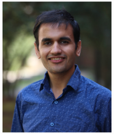
# Program Synthesis as Dependency Quantified Formula Modulo Theory

Priyanka Golia [1,2], Subhajit Roy [1] and Kuldeep S. Meel [2]

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

IJCAI 2021

Program Synthesis: *Given a specification as logical formula, synthesise a program that satisfies the specification.* (Church, 1957)

$$f_1(x_1, x_2) \geq 2 \times x_1 \text{ and}$$
$$f_1(x_1, x_2) \geq 2 \times x_2 \text{ and}$$
$$(f_1(x_1, x_2) == 2 \times x_1 \text{ or}$$
$$f_1(x_1, x_2) == 2 \times x_2)$$

Sythesise a function $f_1$
that satisfies the specification

**Program Synthesis:** *Given a specification as logical formula, synthesise a program that satisfies the specification.* (Church, 1957)

$$f_1(x_1, x_2) \geq 2 \times x_1 \text{ and}$$
$$f_1(x_1, x_2) \geq 2 \times x_2 \text{ and}$$
$$(f_1(x_1, x_2) == 2 \times x_1 \text{ or}$$
$$f_1(x_1, x_2) == 2 \times x_2)$$

→ Synthesiser

Sythesise a function $f_1$
that satisfies the specification

# Program Synthesis

Program Synthesis: *Given a specification as logical formula, synthesise a program that satisfies the specification.* (Church, 1957)

$f_1(x_1, x_2) \geq 2 \times x_1$ and
$f_1(x_1, x_2) \geq 2 \times x_2$ and
$(f_1(x_1, x_2) == 2 \times x_1$ or
$f_1(x_1, x_2) == 2 \times x_2)$

→ Synthesiser →

$f_1(input1, input2)$ {
  If $(input1 \geq input2)$
    Return $2 \times input1$
  Else
    Return $2 \times input2$ }

Sythesise a function $f_1$
that satisfies the specification

- Given:
  - A set of typed function symbols $\{f_1, f_2, \ldots f_k\}$.
  - A background theory $\mathbb{T}$.
  - A specification $\varphi$ over the vocabulary of $\mathbb{T} \cup \{f_1, f_2, \ldots f_k\}$
- Find:
  - A set of expressions $\{e_1, e_2, \ldots e_k\}$ such that the formula $\varphi[f_1/e_1, f_2/e_2, \ldots f_k/e_k]$ is valid modulo $\mathbb{T}$.

$f_1(x_1, x_2) \geq 2 \times x_1$ and
$f_1(x_1, x_2) \geq 2 \times x_2$ and
$(f_1(x_1, x_2) == 2 \times x_1$ or
$f_1(x_1, x_2) == 2 \times x_2)$

→ Synthesiser →

$f_1(input1, input2)$ {
If $(input1 \geq input2)$
Return $2 \times input1$
Else
Return $2 \times input2$ }

Function symbol: $f_1$(arg1,arg2)
$\mathbb{T}$: Linear Interger Arithmetic (LIA)
Specification $\varphi$

# Program Synthesis: $\mathbb{T}$-Constrained Synthesis

- Program synthesis is an exceptionally difficult problem.
- Applied to various domains:
  - program completion.
  - program optimization.
  - programming by example.
- Explored in pragmatic restrictions — reducing search space.

# Program Synthesis: Syntax Guided Synthesis

- Given:
  - A set of typed function symbols $\{f_1, f_2, \dots f_k\}$.
  - A background theory $\mathbb{T}$, and a grammar $\{L_1, \dots L_k\}$ over the vocabulary of $\mathbb{T}$.
  - A specification $\varphi$ over the vocabulary of $\mathbb{T} \cup \{f_1, f_2, \dots f_k\}$.
- Find:
  - The set of expressions $\{e_1 \in L_1, \dots e_k \in L_k\}$ s. t. formula $\varphi[f_1/e_1, \dots f_k/e_k]$ is valid modulo $\mathbb{T}$.

