# Lecture 14

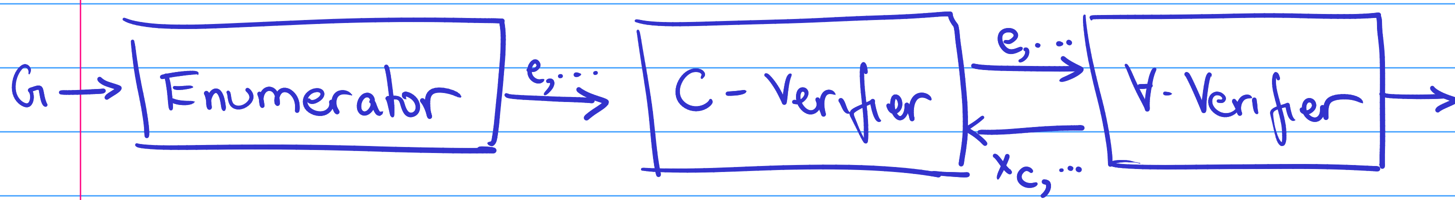
$$e \longrightarrow \boxed{\text{C-Verifier}} \longrightarrow e$$

$$\forall x \in \underline{C}\ \ \varphi(x \quad f)$$

$$\rightsquigarrow\ \forall x \in \underline{C \cup \{x_c\}}\ \ \varphi(x \quad f)$$

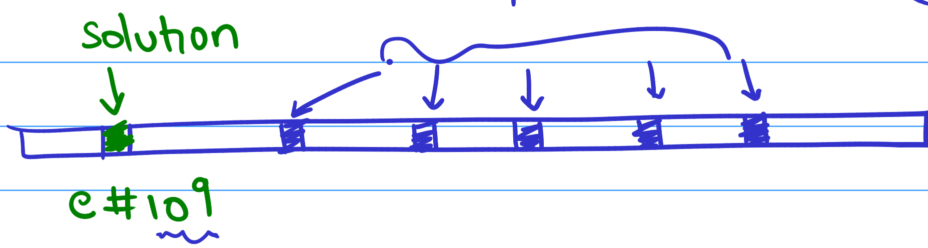
---

# The SyGuS Factory Version 2.0




$10^6$ e/s $\qquad$ $10^5$ e/s $\qquad$ $10^3$ e/s

False positives wrt C (# = $10^6$)



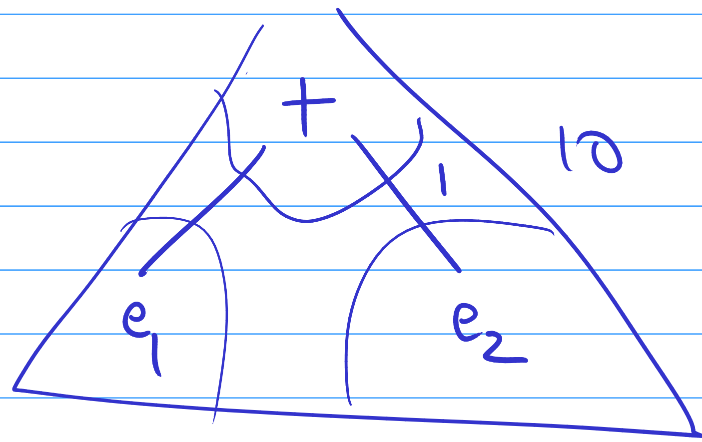

$$\text{Final running time} = \frac{10^9}{10^6} + \frac{10^9}{10^5} + \frac{10^6}{10^3}$$

$$= 1000 + 10000 + 1000$$

$$= 12000 \text{ seconds.}$$

- How to produce all expressions $e \in G$ of size, $|e| = 10$ ?

- Pick a production rule, say $e ::= e_1 + e_2$




$\forall\ n_1\ n_2$ s.t. $n_1 + n_2 = 9$

$\forall\ e_1 \in G^{n_1}\quad \forall\ e_2 \in G^{n_2}$

emit $e_1 + e_2$

$$G = \boxed{\text{Start} ::= 0 \mid 1 \mid x \mid y \mid \text{Start} + \text{Start} \mid \text{Start} - \text{Start}}$$

