Lecture 22

Monday, April 13, 2020 4:58 PM

- Hello!
- Link to today's shared Google Doc

https://docs.google.com/document/d/1A_O7eL12kLEVyB351bNihyI9zg1Uc658yYYqpIraAH8/edit?usp=sharing

---

## Unit 4: Program Synthesis

---

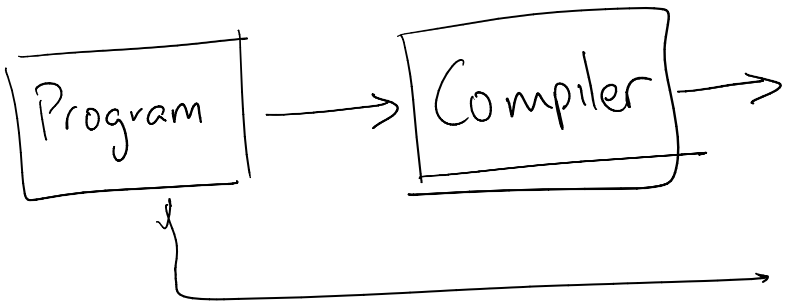
- Computer-Augmented Program Engineering

- How do we help non-programmers to write code?
- Excel: Spreadsheet functions
  + VBScript.
- Can we liberate the programmer from tedium & low-level details?
  - Might not want to write code
    know how to write code

- The program is not the goal.
  - Examples — can we program by examples? (PBE)

---

- Compiler internals.



Machine Code

x86, ARM

1000s of instructions.

Lots of processor-specific trivia.

AVX512

SSE 2.1

⋮

MMX

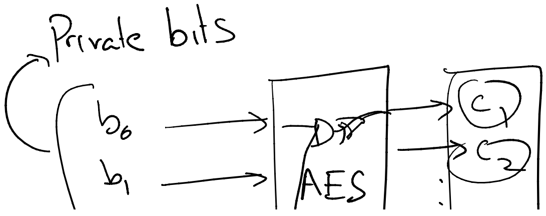
- Semantics have to be equivalent.
- Target has to be high performance
- Compiler has to be fast.

- Can we automate the job of a compiler optimizer?

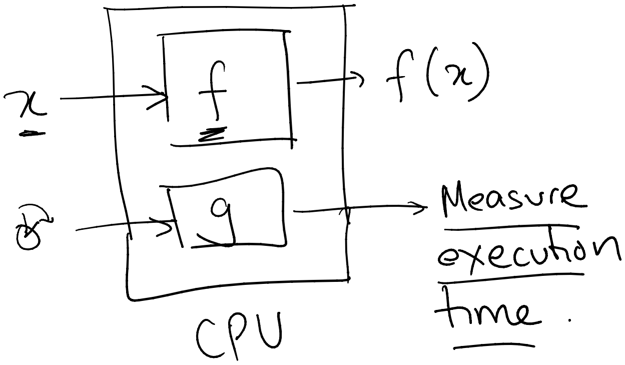
---

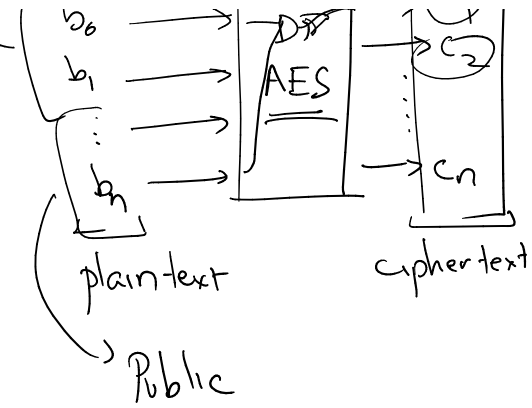
- Can we automatically build secure systems?
  - Side channel leaks

  Privacy violations



Privacy violations





- Timing delays in cryptographic circuits can leak sensitive information.

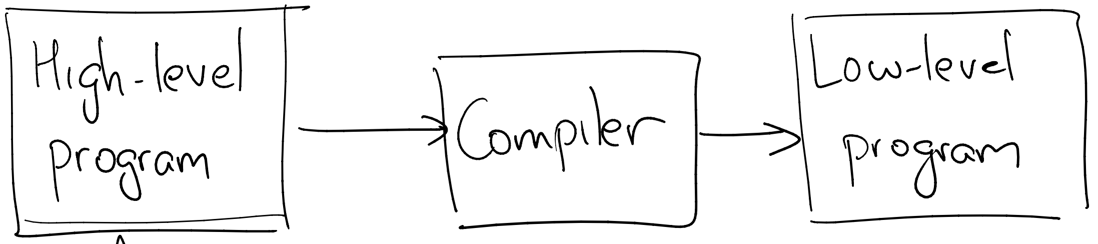
  Prof. Chao Wang

- Synthesis Question: Given an AES implementation that is written in a human-readable form, can we design an equivalent implementation which is free of timing-based side channels?