(Alur et al.,2013 )

$f_1(x_1, x_2) \geq 2 \times x_1$ and
$f_1(x_1, x_2) \geq 2 \times x_2$ and
$(f_1(x_1, x_2) == 2 \times x_1$ or
$f_1(x_1, x_2) == 2 \times x_2)$

S: T|if (C) then T else T
T: 0 | 1 | $x_1$ | $x_2$ | T + T
C: T $\leq$ T

→ Synthesiser →

$f_1(input1, input2)$ {
If ($input1 \geq input2$)
Return $input1 + input1$
Else
Return $input2 + input2$ }

# Dependency Quantified Formula

- Given a quantified formula $\phi$ in theory $\mathbb{T}$ with universal ($\forall$) and existential ($\exists$) quantifiers.
- Existentially quantified variables have explicit dependencies on a subset of universally quantified variables.

$$\phi := \forall x_1, \ldots, x_n \, \exists^{H_1} y_1 \ldots \exists^{H_m} y_m \, \varphi(x_1, \ldots, x_n, y_1, \ldots, y_m)$$

Where each $H_i \subseteq \{x_1, \ldots, x_n\}$.

- $\exists^{H_i}$ is called Henkin quantifier, and $H_i$ is called Henkin dependencies.

# Dependency Quantified Formula

$$\phi := \forall x_1, x_2 \, \exists^{H_1} y_1 \; \varphi(x_1, x_2, y_1)$$

Where $H_1 = \{x_1\}$, and $\varphi(x_1, x_2, y_1) := (x_1 \vee x_2 \vee y_1)$

- Does there exists a function $y_1 := g_1(x_1)$ such that $\varphi(x_1, x_2, g_1(x_1))$ is a tautology?

$$\phi := \forall x_1, x_2\ \exists^{H_1} y_1\ \varphi(x_1, x_2, y_1)$$

Where $H_1 = \{x_1\}$, and $\varphi(x_1, x_2, y_1) := (x_1 \vee x_2 \vee y_1)$

- Does there exists a function $y_1 := g_1(x_1)$ such that $\varphi(x_1, x_2, g_1(x_1))$ is a tautology?
- With $g_1(x_1) = \neg x_1$:

$$\varphi(x_1, x_2, g_1(x_1)) := x_1 \vee x_2 \vee (\neg x_1)$$

| $X$ | $\varphi(x_1, x_2, g_1(x_1))$ |
|---|---|
| $x_1 = 0, x_2 = 0$ | True |
| $x_1 = 0, x_2 = 1$ | True |
| $x_1 = 1, x_2 = 0$ | True |
| $x_1 = 1, x_2 = 1$ | True |

} Dependency Quantified Formula $\phi$ is True.

# Dependency Quantified Formula

- A formula $\phi$ is considered to be DQF($\mathbb{T}$), if it can be represented as $\forall x_1, \ldots x_n, \exists^{H_1} y_1 \ldots \exists^{H_m} y_m \varphi(x_1, \ldots, x_n, y_1, \ldots, y_m)$
- Variables $x_1, \ldots x_n, y_1 \ldots y_m$ and $\varphi(x_1, \ldots, x_n, y_1, \ldots, y_m)$ should be in underlying theory $\mathbb{T}$.
- A $DQF(\mathbb{T})$ formula is True, if there exists function a vector $\boldsymbol{g} : \langle g_1(H_1), \ldots, g_m(H_m) \rangle$ such that $\varphi(x_1, \ldots, x_n, g_1(H_1), \ldots, g_m(H_m))$ is a tautology.
- When $\mathbb{T} =$ Boolean: DQF($\mathbb{T}$) formula is considered as DQBF.

# Our Contributions

- Reduction of program synthesis to DQF($\mathbb{T}$).
- Reduction of DQF(BV) to DQBF — allows us to simply plug-in the state of the art DQBF solvers for BV-constrained synthesis.

