incremental determinization) [Rabe & Seshia'16]

[Comparisons with other tools in FMSD 2020 paper]



Q: QBFEval, A: Arithmetic, F: Factorization, D: Disjunctive Decomposition. TO: Timeout (3600 sec)

- Mixed results: tools have orthogonal strengths
- Using CADET and BFSS as a portfolio solver sounds promising

Another Flavour of Guess-Check-Repair

Manthan: Golia et al [GMR'20]

Another Flavour of Guess-Check-Repair

Manthan: Golia et al [GMR'20]

- “Guess” using machine learning techniques

Another Flavour of Guess-Check-Repair

Manthan: Golia et al [GMR'20]

- “Guess” using machine learning techniques
 - Sample solutions of $\varphi(X, Y)$: **Data-driven!!!**

Another Flavour of Guess-Check-Repair

Manthan: Golia et al [GMR'20]

- “Guess” using machine learning techniques
 - Sample solutions of $\varphi(X, Y)$: Data-driven!!!
 - Learn “Decision Tree” for y_i in terms of X, y_1, \dots, y_{i-1}

Another Flavour of Guess-Check-Repair

Manthan: Golia et al [GMR'20]

- “Guess” using machine learning techniques
 - Sample solutions of $\varphi(X, Y)$: Data-driven!!!
 - Learn “Decision Tree” for y_i in terms of X, y_1, \dots, y_{i-1}
 - Ok to not match data exactly
 - Can have 2-sided errors!
 - Repair learnt decision trees later!
 - Often results in small decision trees

Another Flavour of Guess-Check-Repair

Manthan: Golia et al [GMR'20]

- “Guess” using machine learning techniques
 - Sample solutions of $\varphi(X, Y)$: Data-driven!!!
 - Learn “Decision Tree” for y_i in terms of X, y_1, \dots, y_{i-1}
 - Ok to not match data exactly
 - Can have 2-sided errors!
 - Repair learnt decision trees later!
 - Often results in small decision trees
- “Check” using error formula $\varepsilon(X, Y, Y')$ as before

Another Flavour of Guess-Check-Repair

Manthan: Golia et al [GMR'20]

- “Guess” using machine learning techniques
 - Sample solutions of $\varphi(X, Y)$: Data-driven!!!
 - Learn “Decision Tree” for y_i in terms of X, y_1, \dots, y_{i-1}
 - Ok to not match data exactly
 - Can have 2-sided errors!
 - Repair learnt decision trees later!
 - Often results in small decision trees
- “Check” using error formula $\varepsilon(X, Y, Y')$ as before
- “Repair” using unsatisfiable cores

Another Flavour of Guess-Check-Repair

Manthan: Golia et al [GMR'20]

- “Guess” using machine learning techniques
 - Sample solutions of $\varphi(X, Y)$: Data-driven!!!
 - Learn “Decision Tree” for y_i in terms of X, y_1, \dots, y_{i-1}
 - Ok to not match data exactly
 - Can have 2-sided errors!
 - Repair learnt decision trees later!
 - Often results in small decision trees
- “Check” using error formula $\varepsilon(X, Y, Y')$ as before
- “Repair” using unsatisfiable cores
 - $\sigma \models \varepsilon(X, Y, Y')$
 - Candidates to repair: F_i s.t. $y_i \neq Y'_i$

Manthan: Golia et al [GMR'20]

- “Guess” using machine learning techniques
 - Sample solutions of $\varphi(X, Y)$: **Data-driven!!!**
 - Learn “Decision Tree” for y_i in terms of X, y_1, \dots, y_{i-1}
 - Ok to not match data exactly
 - Can have 2-sided errors!
 - Repair learnt decision trees later!
 - Often results in small decision trees
- “Check” using error formula $\varepsilon(X, Y, Y')$ as before
- “Repair” using unsatisfiable cores
 - $\sigma \models \varepsilon(X, Y, Y')$
 - Candidates to repair: F_i s.t. $y_i \neq Y'_i$
 - $R_i(X, Y) = \varphi(X, Y) \wedge \bigwedge_{i=1}^n (x_i \Leftrightarrow \sigma[x_i]) \wedge (y_i \Leftrightarrow \sigma[Y'_i])$

