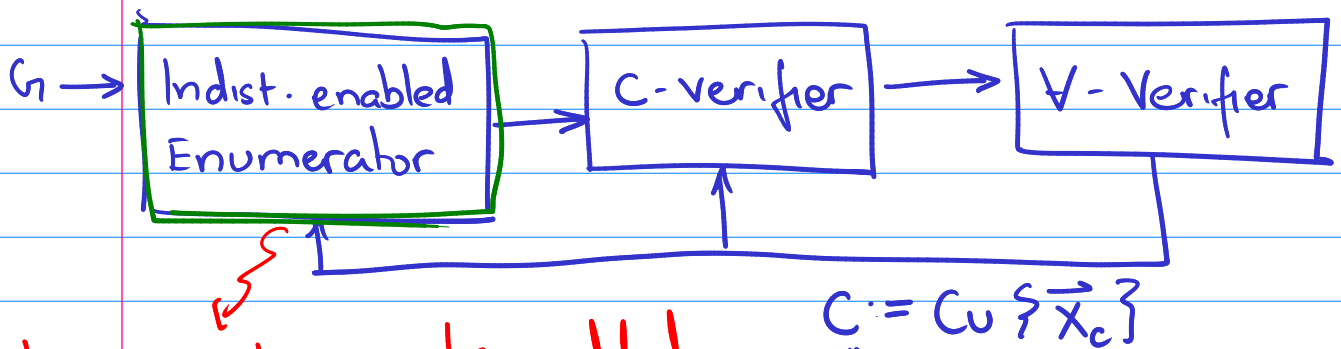
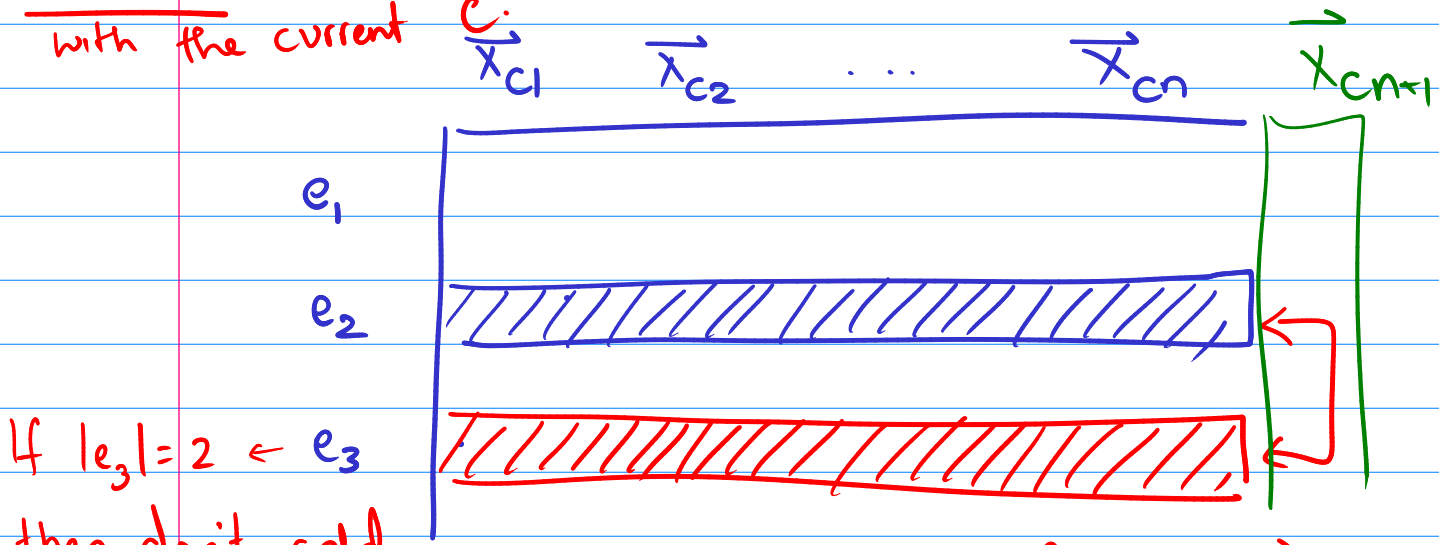
Lecture 15

SyGuS Factory Version 3.0 / ESolver



When counterexample added, restart the enumerator with the current C.

