Lecture 26
Monday, April 27, 2020 5:08 PM

- Hello !

- Assignment 3 released.
  Due date: May 13

- Bonus questions for assignment 3.

- Project presentations.
  Presentation dates: May 6–13. (20 minutes; send email)
  Report submission dates: May 15

My own deadline
May 19.
Exams last day
May 13

Syntax Guided Synthesis

	FV 1.0	FV 2.0	FV 3.0	...	
Time	10100 s	300 s	210 s		≥100 s
Bottleneck	V. Verifier	C-Verifier + Enumerator	C-Verifier + Enumerator		Enumeration bottleneck

The enumerator is the bottleneck. How to build this?

Building a simple bottom-up enumerator.

G:

```
Start ::= x | y | 0 | 1
        | Start + Start
```

All Expressions of G in order = All Exprs of G of size 1 in order + All Exprs of G of size 2 in order +

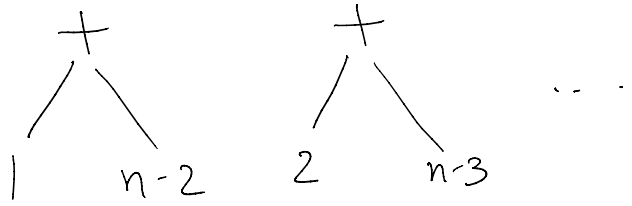
order size 1 in order In order

$$G^* = G^1 ++ G^2 ++ G^3 ++ \cdots$$

How to build G^n ?

If $n=1$: simply $[x, y, 0, 1]$

Otherwise :



Ex: (+ (x y))

For each n_1 n_2 s.t. $n_1+n_2 = n-1$

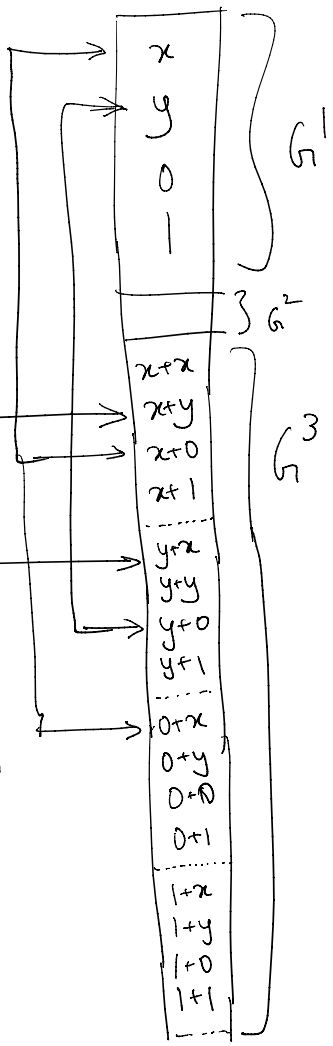
pick e_1 in G^{n_1} $e_2 \in G^{n_2}$

produce e_1+e_2.

Accelerating the Enumerator

G_1:

Start ::= x | y | 0 | 1
| Start + Start



Enumeration of Many Algebraically Equivalent Expressions

$\forall\ e_1\ e_2 \quad e_1 + e_2 = e_2 + e_1$

$\forall\ e_1\ e_2\ e_3 \quad (e_1 + e_2) + e_3 = e_1 + (e_2 + e_3)$

$\forall\ e \quad e + 0 = 0 + e = e$

Question: How to prevent enumeration of algebraically equivalent expressions?

