Lecture 24

Monday, April 20, 2020 5:01 PM

- Hello!

- Assignment 3 is on the way

  Abstract interpretation + Program synthesis

- Assignment 2 was due on Friday.

Discussing Question 4 of Assignment 2

- Given a conjunction φ of literals drawn from the theory of EUF, to find whether φ is satisfiable.

  $\varphi = (t_1 = t_2)$ and $(t_3 = t_4)$ and $\underbrace{f(g(f(a)))}_{\text{All Terms}}$ (a: variable; f, g: UF symbol)

  $(t_5 \neq t_6)$ and ...

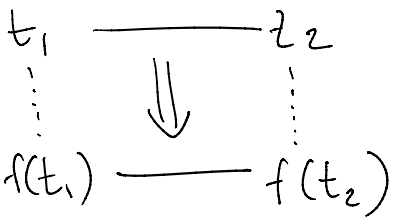
  $T = \mathbb{N}$

- Construct a graph $G = (V\ E)$ as follows.

  - Vertices V = all terms which appear in φ.

    $f(f(g(a)))$ $\xrightarrow{\text{also include}}$ $f(g(a)), g(a), a$

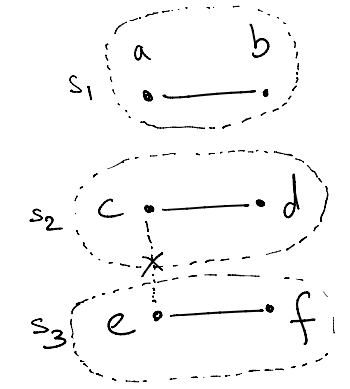
- Draw edges as follows:
  - For each equality literal $t_1 = t_2$ in φ draw the edge $t_1 \text{ —— } t_2$
  - Until fixpoint:
    For each edge $t_1 \text{ —— } t_2$,
    if $f(t_1)$ & $f(t_2)$ are also in V,
    draw $f(t_1) \text{ —— } f(t_2)$



- Check, $\forall$ inequality literals $t_1 \neq t_2$, whether t_1 & t_2 occur in the same SCC of G. If so, unsat.
  Otherwise, sat.
  - If T is finite, you might run out of values
  - So, not necessarily sat if T is finite.

Ex:

$\varphi = (a=b)$ and $(c=d)$ and $(c \neq e)$ and $(e=f)$.



$k = 3$ (# of SCCs)

$T = \mathbb{N}$ (infinite case) sat

$T = \{1,2,3\}$ sat

$T = \{1, 2\}^{a,b,c,d}_{e,f}$ sat

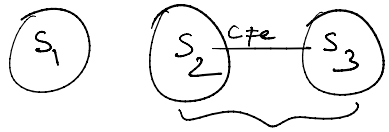
$T = \{1\}$ (because $c \neq e$) unsat

$T = \emptyset$ (no values) unsat

- Observation: If no inequivalence predicates, then satisfiable iff $|T| \geq 1$.

Graph of inequivalence constraints G_2

-



Value of s_2 cannot be the same as value of s_3.

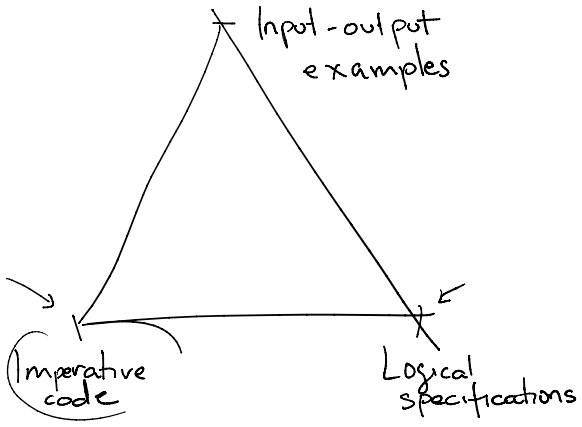
- Formula φ is satisfiable iff G_2 can be coloured with n colours.

- NP-complete.

- Congruence closure when $T = \mathbb{N}$ can be solved in PTIME
- Congruence closure when $T = \{1, 2, \ldots, n\}$ is NP-complete.

- Program Synthesis

- Continuum of specifications



- How do SyGuS solvers work?

CEGIS: Counter-Example Guided Inductive Synthesis.