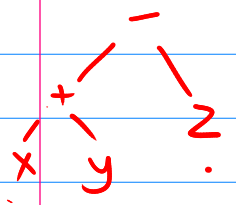


Essentially, we are defining an equivalence relation $e_1 \sim_C e_2$ if $sig(e_1) = sig(e_2)$.

Ex.

$C = \{(x=3, y=2, z=2)$ $(x=8, y=6, z=6)\}$

$e_2 = x$	3	8
$e_3 = x+y-z$	3	8

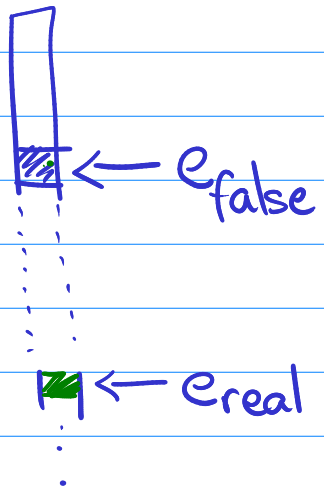


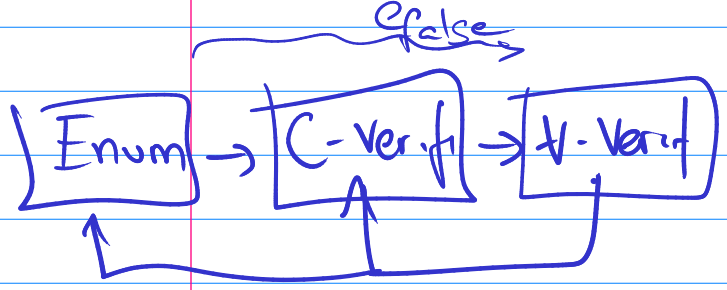
$sig(e_3) = sig(e_2)$ $\quad |e_3| = 5$

So e_3 not inserted into G_1^5

$e_4 = x+(y-y)$	3	8

Claim: If expressions e_1 & e_2 are algebraically equivalent, then $sig(e_1) = sig(e_2)$.