$\forall\ e_1\ e_2 \quad \boxed{e_1 + e_2 = e_2 + e_1}$

$$(x+y) + x = x + (x+y)$$

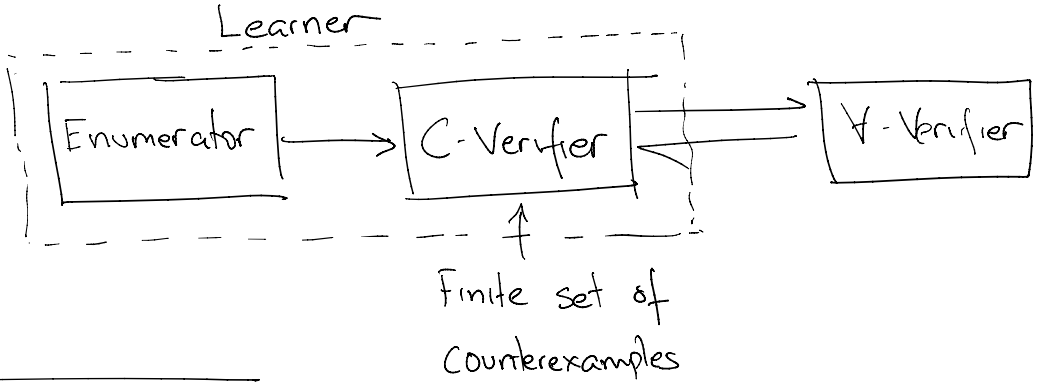
$$(x-y) + (z+x) = (z+x) + (x-y)$$

$$\vdots$$

- Traditionally algebraic laws phrased as

- Traditionally algebraic laws phrased as rewrite rules.

Indistinguishability



Original problem statement: Find $f \in G$ s.t. $\forall \vec{x}\ \varphi(\vec{x}\ f)$ holds.

Approximation used by the C-verifier: $\forall \vec{x} \in C\ \varphi(\vec{x}\ f)$ holds

---

Original form of commutativity: $\forall e_1\ e_2\ \forall \vec{x}\ e_1 + e_2 = e_2 + e_1$

$\forall \vec{x} \in C\ e_1 + e_2 = e_2 + e_1$

Original definition of: Two expressions e_1 e_2 are equivalent if:

definition of equivalence

equivalent if:

$$\forall \vec{x},\ e_1(\vec{x}) = e_2(\vec{x}).$$

Indistinguishable

Two expressions e_1 e_2 are indistinguishable if

$$\forall \vec{x} \in C \quad e_1(\vec{x}) = e_2(\vec{x}).$$

C: Finite set.

Claim: Indistinguishability approximates equivalence.

Indistinguishability $\not\Longleftrightarrow$ Equivalence

Ex: $x+y$ $\quad$ x

$C = \{\{x \mapsto 5, y \mapsto 0\} \quad \{x \mapsto 3, y \mapsto 0\}\}$

Indistinguishable in C

Not equivalent.

Claim: For all specifications φ

for all sets of counterexample valuations $\underline{C}$

If e_1 and e_2 are indistinguishable

then $\forall$ e which satisfies φ

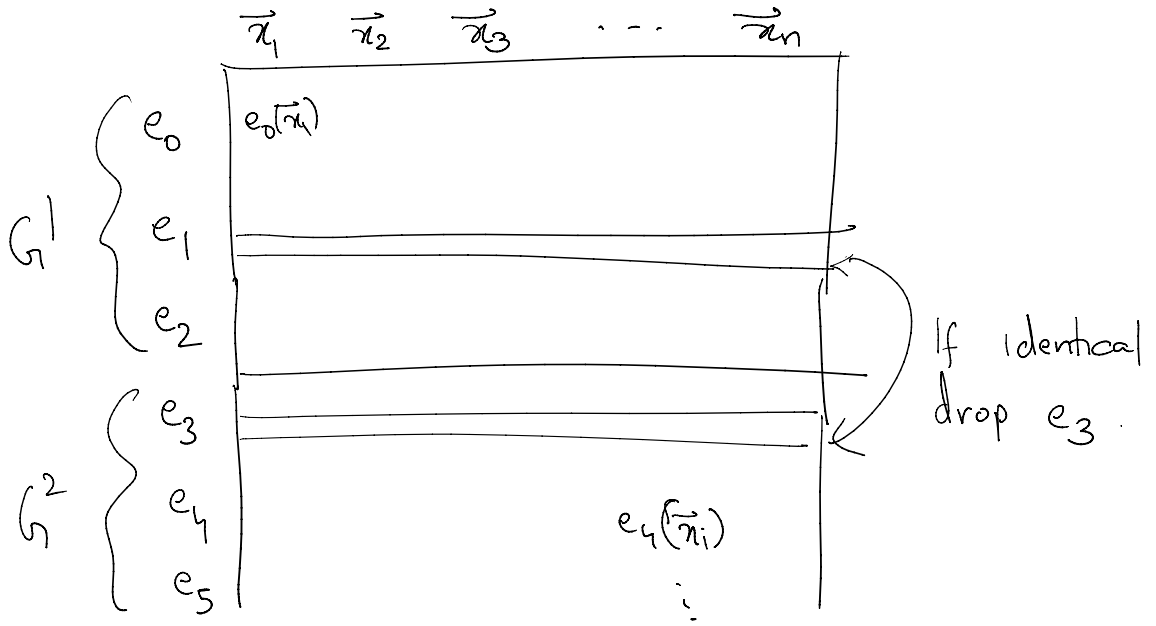
$e[e_1$ replaced with $e_2]$ will also satisfy φ.
in the C-verifier.

