# Boolean Functional Synthesis and its Applications

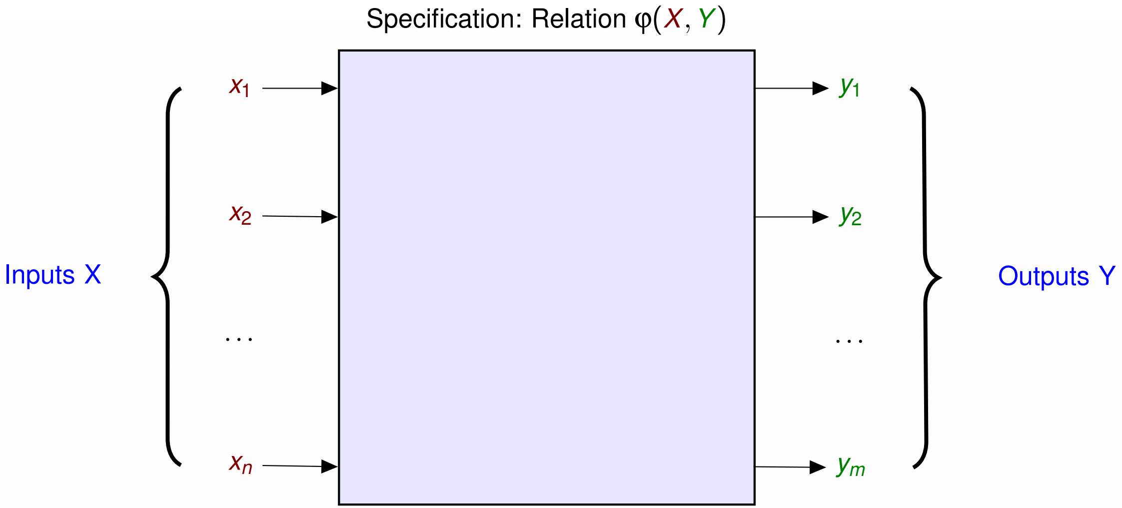
Priyanka Golia [1,2]

Joint work with: Kuldeep S. Meel [1] and Subhajit Roy [1]

[1]National University of Singapore
[2]Indian Institute of Technology Kanpur

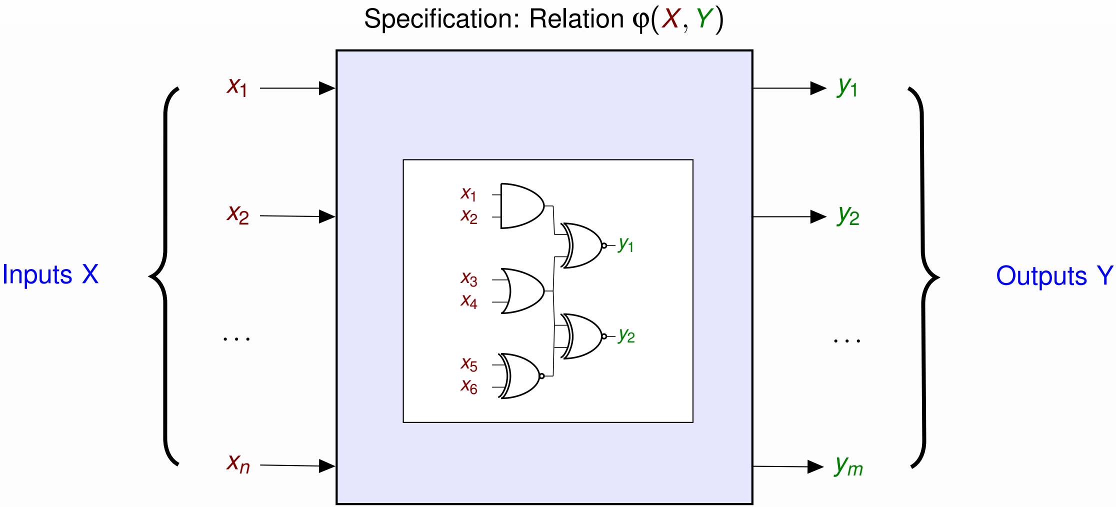
Corresponding Papers: CAV 2020, IJCAI 2021, ICCAD 2021 (Best Paper Award Nomination)