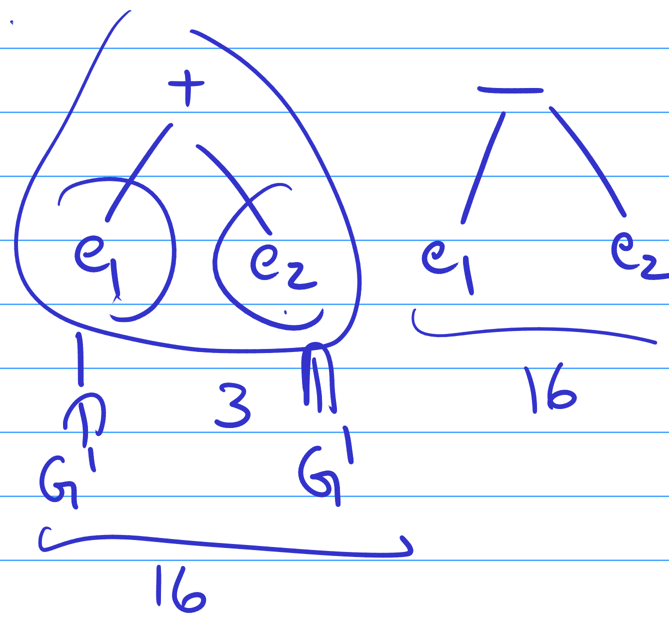
$G^0 = \emptyset$

All expressions of size 0

$G^1 = \{ 0 \quad 1 \quad x \quad y \}$

$G^2 = \emptyset$

$G^3 = \{ 0+0, \ 0+1, \ 0+x, \ 0+y, \ 1+0, \ 1+1, \ 1+x, \ 1+y, \ x+0, \ x+1, \ x+x, \ x+y, \ y+0, \ y+1, \ y+x, \ y+y \} \cup \{ \ldots \}$




$|G^3| = 32$

Expressions of size 3 with minus at the top.

<u>Observation</u>: Of the first 20 expressions emitted by the enumerator, 10 expressions are "useless".

semantically equivalent to something already emitted.

<u>Question</u>: How do you find if $e_1$ & $e_2$ are semantically equivalent?

$\exists x \quad e_1 \neq e_2$ ?

semantically inequiv

Yes

SMT

No

Semantically equiv.

Deduplicating

G → Enumerator → Cache → C-Verifier ⇄ V-Verifier →

SMT

**Axiom**: $e_1 + e_2 = e_2 + e_1$

Equality

Rewriting rule

$$x + (z + y) \rightsquigarrow (x + y) + Z$$

"Equality Saturation"

Proposed for compiler optimization

Who gives us these axioms?

Fundamentally limited in its ability to prove things.

---

**Question**: Semantic equivalence necessarily involves an SMT solver. Is there a "poor man's" semantic equivalence?

set of counterexamples

<u>Idea:</u> Use "$C$" as a proxy for checking equivalence.

$e_1 = x + y$ vs. $y + x = e_2$

$C = \{(x \mapsto 3,\ y \mapsto 9),\ (x \mapsto 8,\ y \mapsto 2)\}$

$e_1 = \boxed{12}$ $e_1 = \boxed{10}$

$e_2 = \boxed{12}$ $e_2 = \boxed{10}$

$e_1$ & $e_2$ "seem" equivalent

$e_1$ & $e_2$ are indistinguishable wrt $C$.

Indistinguishability $\Longleftarrow$ Semantic Equivalence

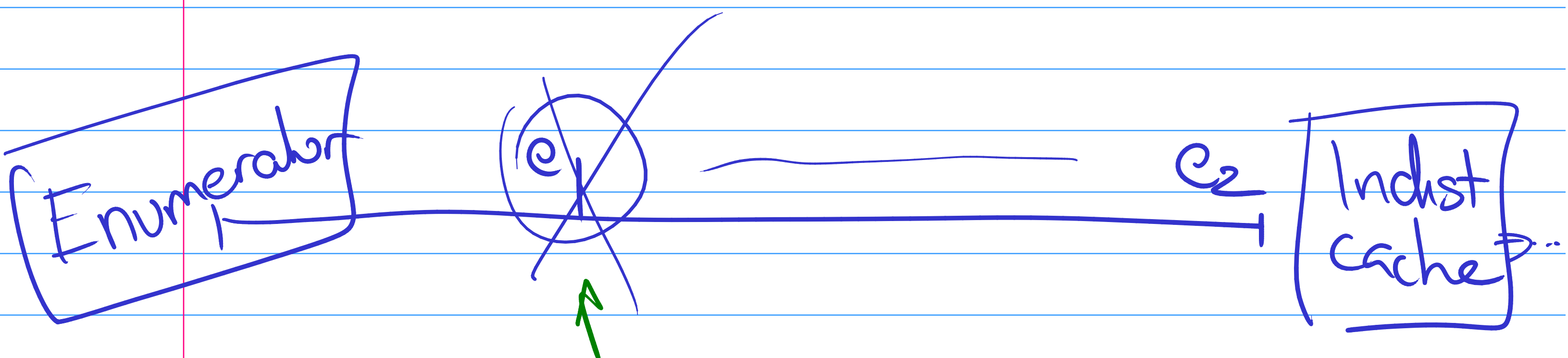
$\not\Longrightarrow$

$$e_1 = x + y - z \qquad\qquad e_2 = x$$

$$C = \{(x \mapsto 1, y \mapsto 2, z \mapsto 2), (x \mapsto 18, y \mapsto 3, z \mapsto 3)\}$$