Enumerator: For each new expression e,

compute the signature:

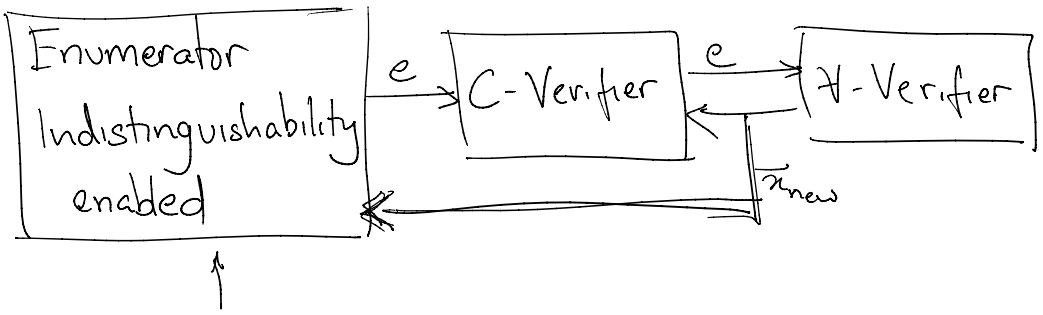
$$[e(\vec{x}_1) \quad e(\vec{x}_2) \quad \cdots \quad e(\vec{x}_n)]$$

where $\{\vec{x}_1 \; \vec{x}_2 \cdots \vec{x}_n\} = C$

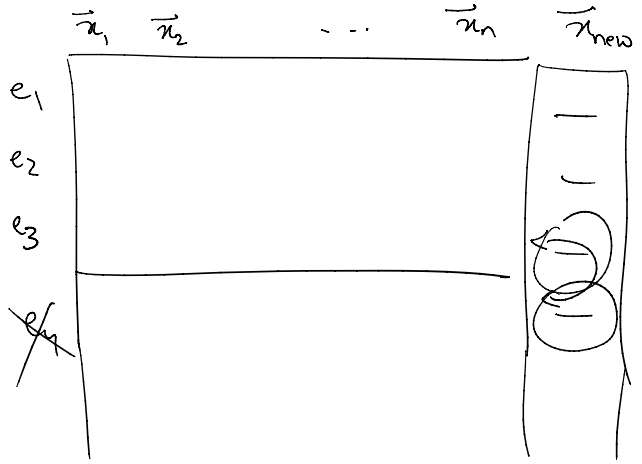


FV 4.0

ESolver



Restarts for each new counterexample.



Updating signatures
||
Adding a new column

Indist Enabled Enumerator:

for n from 1 to ∞:

for each expression $e \in \underbrace{G^n/C}$

yield e.

G^n modulo C.

To enumerate expressions in G^n/C

If n == 1 then $[0\ \ 1\ \ x\ \ y]/C$. Start ::= 0 | 1 | x | y | Start + Start

Else:

Else:

Pick a division: n_1 n_2

s.t. $n = 1 + n_1 + n_2$

For each $e_1 \in G^{n_1}/C$

$e_2 \in G^{n_2}/C$

compute signature of $e_1 + e_2$

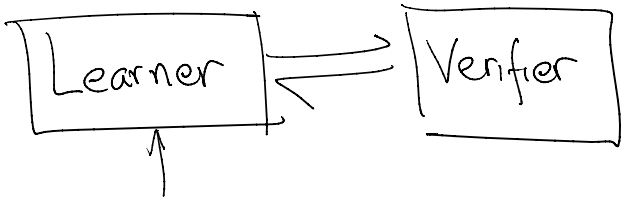
If new then emit $e_1 + e_2$

Given new counterexample point $\vec{x}_{new}$

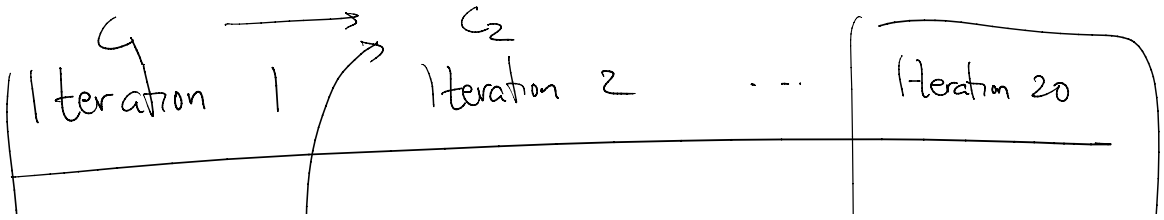
$C := C \cup \{\vec{x}_{new}\}$

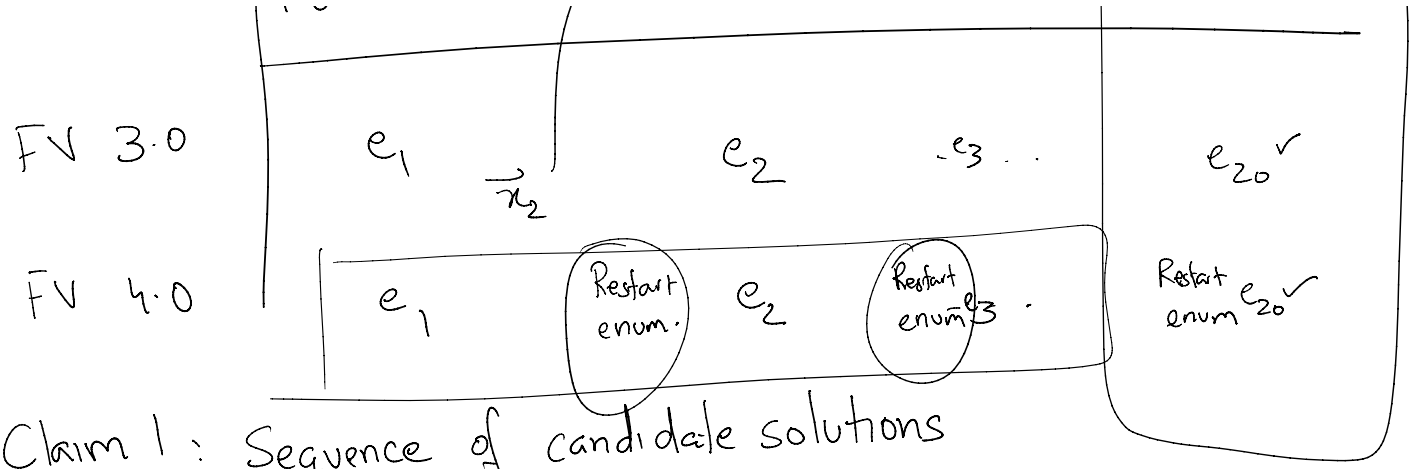
Restart everything else.

Empirical observation



Majority of the time is spent in the last iteration, in the learner.





Claim 1: Sequence of candidate solutions

e_1 e_2 ... e_k

submitted to the V-Verifier is

is identical for FV 3.0 & FV 4.0.

Indistinguishability fails when there are let expressions

```
let y = 5 in
let e = x+y in
e+3
```

```
let y = 0 in
let e = x+y in
e+3
```

$y := 5$;

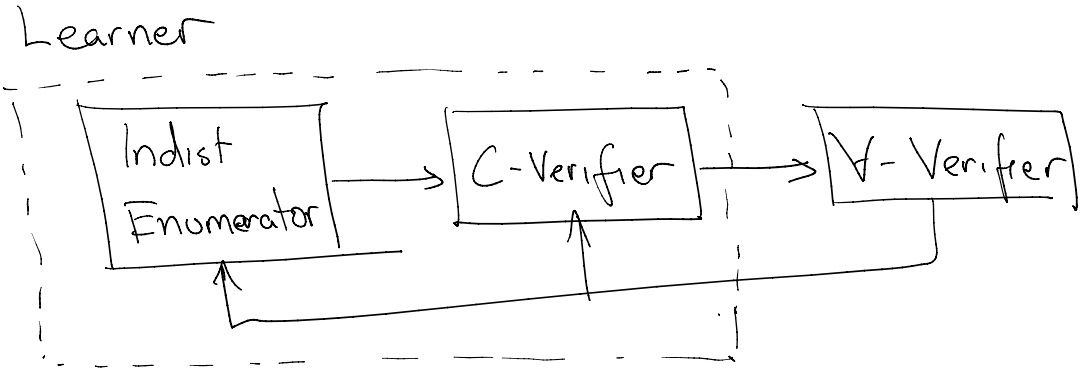
$e := x+y$;