# Synthesis



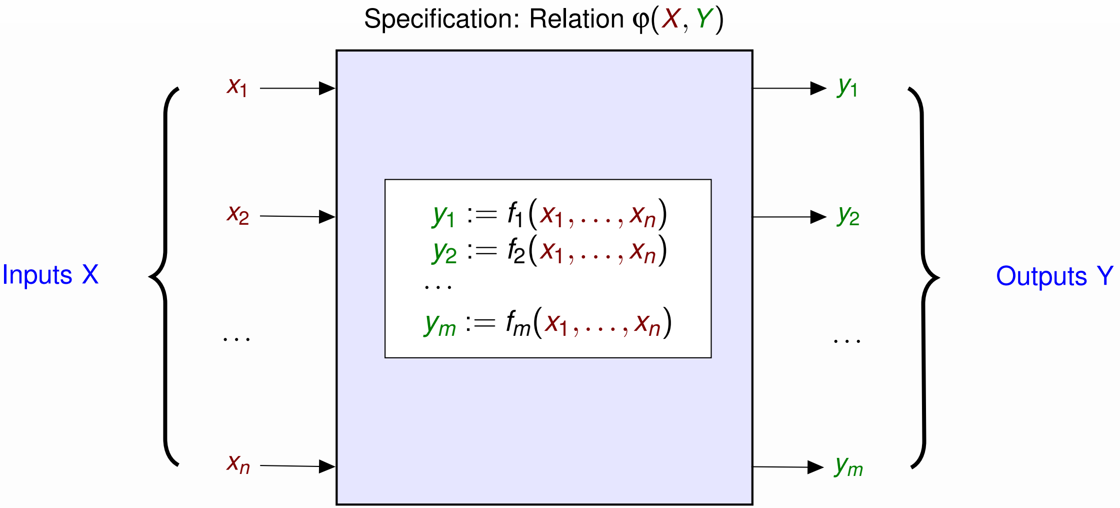
Specification: Relation φ(X, Y)
Inputs X
$x_1$
$x_2$
...
$x_n$
$y_1$
$y_2$
...
$y_m$
Outputs Y

# Synthesis



Specification: Relation φ(X, Y)
Inputs X
x1
x2
. . .
xn
Outputs Y
y1
y2
. . .
ym
x1
x2
x3
x4
x5
x6
y1
y2

# Synthesis


Specification: Relation $\varphi(X, Y)$
Inputs X
$x_1$
$x_2$
$\dots$
$x_n$
$y_1 := f_1(x_1, \dots, x_n)$
$y_2 := f_2(x_1, \dots, x_n)$
$\dots$
$y_m := f_m(x_1, \dots, x_n)$
$y_1$
$y_2$
$\dots$
$y_m$
Outputs Y

# Functional Synthesis

**Given** $\varphi(X, Y)$ over inputs $X = \{x_1, x_2, \ldots, x_n\}$ and outputs $Y = \{y_1, y_2, \ldots, y_m\}$.

**Synthesize** A function vector $F = \{f_1, f_2, \ldots, f_m\}$, such that $y_i := f_i(x_1, \ldots, x_n)$ such that:

$$\exists Y \varphi(X, Y) \equiv \varphi(X, F(X))$$

Each $f_i$ is called Skolem function and $F$ is called Skolem function vector.