SyGuS ——— CEGIS

↑ Name of the problem

↓ Sorting

↑ Solution methodology

↓ QuickSort

SyGuS problem

- Given grammar G

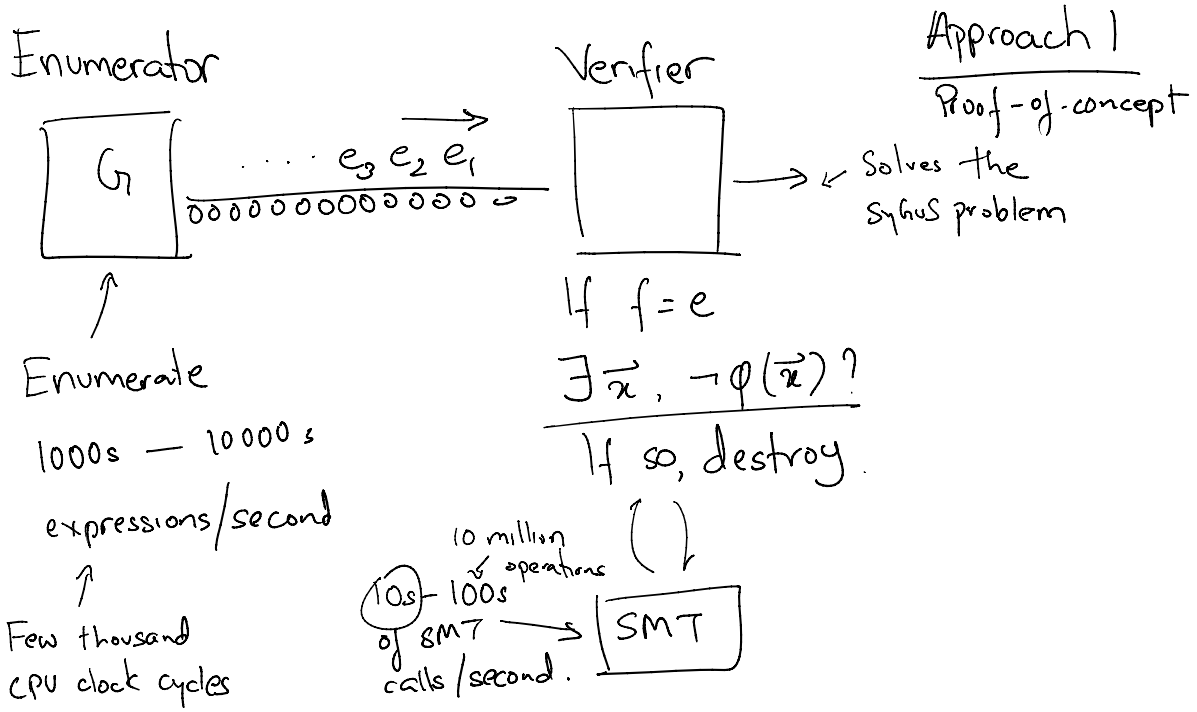
  constraints φ

  find f in G which satisfies the constraints.

- Given grammar G, specification $\varphi(\vec{x})$

find f such that

$\forall \vec{x}, \varphi(\vec{x})$ holds.



- Given grammar G, specification $\varphi(\vec{x})$ Approach 2

find f such that

$\forall \vec{x}, \varphi(\vec{x})$ holds.

$\forall x \in \mathbb{N}, \varphi(x)$ holds.

$\Rightarrow$ In particular,

$\varphi(0), \varphi(1), \varphi(3), \ldots$

$\wedge$ $\vee$

$\forall$ $\exists$

$\varphi(0), \varphi(1), \varphi(3), \ldots$ ∀ ∃

In particular, $\forall x \in$ Finite Set, $\varphi(x)$ holds

$\{0, 1, 3, 8, 11, 12\} \subseteq \mathbb{N}$

$\varphi(0) \wedge \varphi(1) \wedge \varphi(3) \wedge \cdots \wedge \varphi(12)$. holds

---

Original SyGuS problem ①
- Given grammar G, specification $\varphi(x)$
  find f such that
  $\forall x \in \mathbb{N}, \varphi(x)$ holds.

②

Found Finite Set

$C = \{1, 2, 3, 12\} \subseteq \mathbb{N}$

Problem Version 2.0 ③
- Given grammar G, specification $\varphi(x)$
  find f s.t.
  $\forall x \in C, \varphi(x)$ holds
  $\forall x \in \mathbb{N}, \varphi(x)$ holds