$$e_1 = 1 \qquad\qquad e_1 = 18$$

$$e_2 = 1 \qquad\qquad e_2 = 18$$




<u>Worry</u>: If $e_1$ is the answer, there is the risk of an overeager cache killing $e_1$.

$\to \boxed{\text{Enumerator}} \to \boxed{\text{Cache}} \to$ vs. $\to \boxed{\text{Enumeration modulo indistinguishability}}$

$$G^1 = \underline{\qquad\qquad\qquad}$$

$$G^2 = \underline{\qquad\qquad\qquad} \quad \ldots, x+y, \ldots, \cancel{y+x}, \ldots$$

$$G^3 = (G^1 + G^1) \cup (G^1 - G^1)$$

$$G^5 = (G^1 + G^3) \cup \ldots \quad x+(x+y), \ldots \quad \cancel{x+(y+x)}$$

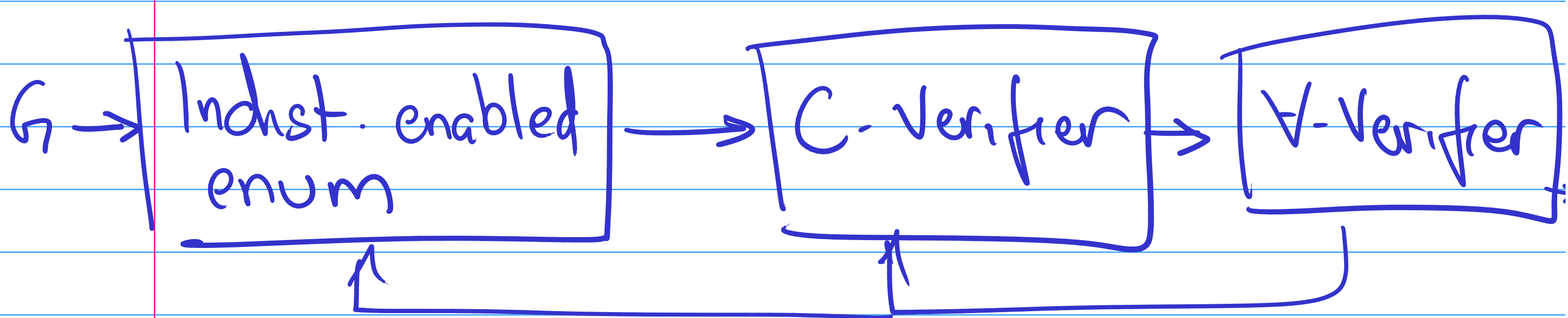
$$G^* = G^1 \cup G^2 \cup G^3 \cup \ldots$$

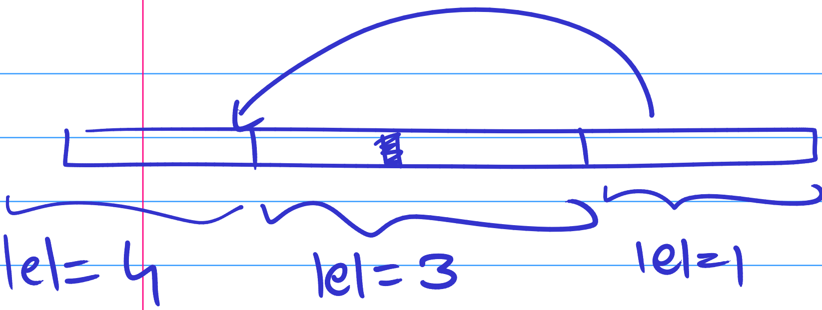
---

As we are listing elements of $G^n$, pass them through indistinguishability filter. Check if signature seen before. Don't add to $G^n$ if this is the case.

This approach kills not just $y+x$, but all expressions which use $y+x$.

## SyGuS Factory V 3.0 / Enumerative SyGuS solver






$e ::= 0 \mid 1 \mid x \mid y \mid$ if $e_1 \le e_2$ then $e_3$ else $e_4$

<u>Question</u>: How many expressions $e \in G^n$?