Key Challenge: $\varphi(X, Y)$ is a relation

# Non-uniqueness of Skolem Functions

Let $X = \{x_1, x_2\}$, $Y = \{y_1\}$ and $\varphi(X, Y) = x_1 \vee x_2 \vee y_1$

Possible Skolem function: $f(x_1, x_2) := \neg(x_1 \vee x_2)$

# Non-uniqueness of Skolem Functions

Let $X = \{x_1, x_2\}$, $Y = \{y_1\}$ and $\varphi(X, Y) = x_1 \vee x_2 \vee y_1$

Possible Skolem function: $f(x_1, x_2) := \neg(x_1 \vee x_2)$

$$\varphi(X, F(X)) = x_1 \vee x_2 \vee (\neg(x_1 \vee x_2))$$

| $X$ | $\exists Y \varphi(X, Y)$ | | $\varphi(X, F(X))$ |
|---|---|---|---|
| $x_1 = 0, x_2 = 0$ | $y_1 = 1$ | True | True |
| $x_1 = 0, x_2 = 1$ | $y_1 = 1$ | True | True |
| $x_1 = 1, x_2 = 0$ | $y_1 = 1$ | True | True |
| $x_1 = 1, x_2 = 1$ | $y_1 = 1$ | True | True |

$$\exists Y \varphi(X, Y) \equiv \varphi(X, F(X))$$

# Non-uniqueness of Skolem Functions

Let $X = \{x_1, x_2\}$, $Y = \{y_1\}$ and $\varphi(X, Y) = x_1 \lor x_2 \lor y_1$

Possible Skolem function: $f(x_1, x_2) := \neg(x_1 \lor x_2)$

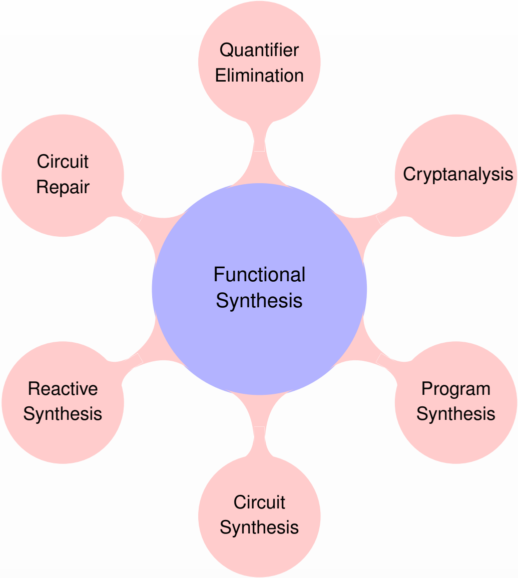
$$\varphi(X, F(X)) = x_1 \lor x_2 \lor (\neg(x_1 \lor x_2))$$

| $X$ | $\exists Y \varphi(X, Y)$ | | $\varphi(X, F(X))$ |
|---|---|---|---|
| $x_1 = 0, x_2 = 0$ | $y_1 = 1$ | True | True |
| $x_1 = 0, x_2 = 1$ | $y_1 = 1$ | True | True |
| $x_1 = 1, x_2 = 0$ | $y_1 = 1$ | True | True |
| $x_1 = 1, x_2 = 1$ | $y_1 = 1$ | True | True |

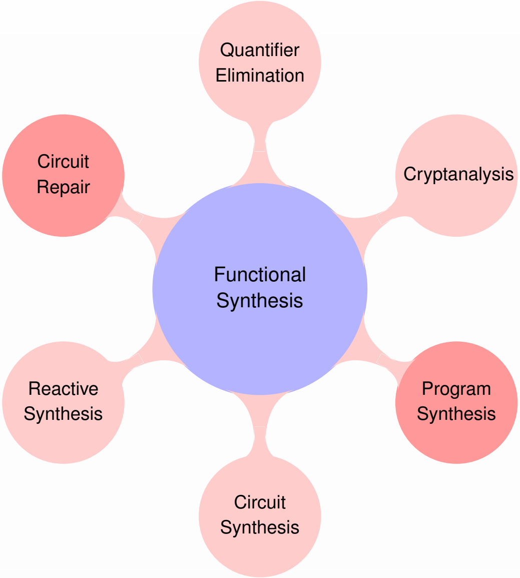
$\exists Y \varphi(X, Y) \equiv \varphi(X, F(X))$