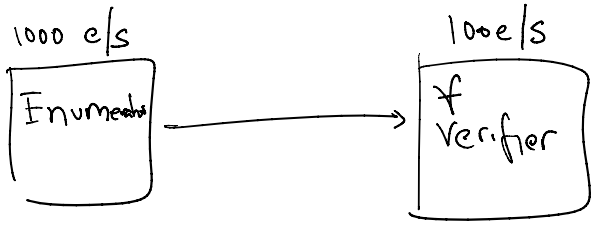
$C = \{x_1, x_2 \cdots x_k\}$

$\varphi(x_1) \wedge \varphi(x_2) \wedge \cdots \wedge \varphi(x_k)$

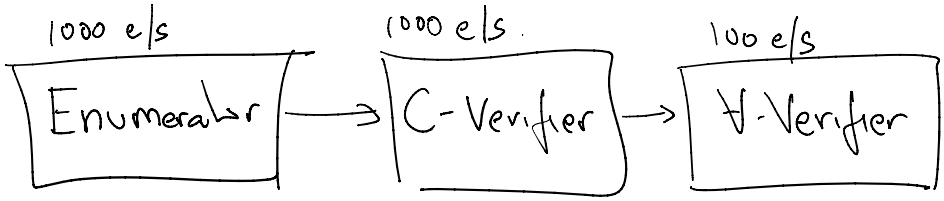
↑ Can be trivially evaluated w/o an SMT solver.

The SyGuS Expression Factory

Factory Version 1.0 :

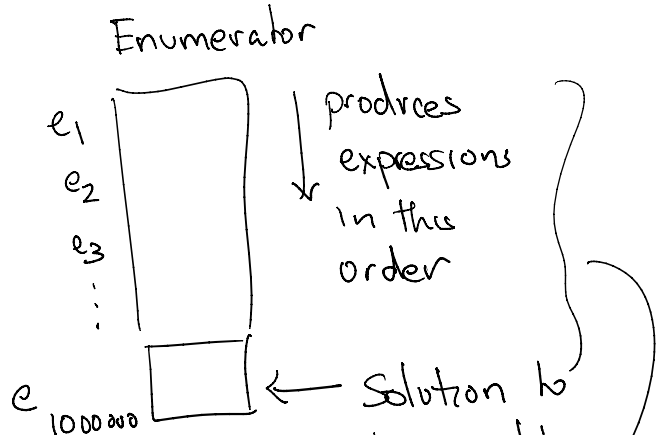


Factory Version 2.0 :



$\forall x \in C$, $\varphi(x)$ holds

$\varphi(x_1) \wedge \varphi(x_2) \wedge \cdots \wedge \varphi(x_k)$



Assume that 10000 expressions are C-safe.

Factory Version 1.0 will take 10000 s to solve the problem.

Factory Version 2.0 will solve the problem in 100 s.

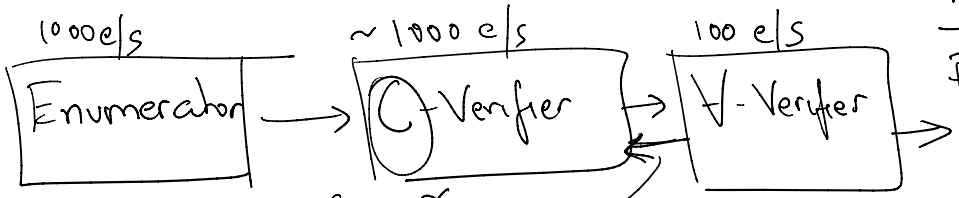
Approach 3

1000 e/s ~1000 e/s 100 e/s

Approach 3

Fixing C

Factory Version 3.0



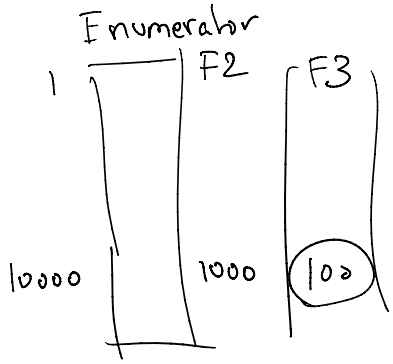
$C := \emptyset$

$C := C \cup \{x_0\}$

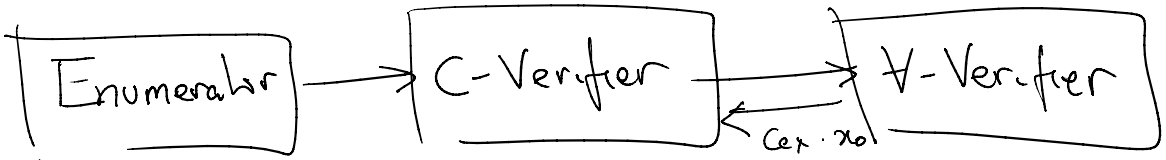
Back-pressure

If e fails, there is a counterexample, x_0

& e is destroyed.



Factory 3.0



CEGIS