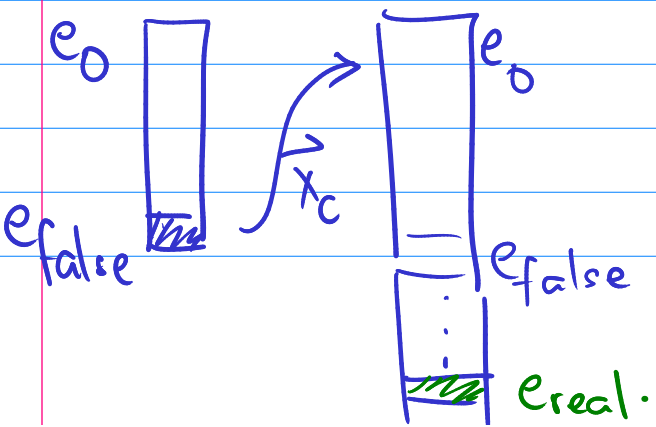




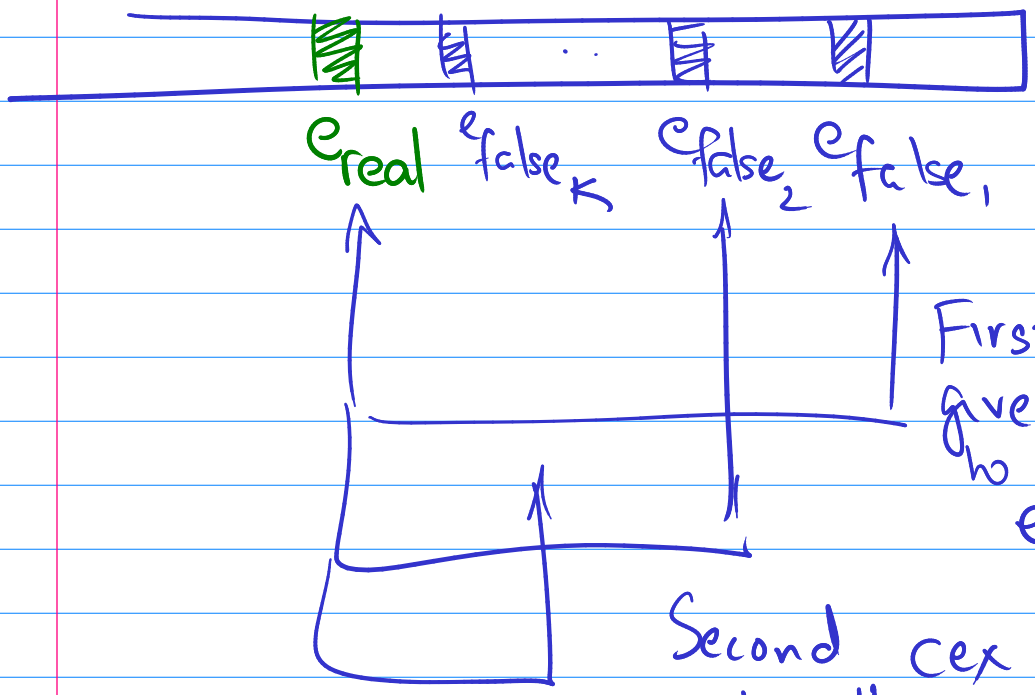
Claim: If e_{false} is the first expression to survive the C-verifier, $e_{false} \sim_C e_{real}$, & e_{real} is the smallest solution to the SyGuS problem, then, if $\vec{x}_C$ is the cex returned by the V-verifier

$$e_{false} \not\sim_{C \cup \{\vec{x}_C\}} e_{real}$$

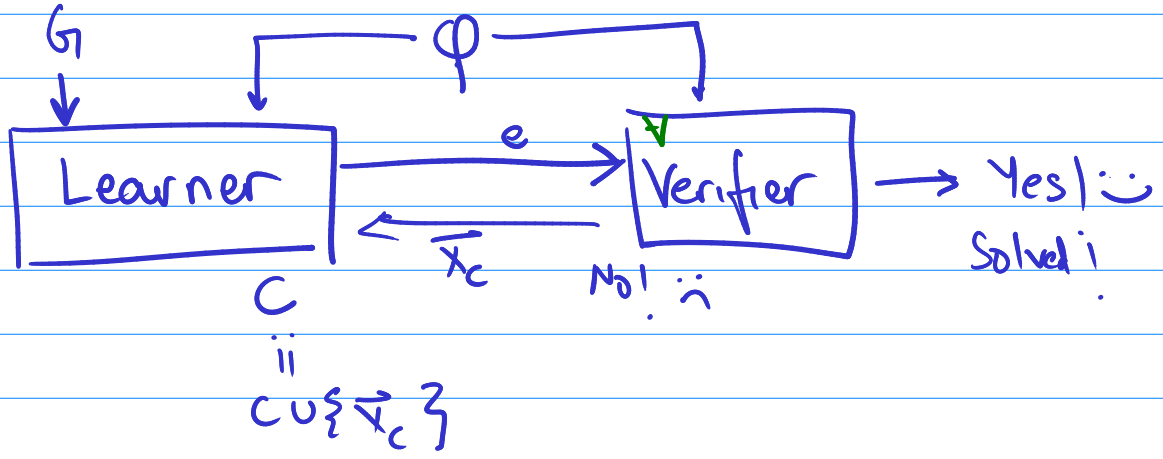
$$e_{false}(\vec{x}_C) \neq e_{real}(\vec{x}_C).$$