# Program synthesis as DQF($\mathbb{T}$)

$f_1(x_1, x_2) \geq 2 \times x_1$ and
$f_1(x_1, x_2) \geq 2 \times x_2$ and
$(f_1(x_1, x_2) == 2 \times x_1$ or
$f_1(x_1, x_2) == 2 \times x_2)$

$f_2(x_3, x_4) \leq 2 \times x_3$ and
$f_2(x_3, x_4) \leq 2 \times x_4$ and
$(f_2(x_3, x_4) == 2 \times x_3$ or
$f_2(x_3, x_4) == 2 \times x_4)$

- Sythesise functions $f_1, f_2$ that satisfies the specification.

# Program synthesis as DQF($\mathbb{T}$)

$$f_1(x_1, x_2) \geq 2 \times x_1 \text{ and}$$
$$f_1(x_1, x_2) \geq 2 \times x_2 \text{ and}$$
$$(f_1(x_1, x_2) == 2 \times x_1 \text{ or}$$
$$f_1(x_1, x_2) == 2 \times x_2)$$

$$f_2(x_3, x_4) \leq 2 \times x_3 \text{ and}$$
$$f_2(x_3, x_4) \leq 2 \times x_4 \text{ and}$$
$$(f_2(x_3, x_4) == 2 \times x_3 \text{ or}$$
$$f_2(x_3, x_4) == 2 \times x_4)$$

$$y_1 \geq 2 \times x_1 \text{ and}$$
$$y_1 \geq 2 \times x_2 \text{ and}$$
$$(y_1 == 2 \times x_1 \text{ or}$$
$$y_1 == 2 \times x_2)$$

$$y_2 \leq 2 \times x_3 \text{ and}$$
$$y_2 \leq 2 \times x_4 \text{ and}$$
$$(y_2 == 2 \times x_3 \text{ or}$$
$$y_2 == 2 \times x_4)$$

- Sythesise functions $f_1, f_2$ that satisfies the specification.
- Replace every call of functions $f_1, f_2$ by new variables $y_1, y_2$ in specification.

# Program synthesis as DQF($\mathbb{T}$)

$y_1 \geq 2 \times x_1$ and
$y_1 \geq 2 \times x_2$ and
$(y_1 == 2 \times x_1$ or
$y_1 == 2 \times x_2)$

$y_2 \leq 2 \times x_3$ and
$y_2 \leq 2 \times x_4$ and
$(y_2 == 2 \times x_3$ or
$y_2 == 2 \times x_4)$

$$\phi := \forall x_1, x_2, x_3, x_4 \exists^{H_1} y_1 \exists^{H_2} y_2 \; \varphi(x_1, x_2, x_3, x_4, y_1, y_2)$$

Where $H_1 := \{x_1, x_2\}$ and $H_2 := \{x_3, x_4\}$.

# Program synthesis as DQF($\mathbb{T}$)

$y_1 \geq 2 \times x_1$ and
$y_1 \geq 2 \times x_2$ and
$(y_1 == 2 \times x_1$ or
$y_1 == 2 \times x_2)$

$y_2 \leq 2 \times x_3$ and
$y_2 \leq 2 \times x_4$ and
$(y_2 == 2 \times x_3$ or
$y_2 == 2 \times x_4)$

$$\phi := \forall x_1, x_2, x_3, x_4 \exists^{H_1} y_1 \exists^{H_2} y_2\ \varphi(x_1, x_2, x_3, x_4, y_1, y_2)$$

Where $H_1 := \{x_1, x_2\}$ and $H_2 := \{x_3, x_4\}$.

- DQF($\mathbb{T}$) solvers find function vector $\langle g_1(x_1, x_2), g_2(x_3, x_4)\rangle$ such that $\varphi(x_1, x_2, x_3, x_4, g_1(x_1, x_2), g_2(x_3, x_4))$ is a tautology.

