

# CS480

# Translators

Parsing

Chap. 4

# Criteria for Parsing

- Efficient – proportional to size
- Determine action by fixed # tokens
- Practical Considerations

- 1 Lookahead

- No backtracking

- LL(1) grammar

- What is LL(k)?

left to right scan  
left most derivation  
K tokens for lookahead

program

how many tokens sent to parser at once or  
do you allow backtracking in the  
one lookahead that can be parsed predictively w/o back-track

# LL Grammars

- Top-down Parsing

- Recursive descent

- General

- Predictive

- Table-driven

explicit stack

milestone 3

only look at next token at a time!

not predictive

infinite recursion

$S \rightarrow TS$

$S \rightarrow \epsilon$

$S \rightarrow SS$

non-terms  
on stack

# Top Down Parsing

$A \rightarrow Xa$   
 $A \rightarrow Xa$

```
void A() {  
1)   Choose an A-production,  $A \rightarrow X_1 X_2 \cdots X_k$ ;  
2)   for (  $i = 1$  to  $k$  ) {  
3)       if (  $X_i$  is a nonterminal )  
4)           call procedure  $X_i()$ ;  
5)       else if (  $X_i$  equals the current input symbol  $a$  )  
6)           advance the input to the next symbol;  
7)       else /* an error has occurred */;  
    }  
}
```

- How does this change for the production below?

$A \rightarrow ab \mid a$

# Defining an LL Grammar

- Need two definitions:
- **First and Follow** — *non terms*

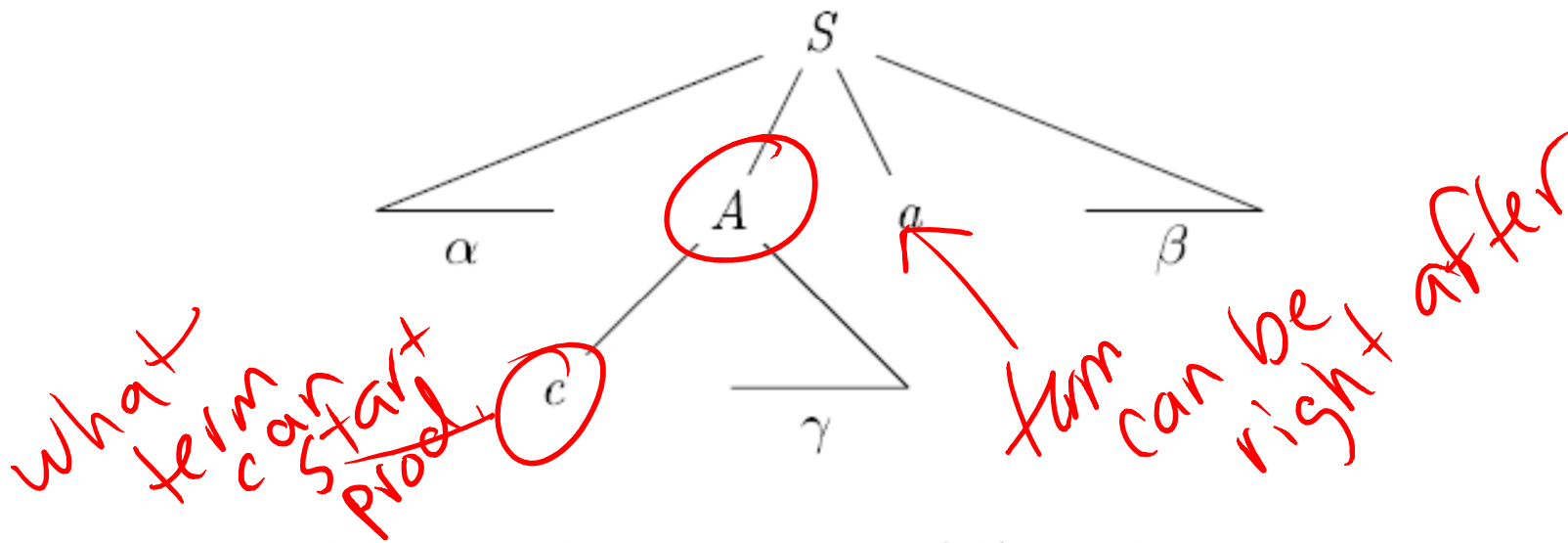


Figure 4.15: Terminal  $c$  is in  $\text{FIRST}(A)$  and  $a$  is in  $\text{FOLLOW}(A)$