Other possible Skolem functions: $f_1(x_1, x_2) = \neg x_1 \quad f_1(x_1, x_2) = \neg x_2 \quad f_1(x_1, x_2) = 1$

# Applications



# Applications



# Application Domain 1: Program Synthesis

Golia et al., IJCAI'21

$g(x_1, x_2) \geq x_1$ and
$g(x_1, x_2) \geq x_2$ and
$(g(x_1, x_2) == x_1$ or
$g(x_1, x_2) == x_2)$

- Synthesize program representing function $g$ that satisfies the specification.

# Application Domain 1: Program Synthesis

## Golia et al., IJCAI'21

$g(x_1, x_2) \geq x_1$ and
$g(x_1, x_2) \geq x_2$ and
$(g(x_1, x_2) == x_1$ or
$g(x_1, x_2) == x_2)$

$y_1 \geq x_1$ and
$y_1 \geq x_2$ and
$(y_1 == x_1$ or
$y_1 == x_2)$

- Synthesize program representing function $g$ that satisfies the specification.
- Replace every call of functions $g$ by a new variable $y_1$ in the specification.

$$\forall x_1, x_2 \, \exists y_1 \; \varphi(x_1, x_2, y_1)$$

# Application Domain 1: Program Synthesis

## Golia et al., IJCAI'21

$g(x_1, x_2) \geq x_1$ and
$g(x_1, x_2) \geq x_2$ and
$(g(x_1, x_2) == x_1$ or
$g(x_1, x_2) == x_2)$

$y_1 \geq x_1$ and
$y_1 \geq x_2$ and
$(y_1 == x_1$ or
$y_1 == x_2)$

- Synthesize program representing function $g$ that satisfies the specification.
- Replace every call of functions $g$ by a new variable $y_1$ in the specification.
- Works with appropriate caveats, e.g., outputs depend on all inputs.

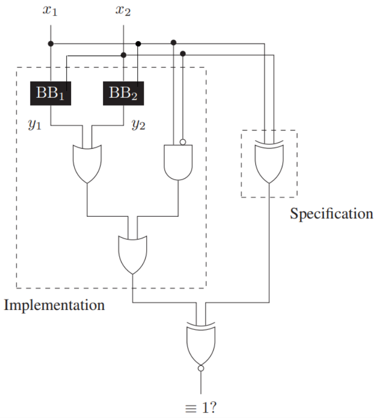
$$\forall x_1, x_2 \, \exists y_1 \; \varphi(x_1, x_2, y_1)$$

*The synthesized skolem function is an implementation of the function* $g(x_1, x_2)$.

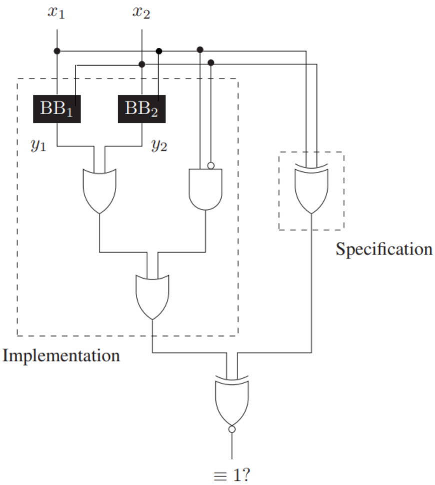
# Application Domain 2: Circuit Repair

- **Given:** An incomplete implementation and specification.
- **Objective:** Complete the implementation s.t. it is functionally equivalent to specification.