# Program synthesis as DQF($\mathbb{T}$)

$y_1 \geq 2 \times x_1$ and
$y_1 \geq 2 \times x_2$ and
$(y_1 == 2 \times x_1$ or
$y_1 == 2 \times x_2)$

$y_2 \leq 2 \times x_3$ and
$y_2 \leq 2 \times x_4$ and
$(y_2 == 2 \times x_3$ or
$y_2 == 2 \times x_4)$

$$\phi := \forall x_1, x_2, x_3, x_4 \exists^{H_1} y_1 \exists^{H_2} y_2 \, \varphi(x_1, x_2, x_3, x_4, y_1, y_2)$$

Where $H_1 := \{x_1, x_2\}$ and $H_2 := \{x_3, x_4\}$.

- DQF($\mathbb{T}$) solvers find function vector $\langle g_1(x_1, x_2), g_2(x_3, x_4) \rangle$ such that $\varphi(x_1, x_2, x_3, x_4, g_1(x_1, x_2), g_2(x_3, x_4))$ is a tautology.
- $g_1(x_1, x_2)$ required set of expression for $f_1(input1, input2)$.
- $g_2(x_3, x_4)$ required set of expression for $f_2(input1, input2)$.

# Multiple CallSign: Program synthesis as DQF($\mathbb{T}$)

$f_1(x_1, x_2) \geq 2 \times x_1$ and
$f_1(x_1, x_2) \geq 2 \times x_2$ and
$(f_1(x_1, x_2) == 2 \times x_1$ or
$f_1(x_1, x_2) == 2 \times x_2)$

- Henkin dependencies for $y_1$ are $x_1$ and $x_2$.

$$\phi := \forall x_1, x_2 \exists^{H_1} y_1 \; \varphi(x_1, x_2, y_1)$$

  Where $H_1 := \{x_1, x_2\}$.

- DQF($\mathbb{T}$) solvers find function vector $\langle g_1(x_1, x_2) \rangle$ such that $\varphi(x_1, x_2, g_1(x_1, x_2))$ is a tautology.

# Multiple CallSign: Program synthesis as DQF($\mathbb{T}$)

$$f_1(x_1, x_2) \geq 2 \times x_1 \text{ and}$$
$$f_1(x_2, x_3) \geq 2 \times x_3 \text{ and}$$
$$(f_1(x_1, x_3) == 2 \times x_1 \text{ or}$$
$$f_1(x_1, x_3) == 2 \times x_3)$$

- **Multiple CallSigns**: Not every call of function $f_1$ have same set of arguments.

$f_1(input1, input2)$ {
  If ($input1 \geq input2$)
    Return $2 \times input1$
  Else
    Return $2 \times input2$ }

# Multiple CallSign: Program synthesis as DQF($\mathbb{T}$)

$f_1(x_1, x_2) \geq 2 \times x_1$ and
$f_1(x_2, x_3) \geq 2 \times x_3$ and
$(f_1(x_1, x_3) == 2 \times x_1$ or
$f_1(x_1, x_3) == 2 \times x_3)$

- Replace every instance of $f_1$ by same variable: $y_1 := f_1(x_1, x_2)$ and $y_1 := f_1(x_2, x_3)$...
  $H_1 = \{x_1, x_2, x_3\}$ Not Correct!!

$f_1(x_1, x_2) \geq 2 \times x_1$ and
$f_1(x_2, x_3) \geq 2 \times x_3$ and
$(f_1(x_1, x_3) == 2 \times x_1$ or
$f_1(x_1, x_3) == 2 \times x_3)$

- Replace every instance of $f_1$ by same variable: $y_1 := f_1(x_1, x_2)$ and $y_1 := f_1(x_2, x_3)$...
  $H_1 = \{x_1, x_2, x_3\}$ Not Correct!!
- Different variable $y$ variables for every instance of $f_1$ ?

$$y_1 := f_1(x_1, x_2), y_2 := f_1(x_2, x_3), y_3 := f_1(x_1, x_3)$$

- But, we want to synthesise a Single function that satisfies All constraints.