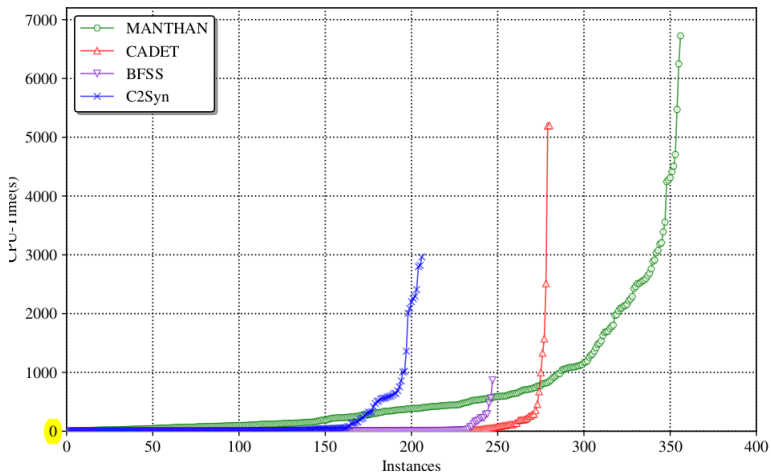
Manthan: Golia et al [GMR'20]

- “Guess” using machine learning techniques
 - Sample solutions of $\varphi(X, Y)$: **Data-driven!!!**
 - Learn “Decision Tree” for y_i in terms of X, y_1, \dots, y_{i-1}
 - Ok to not match data exactly
 - Can have 2-sided errors!
 - Repair learnt decision trees later!
 - Often results in small decision trees
- “Check” using error formula $\varepsilon(X, Y, Y')$ as before
- “Repair” using unsatisfiable cores
 - $\sigma \models \varepsilon(X, Y, Y')$
 - Candidates to repair: F_i s.t. $y_i \neq Y'_i$
 - $R_i(X, Y) = \varphi(X, Y) \wedge \bigwedge_{i=1}^n (x_i \Leftrightarrow \sigma[x_i]) \wedge (y_i \Leftrightarrow \sigma[Y'_i])$
 - Find repair formula β from Unsat Core of $R_i(X, Y)$

Manthan: Golia et al [GMR'20]

- “Guess” using machine learning techniques
 - Sample solutions of $\varphi(X, Y)$: **Data-driven!!!**
 - Learn “Decision Tree” for y_i in terms of X, y_1, \dots, y_{i-1}
 - Ok to not match data exactly
 - Can have 2-sided errors!
 - Repair learnt decision trees later!
 - Often results in small decision trees
- “Check” using error formula $\varepsilon(X, Y, Y')$ as before
- “Repair” using unsatisfiable cores
 - $\sigma \models \varepsilon(X, Y, Y')$
 - Candidates to repair: F_i s.t. $y_i \neq Y'_i$
 - $R_i(X, Y) = \varphi(X, Y) \wedge \bigwedge_{i=1}^n (x_i \Leftrightarrow \sigma[x_i]) \wedge (y_i \Leftrightarrow \sigma[Y'_i])$
 - Find repair formula β from Unsat Core of $R_i(X, Y)$
 - Update
 - F_i to $F_i \vee \beta$ if $\sigma[y_i] = 0$
 - F_i to $F_i \wedge \neg\beta$ if $\sigma[y_i] = 1$

Some more experimental comparisons [GMR20]



[Courtesy “Manthan: A Data Driven Approach to Boolean Functional Synthesis”,
Golia et al, CAV 2020]

- Boolean functional synthesis has diverse applications, including in temporal synthesis

Conclusion

- Boolean functional synthesis has diverse applications, including in temporal synthesis
- Guess-check-repair: a powerful paradigm

- Boolean functional synthesis has diverse applications, including in temporal synthesis
- Guess-check-repair: a powerful paradigm
- Different approaches to guessing possible
 - Based on approximations of quantifier elimination
 - Based on machine learning

- Boolean functional synthesis has diverse applications, including in temporal synthesis
- Guess-check-repair: a powerful paradigm
- Different approaches to guessing possible
 - Based on approximations of quantifier elimination
 - Based on machine learning
- Different approaches to counterexample-guided repair possible
 - Expansion of under-approximations
 - Unsatisfiable core based repair

- Boolean functional synthesis has diverse applications, including in temporal synthesis
- Guess-check-repair: a powerful paradigm
- Different approaches to guessing possible
 - Based on approximations of quantifier elimination
 - Based on machine learning
- Different approaches to counterexample-guided repair possible
 - Expansion of under-approximations
 - Unsatisfiable core based repair
- Recent results (BFSS, Manthan) extremely promising!!!