(w/o regard for $\sim_C$) All expressions ordered acc. to increasing size



The CEGIS Loop



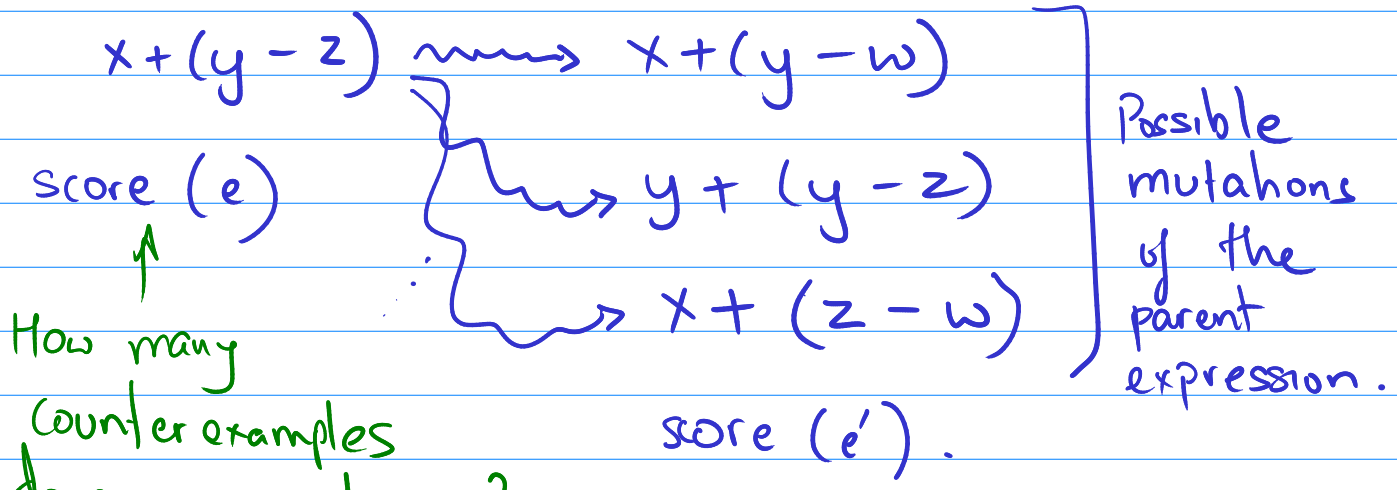
Q1: Is there a learner which can learn regardless of the verifier? YNN ← Isn't ESolver already an example of a robust learner?

Q2: Is there a verifier which can make an arbitrary learner succeed? YYY

- If the space of target concepts is finite, then every verifier is robust.

- If the space of target concepts is infinite, imagine a stubborn learner: it produces an e which works for everything in C, but is still wrong for φ.

Other Learners : Stochastic Sygus Solver



- If score(e) = |C|, then we forward it to the verifier.
- Otherwise mutate $e \rightsquigarrow e'$.

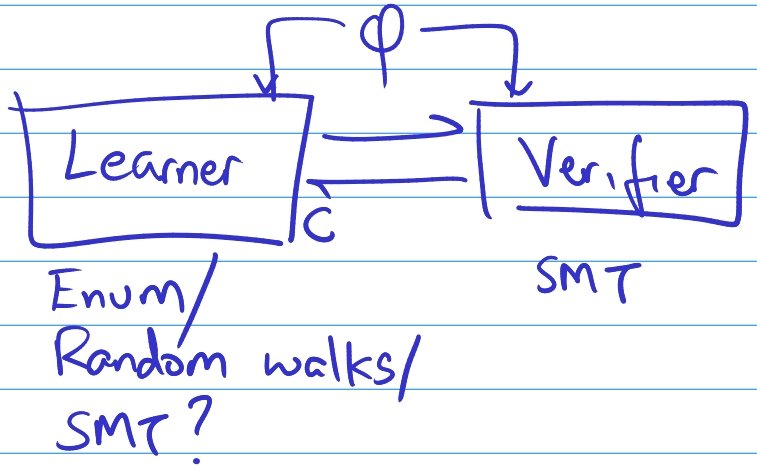
Compare score (e) vs. score (e')