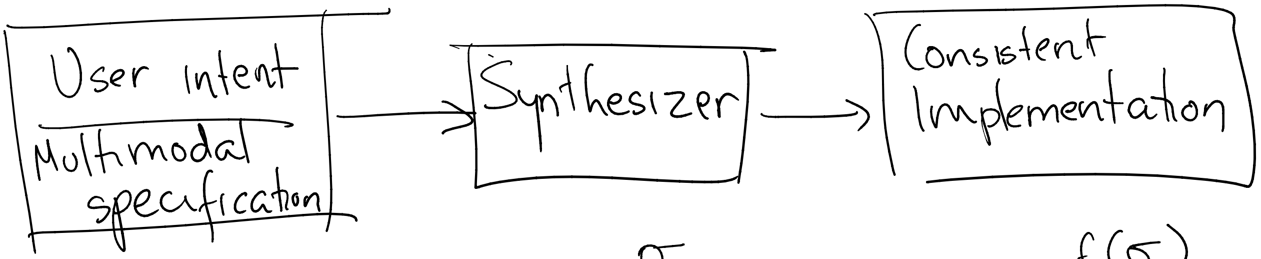
---

- Programming-by-Example / FlashFill
- Compiler (super) optimizations / STOKE  sketch.

- Computer security / side channel leaks.



Complete descriptions of the function to be computed.



- Input-output examples

$\overset{\sigma}{\text{"Mukund Raghothaman"}} \mapsto \overset{f(\sigma)}{\text{"MR"}}$

"Bart Simpson" ↦ "BS"

- Logical formulas — $\forall \sigma, |f(\sigma)| \leq |\sigma|$.

- Snippets of implementation — If there are no spaces in the name, simply output the first character.

Traditional programming

Unimodal specifications

Unimodal specifications
High-level complete program ⟹ Low-level complete program.

## Renaissance of program synthesis

(2006 —— 2011 ———— )

↑ Sketch ↑ FlashFill

### Enabling technologies

- Mature constraint solving technology
  SAT / SMT solvers
- Growth in CPU power
- Better algorithms
- Better HCI.

## Outline of ideas

- How to specify programs.
  - Syntax-Guided Synthesis (SyGuS)

- Syntax-Guided Synthesis (SyGuS)

User Intent = Semantic specification
+ Syntactic specification.

- Algorithms for Synthesis.
  - Version spaces
  - Counter-example Guided Inductive Synthesis (CEGIS)
    - Enumerative
    - Constraint-based
    - Stochastic solvers

      Monte Carlo methods.

  - Deductive program synthesis

    Use axioms + rewrite rules.

    Programmer Expression Graph. / Equality saturation.

- Applications
  - Program termination
  - Invariant generation
  - Flash Fill

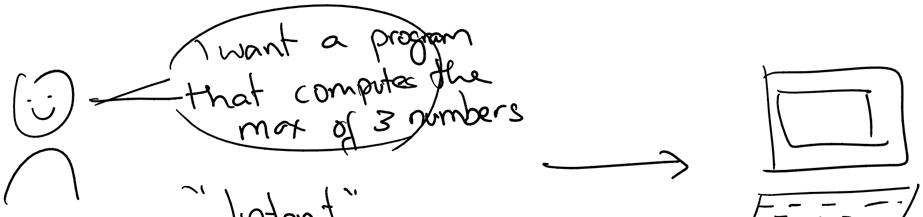
---

## Syntax-Guided Synthesis (SyGuS)

SyGuS = Semantic Specification
+ Syntactic Specification.

---

<u>Ex</u>: maximum of 3 numbers



"Intent"

$\Downarrow$

Formalize their intent.

$\forall \vec{x}$
$\varphi(\vec{x}, f(\vec{x}))$

$$\forall x\, y\, z,\ f(x, y, z) \geq x$$
$$\text{and}\quad f(x, y, z) \geq y$$

$\forall x$

$\varphi(\vec{x}, f(\vec{x}))$

and $f(x,y,z) \geq y$

and $f(x,y,z) \geq z$

and $(f(x,y,z) = x$ or
$f(x,y,z) = y$ or
$f(x,y,z) = z)$.

Synthesizer returns a $\overset{\text{program}}{\text{function}}$ f

s.t $\forall x, y, z, \varphi(x,y,z, f(x,y,z))$.

Syntactic constraint.

Target language may allow linear integer arithmetic, + branching on linear inequalities

```
f(x,y,z) = if (x ≤ y) {
               if (y ≤ z) z else y
           } else {
               if (x ≤ z) z else x
           }
```

Ex 2: max2

$$\varphi = \forall x\, y \;\; f(x\; y) \geq x$$
$$\text{and} \;\; f(x, y) \geq y$$
$$\text{and} \;\; (f(x,y) = x \;\; \text{or} \;\; f(x,y) = y).$$

Target does not have if statements,

- but it can compute the absolute value of an expression. $|\underline{\underline{e}}|$.

- Add, subtract, multiply, divide numbers.

$$\boxed{f(x,y) = \frac{x + y + |x - y|}{2}}$$

$f(0,0) = 0$ ✓

$f(0,1) = \frac{1 + |-1|}{2} = 1$ ✓

$f(1,0) = \frac{1 + |1|}{2} = 1$ ✓

$f(3,8) = \frac{3+8+|-5|}{2} = 8$ ✓

$f(8,3) = \frac{8+3+|5|}{2} = 8$ ✓

$f(x,y) = f(y,x)$.