- **Given:** An incomplete implementation and specification.
- **Objective:** Complete the implementation s.t. it is functionally equivalent to specification.

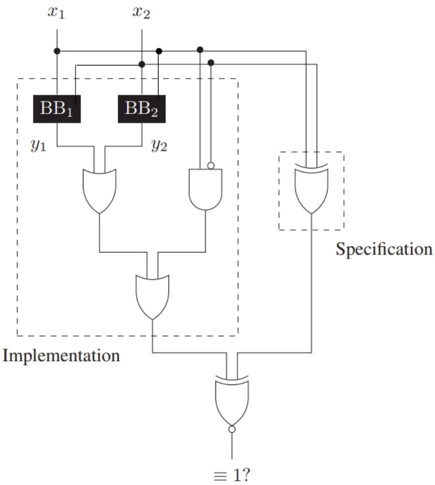


# Application Domain 2: Circuit Repair




- Inputs $x_1, x_2$, Outputs $y_1, y_2$.
- Synthesise functions(circuits) for $y_1, y_2$ such that it satisfy the given specification.

# Application Domain 2: Circuit Repair




- Inputs $x_1, x_2$, Outputs $y_1, y_2$.
- Synthesise functions(circuits) for $y_1, y_2$ such that it satisfy the given specification.

$$\forall x_1, x_2 \exists y_1 y_2 \neg(((y_1 \vee y_2) \vee (x_1 \wedge \neg x_2)) \oplus (x_1 \oplus x_2))$$

Image is taken(modified) from Equivalence Checking of Partial Designs Using Dependency Quantified Boolean Formulae, Gitina et al '13
Engineering change order for combinational and sequential design rectification, Jiang et. al'20
Synthesis and optimization of multiple portions of circuits for ECO based on set-covering and QBF formulations, Fujita et al'20

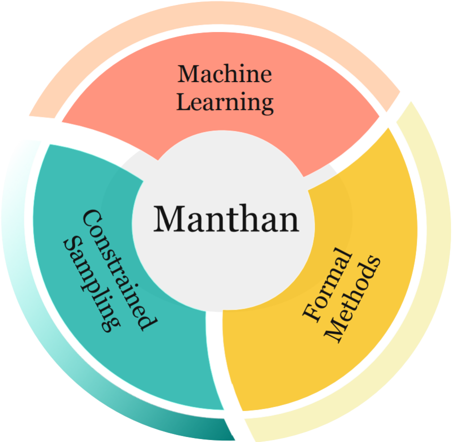
# Diverse Approaches

- From the proof of validity of $\forall X \exists Y \varphi(X, Y)$
  (Bendetti et al., 2005)
  (Jussilla et al., 2007)
  (Heule et al., 2014)
- Quantifier instantiation in SMT solvers
  (Barrett et al., 2015)
  (Bierre et al.,2017)
- Input-Output Separation
  (Chakraborty et al., 2018)
- Knowledge representation
  (Kukula et al., 2000 )
  (Trivedi et al., 2003)
  (Jiang, 2009)
  (Kuncak et al., 2010)
  (Balabanov and Jiang, 2011)
  (John et al., 2015)
  (Fried, Tabajara, Vardi, 2016,2017)
  (Akshay et al., 2017,2018)
  (Chakraborty et al., 2019)
- Incremental determinization
  (Rabe et al., 2015, 2018, 2019)