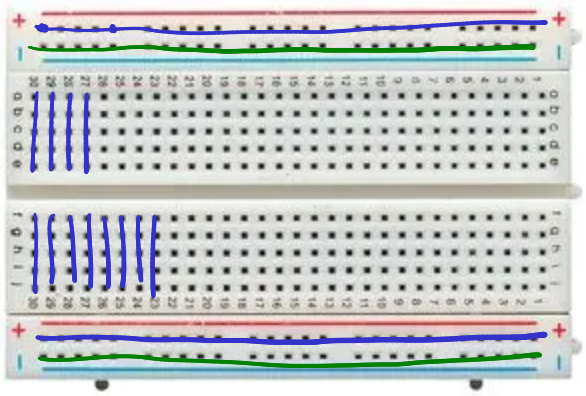
If $s_e \leq s_{e'}$, then jump!

Otherwise, with some probability $f(s_e, s_{e'})$, jump anyway!

↳ Metropolis Hastings algorithm / MCMC - methods. Schkufza et al. ASPLOS 2013.

Other Learners: Constraint-Based Solver



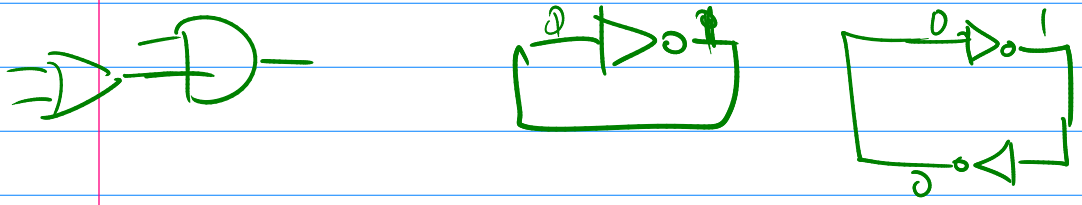


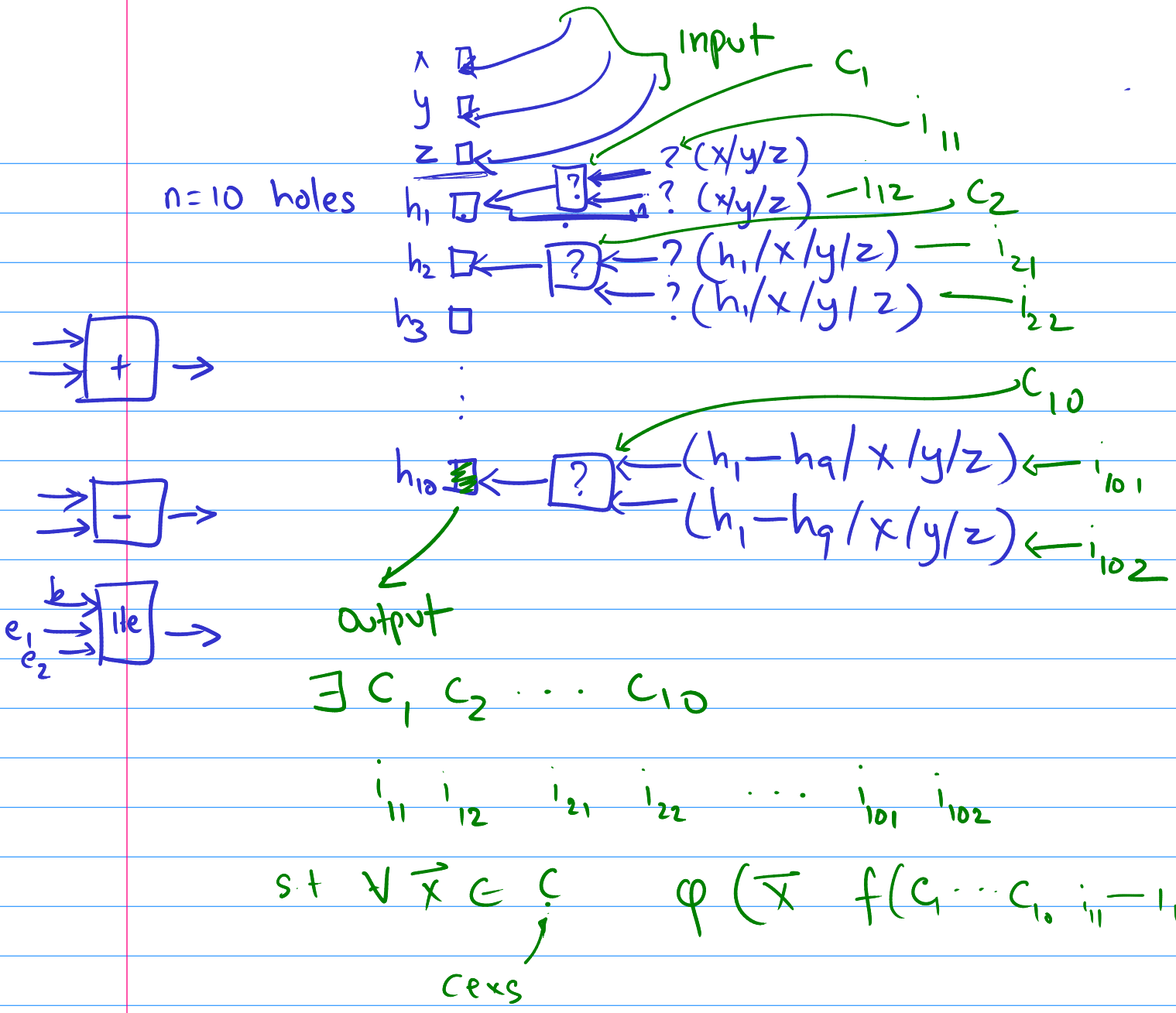
Problem: Given quantifier free formula φ

grammar G ← If each rule is a little chip/component placeable on a breadboard

Set of cexs C

circuit ← find an expression $e \in G$ which works(φ) on all C.





$$\exists\, c_1\, c_2 \cdots c_{10}$$

$$i_{11}\ i_{12}\ i_{21}\ i_{22} \cdots i_{101}\ i_{102}$$

$$\text{s.t}\ \forall \vec{x} \in \underbrace{C}_{\text{cexs}} \quad \varphi\left(\vec{x}\ f(c_1 \cdots c_{10}\ i_{11} \cdots i_{102})\right)$$

Let $h_{-2} = x$

$h_{-1} = y$

$h_0 = z$

let $h_1 = ??(\ ?\ ,\ ?\ )$

$h_2 = ?\ (\ ?\ ,\ ?\ )$

⋮

in h_{10}

Program Verification

Unit 3: Proof Techniques

- We need a language: Imp(erative)

All variables hold mathematical integers.

```
x := input()
x := x+2;
y := 0;
while (y < x) {
    y := y+2
    if (y-x = 5)
    then crash!
}
```

Outcome 1

Fin & halt

Outcome 2

Crash!

Outcome 3

Loop forever.

$AExp ::= x \mid y \mid z \mid \cdots$ (Var)

$\mid$ input (Ask user for input)

$\mid AExp_1 + AExp_2$

$\mid AExp_1 - AExp_2$

$Cmd ::= v := AExp$ (v is a Var) (Assignments)

$\mid Cmd_1 ; Cmd_2$

$BExp ::= AExp \leq AExp \mid$ if (BExp) then Cmd_1 else Cmd_2

$\mid BExp_1 \wedge BExp_2$

$\mid BExp_1 \vee BExp_2$

$\mid \neg BExp_1 \mid$ while (BExp) do Cmd_1

$\mid$ skip