CS 444 - Compiler Construction **Winter 2020**

Lecture 5: January 20th, 2020

Lecturer: Ondřej Lhoták *Notes By: Harsh Mistry*

5.1 Analysis Continued

5.1.1 Parsing Continued

Example 5.1 *LL(1) grammar.*

- $S \to E\$$ *(Added after augmentation)*
- $E \to aE^{|}$
- $E^{|} \to +a$
- $E^{|} \to \epsilon$

input "a+a" yeids the parsed derivation of $E \to aE^{|} \to a + a$

Definition 5.2 *Augment a grammer by adding new production rules or terminal characters.*

- Use a stack to represent α
- $\alpha = ywA\gamma$
  - Pop w off stack, compute with actual input
  - Pop A off stack, use A and next input symbol to chose $A \to \beta$, pushβ

5.1.1.1 LL(1) Parsing Algorithm

- Parse =
  - Push S$ on stack
  - for a in x$
    - While top of stack is a non-terminal
      - Pop A
      - Let $A \to \beta = predict(A, a)$
      - Push β
    - Pop b (Pop top of stack and result is assigned to b)
    - if $b \neq a$, then throw ERROR
  - Predict(A, a) =
    - $A \to \beta \in R \mid \exists_\gamma, \beta \to^* a\gamma$

- OR
  - $(\beta \rightarrow^* \epsilon)$
  - AND $(\exists \gamma\delta \ , \ S \rightarrow^* \gamma A a \delta)$

Definition 5.3 *If* $|Predict(A, a)| \leq 1$ *for all* A, a*, then grammar is* ***LL(1)***