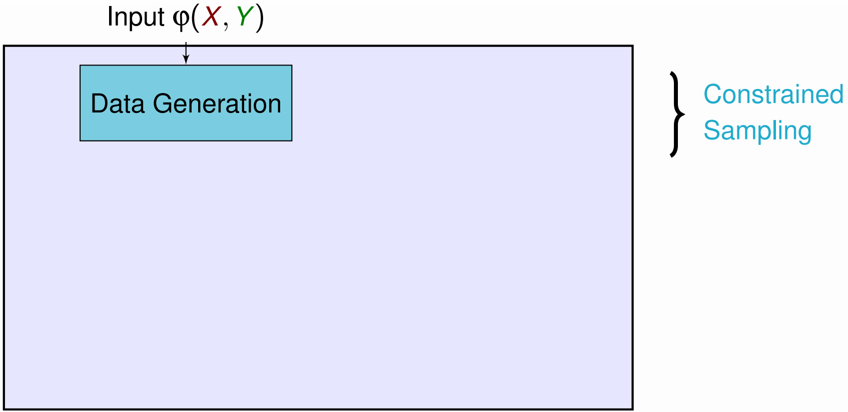
# Diverse Approaches

- From the proof of validity of $\forall X \exists Y \varphi(X, Y)$
  - (Bendetti et al., 2005)
  - (Jussilla et al., 2007)
  - (Heule et al., 2014)
- Quantifier instantiation in SMT solvers
  - (Barrett et al., 2015)
  - (Bierre et al.,2017)
- Input-Output Separation
  - (Chakraborty et al., 2018)
- Knowledge representation
  - (Kukula et al., 2000 )
  - (Trivedi et al., 2003)
  - (Jiang, 2009)
  - (Kuncak et al., 2010)
  - (Balabanov and Jiang, 2011)
  - (John et al., 2015)
  - (Fried, Tabajara, Vardi, 2016,2017)
  - (Akshay et al., 2017,2018)
  - (Chakraborty et al., 2019)
- Incremental determinization
  - (Rabe et al., 2015, 2018, 2019)