return $e+3$

$x+y$ is distinguishable from x

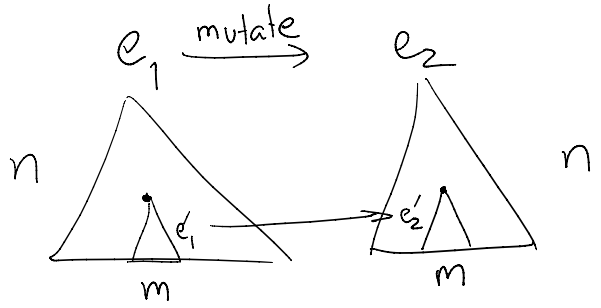
$x+y$ & x are indistinguishability.

FV 4.0



Other Instantiations of CEGIS

1. Stochastic Solver: Find expressions by a random walk.



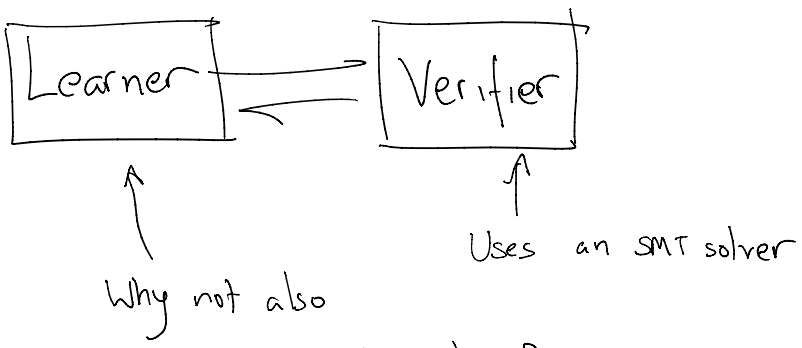
Score of $e_1 \geq$ Score of e_2

Markov Chain Monte Carlo techniques

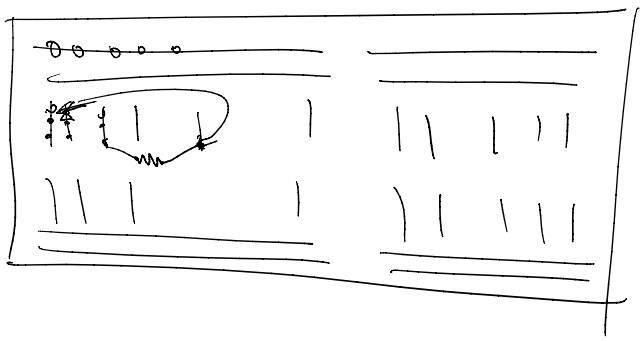
Metropolis Hastings method.

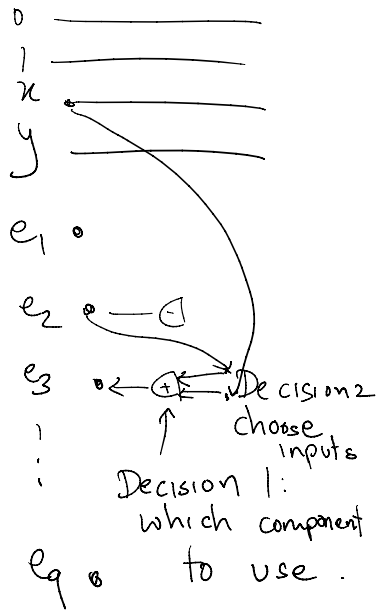
Provides probabilistic formula to accept e_2.

2. Constraint-based solving



$$e = \frac{(x+y)+\left(y+\frac{(z+x)}{3}\right)}{9}$$





---

3. Unification

$$\forall\ x\ y \quad f(x\ y) \geq x$$

$$f(x\ y) \geq y$$

$$(f(x\ y) = x \quad \text{or} \quad f(x\ y) = y).$$

$$C = \{\ \underline{(x \mapsto 3 \quad y \mapsto 2)} \quad \underline{(x \mapsto 1 \quad y \mapsto 18)}\ \}$$

FSolver → x works here

$(x \mapsto 1 \quad y \mapsto 18)$ } y works here

Unification engine says $\{x \; y\}$ exhaustively cover C.

Question: How to figure out when to use x & when to use y?

EUSolver. [Use decision tree learning to produce a classifier.

On Wednesday:

1. Complexity of program synthesis:

   TQBF PSPACE-completeness

2. Use of SyGuS to learn program invariants

3. Course recap.