WLOG, consider $f(x,y)$ with $x \geq y$.

In this case,

$$x - y \geq 0$$
$$|x-y| = (x-y)$$
$$f(x,y) = \frac{x + \cancel{y} + x - \cancel{y}}{2}$$
$$= x.$$

- Syntactic constraints expressed as context free grammars

IntExpr ::= 0 | 1 | …<br>
| $x$ | $y$ | $z$<br>
| $\text{IntExpr}_1 + \text{IntExpr}_2$<br>
| $\text{IntExpr}_1 - \text{IntExpr}_2$<br>
| if (Bool Expr) { $\text{IntExpr}_1$ } else { $\text{IntExpr}_2$ }

Bool Expr ::= true | false<br>
| $\text{IntExpr}_1 \leq \text{IntExpr}_2$<br>
| …<br>
| $\text{BoolExpr}_1$ and $\text{BoolExpr}_2$<br>
| $\text{BoolExpr}_1$ or $\text{BoolExpr}_2$<br>
| not Bool Expr

$$\forall x, y, z \;\; \underline{\varphi(x, y, z, \underline{f(x, y, z)})}$$

$$\exists x, y, z \;\; \overline{\varphi}(x, y, z, f(x, y, z))$$

SAT

- Plug candidate $f$ into $\varphi$.
- Check if $\overline{\varphi}$ is satisfiable
- If sat, counter-example found. $f$ does not work.
- Otherwise, $f$ works.

$\Rightarrow \underline{0}$ × $(x=1, y=1, z=1)$

$\Rightarrow \underline{1}$ × $(x=2, y=2, z=2)$

$\Rightarrow \underline{x}$ × $(x=1, y=2, z=2)$

$\underline{y}$ ✓

$\underline{z}$ ×

$0+0$ ×

$0+1$ ×

$1+0$ ×

$1+1$ ×

$\underline{x+1}$ ×

$\underline{x-y}$

⋮

if $(x \le y)$ {$x$} else {$y$}

⋮

```
if(x≤y) { if (y ≤ z){...}
          else{...} }
else {...}
```

---

Ex: $\forall x\, y \quad f(x\, y) \geq x$ and $f(x,y) \geq y$ ~~and (_ or _)~~

Spec: $\forall x\, y \quad f(x\, y) \geq x$ and $f(x\, y) \geq y$

Candidate program: $f(x\, y) = \underline{x+y}$.

Goal: TST: $\forall x\, y \quad \underline{(x+y) \geq x \text{ and } (x+y) \geq y}$

Ask machine: $\underline{\exists x\, y}$ s.t not $(x+y \geq x$ and $x+y \geq y)$ ?

SMT — ① Formula is UNSAT.

Candidate program works.

② Formula is SAT.

Satisfying assignment is a counter-example.

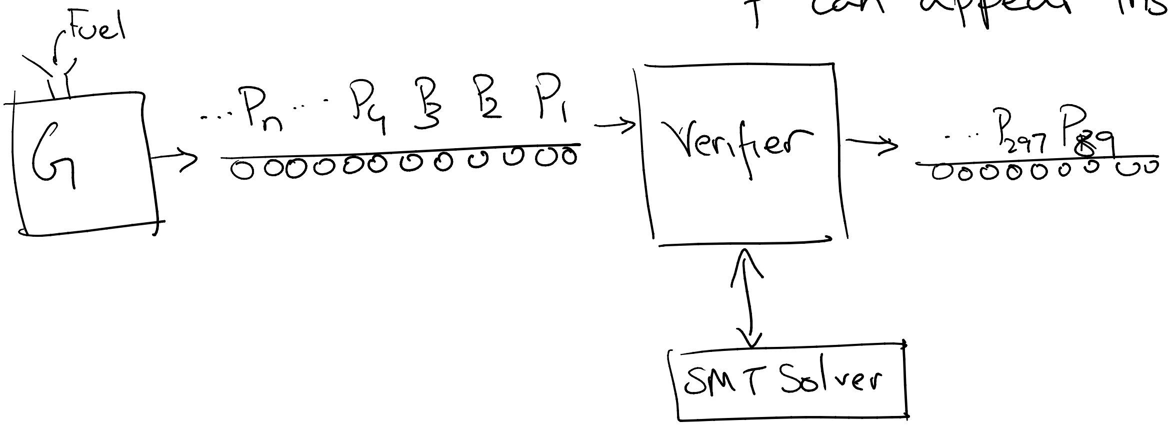
---

SyGuS problem instance

# SyGuS problem instance

Find expression $f$ from context-free grammar $G$ such that $\forall x, y, \ldots,\ \varphi(x, y, \ldots)$.

$f$ can appear inside $\varphi$.


Fuel
G
$\cdots P_n \cdots P_4\ P_3\ P_2\ P_1$
Verifier
$\cdots P_{297}\ P_{89}$
SMT Solver