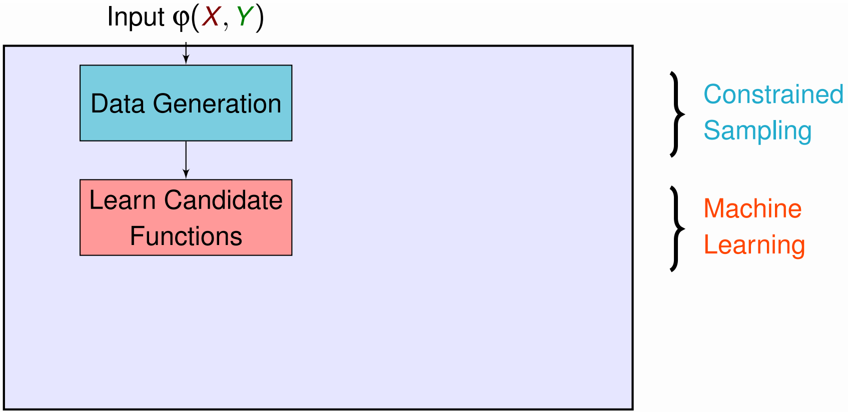
Scalability remains the holy grail


Machine Learning
Manthan
Constrained Sampling
Formal Methods

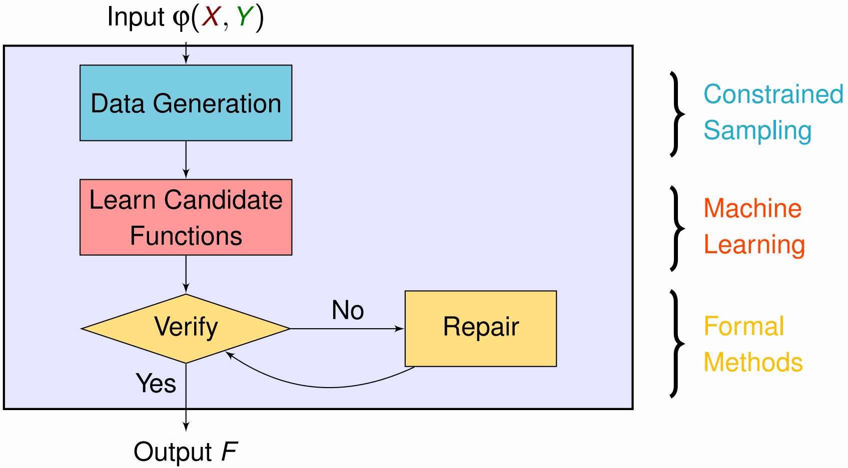
# Manthan



# Manthan



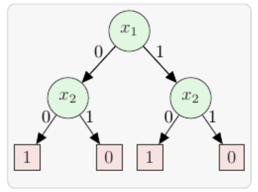
Input φ(X, Y)
Data Generation
Learn Candidate Functions
Constrained Sampling
Machine Learning


Input φ(X, Y)
Data Generation
Learn Candidate Functions
Verify
No
Repair
Yes
Output F
Constrained Sampling
Machine Learning
Formal Methods

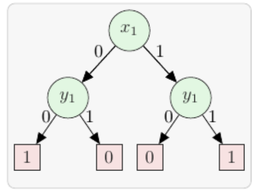
$\varphi(x_1, x_2, y_1, y_2)$ → 

| $x_1$ | $x_2$ | $y_1$ | $y_2$ |
|---|---|---|---|
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |

| $x_1$ | $x_2$ | $y_1$ | $y_2$ |
|---|---|---|---|
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |




$p_1 := (\neg x_1 \wedge \neg x_2)$,
$p_2 := (x_1 \wedge \neg x_2)$
$f_1$ = if $p_1$ then 1
elif $p_2$ then 1
else 0




$p_1 := (\neg x_1 \wedge \neg y_1)$,
$p_2 := (x_1 \wedge y_1)$
$f_2$ = if $p_1$ then 1
elif $p_2$ then 1
else 0

# Verification of Candidate Functions

## Reaping the Fruits of Formal Methods Revolution

$$E(X, Y, Y') := \varphi(X, Y) \wedge \neg\varphi(X, Y') \wedge (Y' \leftrightarrow F(X))$$

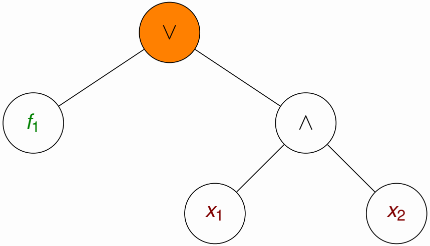
(JSCTA'15)

- If $E(X, Y, Y')$ is UNSAT: $\exists Y \varphi(X, Y) \equiv \varphi(X, F(X))$
  - Return $F$
- If $E(X, Y, Y')$ is SAT: $\exists Y \varphi(X, Y) \not\equiv \varphi(X, F(X))$
  - Let $\sigma \models E(X, Y, Y')$ be a counterexample to fix.

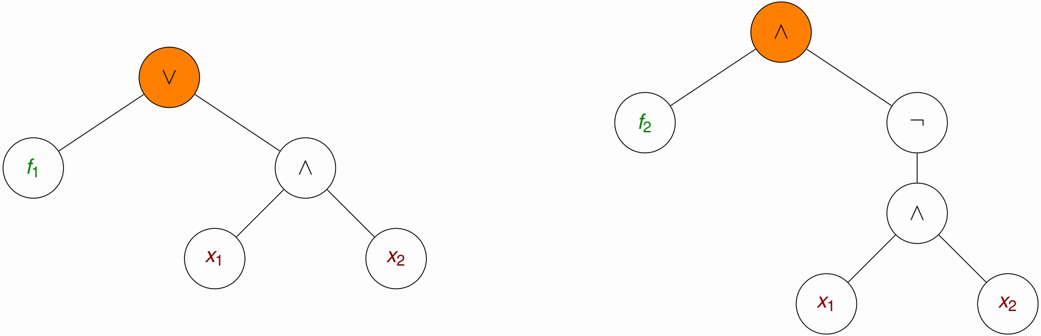
- $\sigma = \{x_1 \mapsto 1, x_2 \mapsto 1, y_1 \mapsto 1, y_2 \mapsto 0, y_1' \mapsto 0, y_2' \mapsto 1\}$.