# Multiple CallSign: Program synthesis as DQF($\mathbb{T}$)

$f_1(x_1, x_2) \geq 2 \times x_1$ and
$f_1(x_2, x_3) \geq 2 \times x_3$ and
$(f_1(x_1, x_3) == 2 \times x_1$ or
$f_1(x_1, x_3) == 2 \times x_3)$

- Replace every instance of $f_1$ by same variable: $y_1 := f_1(x_1, x_2)$ and $y_1 := f_1(x_2, x_3)$...
  $H_1 = \{x_1, x_2, x_3\}$ Not Correct!!
- Different variable $y$ variables for every instance of $f_1$ ?

$$y_1 := f_1(x_1, x_2), y_2 := f_1(x_2, x_3), y_3 := f_1(x_1, x_3)$$

- But, we want to synthesise a Single function that satisfies All constraints.

We need to take care of Multiple CallSign instances to reduce to DQF($\mathbb{T}$)

# Multiple CallSign: Program synthesis as DQF($\mathbb{T}$)

- Introduce function arity many new variables.
- Here a,b are newly introduced variables.

$f_1(x_1, x_2) \geq 2 \times x_1$ and
$f_1(x_2, x_3) \geq 2 \times x_3$ and
$(f_1(x_1, x_3) == 2 \times x_1$ or
$f_1(x_1, x_3) == 2 \times x_3)$

If $(x_1 == a)$ and $(x_2 == b)$ then
$f_1(x_1, x_2) == f_1(a, b)$
If $(x_2 == a)$ and $(x_3 == b)$ then
$f_1(x_2, x_3) == f_1(a, b)$
If $(x_1 == a)$ and $(x_3 == b)$ then
$f_1(x_1, x_3) == f_1(a, b)$

# Multiple CallSign: Program synthesis as DQF($\mathbb{T}$)

- Introduce function arity many new variables.
- Here a,b are newly introduced variables.

$y_1 \geq 2 \times x_1$ and
$y_2 \geq 2 \times x_3$ and
$(y_3 == 2 \times x_1$ or
$y_3 == 2 \times x_3)$

If $(x_1 == a)$ and $(x_2 == b)$ then
$y_1 == y_4$
If $(x_2 == a)$ and $(x_3 == b)$ then
$y_2 == y_4$
If $(x_1 == a)$ and $(x_3 == b)$ then
$y_3 == y_4$

- Have different $Y$ variables for every function instances.
  $y_1 := f_1(x_1, x_2), y_2 := f_1(x_2, x_3),$
  $y_3 := f_1(x_1, x_3), y_4 := f_1(a, b)$

  $$\forall X\, \exists^{H_1} y_1 \exists^{H_2} y_2 \exists^{H_3} y_3 \exists^{H_4} y_4 \varphi(X, Y)$$

  Where $H_1 = \{x_1, x_2\}, H_2 = \{x_2, x_3\},$
  $H_3 = \{x_1, x_3\}, H_4 = \{a, b\}$
- $g_4(a, b)$ required set of expression for $f_1(input1, input2)$.

# Experimental Evaluations

- Objective: Does DQBF solvers perform on par with state-of-the-art program synthesis tools?
- We compared Syntax guided synthesis tools (SyGuS) tools, DQBF tools over 645 instances from SyGuS competitions.

| Syntax-Guided | DQBF-based |
|---|---|
| CVC4,ESolver<br>EUSolver,DryadSynth<br>Stochpp | CADET, DCAQE<br>Manthan, DepQBF<br>DQBDD |

# Experimental Evaluations

- Number of SyGuS instances solved using different techniques.
- Timeout. 900s.

| | Total | SyGuS-tools | DQBF-based |
|---|---|---|---|
| SyGuS Instances | 645 | 513 | 610 |

# Experimental Evaluations

- Number of SyGuS instances solved using different techniques.
- Timeout. 900s.

| | Total | SyGuS-tools | DQBF-based |
|---|---|---|---|
| SyGuS Instances | 645 | 513 | 610 |

- DQBF solvers performs better than the syntax-guided synthesis.

# Conclusion

- Reduction of program synthesis to DQF($\mathbb{T}$).
- The special case, $\mathbb{T} = BV$ can further be converted to DQBF instances.
- The general purpose DQBF solvers performs better than the syntax guided synthesis.

`https://github.com/meelgroup/DeQuS`

Thanks!