- $\sigma = \{x_1 \mapsto 1, x_2 \mapsto 1, y_1 \mapsto 1, y_2 \mapsto 0, y_1' \mapsto 0, y_2' \mapsto 1\}$.
- Repair: If $\underbrace{x_1 \wedge x_2}_{\beta=\{x_1,x_2\}}$ then $y_1 = 1$

- $\sigma = \{x_1 \mapsto 1, x_2 \mapsto 1, y_1 \mapsto 1, y_2 \mapsto 0, y_1' \mapsto 0, y_2' \mapsto 1\}$.
- Repair: If $\underbrace{x_1 \wedge x_2}_{\beta=\{x_1,x_2\}}$ then $y_1 = 1$



- $\sigma = \{x_1 \mapsto 1, x_2 \mapsto 1, y_1 \mapsto 1, y_2 \mapsto 0, y_1' \mapsto 0, y_2' \mapsto 1\}$.
- Repair: If $\underbrace{x_1 \wedge x_2}_{\beta=\{x_1,x_2\}}$ then $y_1 = 1$



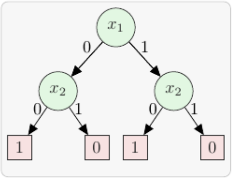
# Manthan

$\varphi(X, Y)$
$X = \{x_1, x_2\}$
$Y = \{y_1, y_2\}$

Data Generation →

| $x_1$ | $x_2$ | $y_1$ | $y_2$ |
|---|---|---|---|
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

Learn Candidates →




Verify Candidates ↓

Check Satisfiability of $E(X, Y, Y')$

SAT, $\sigma$ → $G_\sigma(X, Y)$

UNSAT Core-based Repair

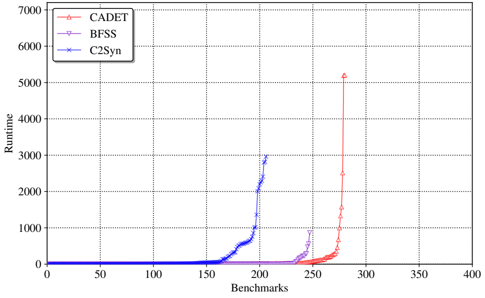
UNSAT ↓

Return F

# Experimental Evaluations

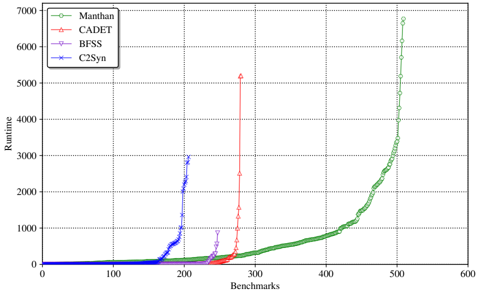
- 609 Benchmarks from:
  - QBFEval competition
  - Arithmetic
  - Disjunctive decomposition
  - Factorization
- Compared Manthan with State-of-the-art tools: CADET ( Rabe et al., 2019 ), BFSS (Akshay et al. ,2018), C2Syn (Chakraborty et al., 2019).
- Timeout: 7200 seconds.

# Experimental Evaluations




| C2Syn | BFSS | CADET |
|---|---|---|
| 206 | 247 | 280 |

# Experimental Evaluations




| C2Syn | BFSS | CADET | Manthan |
|---|---|---|---|
| 206 | 247 | 280 | 509 |

An increase of 223 benchmarks.

Manthan: A Data-Driven Approach for Boolean Functional Synthesis.

- Constrained Sampling
- Decision List Classifier
- Formal Methods

Solves 509 benchmarks — state of the art could solve 280

https://github.com/meelgroup/manthan

Thanks!