# First( $\alpha$ )

- If  $\alpha$  is any string in grammar,  $\text{First}(\alpha)$  is set of terminals that begin strings derived from  $\alpha$ .
- If  $\alpha \xRightarrow{*} \varepsilon$ , then  $\varepsilon$  is in  $\text{First}(\alpha)$ .
- What does it mean if  $A \rightarrow \alpha \mid \beta$ , and  $\text{First}(\alpha)$  and  $\text{First}(\beta)$  are disjoint?

predictive parser

# Follow(A)

- If  $A$  is a nonterminal in grammar,  $\text{Follow}(A)$  is set of terminals that can appear immediately to the right of  $A$ .
- If  $A$  can be the rightmost symbol, then  $\$$  is in  $\text{Follow}(A)$ .
- What is  $\$$ ?

*nothing follows*

# Compute $\text{First}(X)$ for all symbols

- ~~• If  $X$  is a terminal, then  $\text{First}(X) = \{X\}$~~
- If  $X$  is a nonterminal and  $X \rightarrow Y_1 Y_2 \dots Y_k$  is a production, then place  $a$  in  $\text{First}(X)$  if for some  $i$ ,  $a$  is in  $\text{First}(Y_i)$  and  $\epsilon$  is in all  $\text{First}(Y_1), \dots, \text{First}(Y_{i-1})$ . If  $\epsilon$  is in  $\text{First}(Y_j)$  for all  $j=1, 2, \dots, k$ , then add  $\epsilon$  to  $\text{First}(X)$ .
- If  $X \rightarrow \epsilon$ , then  $\epsilon$  is in  $\text{First}(X)$



# Compute Follow(A) for nonterminals

- Place <sup>①</sup>\$ in Follow(S), where S is start symbol and \$ is the input endmarker.
- If there is a production  $A \rightarrow \alpha B \beta$ , then everything in First( $\beta$ ), except  $\epsilon$ , is in Follow( $B$ ).  
*Handwritten notes: "A is a non-terminal" points to A, "non-term" points to B, "no empty follows" points to the entire rule.*
- If there is a production  $A \rightarrow \alpha B$  or  $A \rightarrow \alpha B \beta$ , where  $\epsilon$  is in First( $\beta$ ), then everything in Follow(A) is Follow(B).  
*Handwritten notes: "epsilon in the first" points to First(beta) in the second rule, and "epsilon in the first" points to epsilon in this rule.*

# Example First and Follow

$E \rightarrow \textcircled{T} E'$

$E' \rightarrow \textcolor{red}{+} T E' \mid \epsilon$

$T \rightarrow \textcolor{red}{F} T'$

$T' \rightarrow \textcolor{red}{*} F T' \mid \epsilon$

$F \rightarrow \textcolor{red}{(} E \textcolor{red}{)} \mid \textcolor{red}{id}$

- First(E), First(E'), First(T), First(T'), First(F)?

- Follow(E), Follow(E'), Follow(T), Follow(T'), Follow(F)?

$\text{First}(E) = \{ (, id \}$

$\text{First}(T) = \{ \textcolor{red}{+}, \epsilon \}$

$\text{First}(F) = \{ \textcolor{red}{*}, \epsilon \}$

$\text{Follow}(E) = \{ ), \$ \}$

$\text{Follow}(E') = \{ ), \$ \}$

$\text{Follow}(T) = \{ \textcolor{red}{+}, ), \$ \}$

$\text{Follow}(T') = \{ \textcolor{red}{*}, \epsilon \}$

$\text{Follow}(F) = \{ \textcolor{red}{*}, \textcolor{red}{+}, ), \$ \}$

$\text{First}(E) = \{ (, id \}$   
 $\text{First}(T) = \{ \textcolor{red}{+}, \epsilon \}$   
 $\text{First}(F) = \{ \textcolor{red}{*}, \epsilon \}$

# Do we have LL Grammar? Predictive Parsing Table

- For each production  $A \rightarrow \alpha$  in the grammar:
  - For each terminal **a** in  $\text{First}(\alpha)$ , add  $A \rightarrow \alpha$  to  $M[A, a]$
  - If  $\epsilon$  is in  $\text{First}(\alpha)$ , then for each terminal **b** in  $\text{Follow}(A)$ , add  $A \rightarrow \alpha$  to  $M[A, b]$ . If  $\$$  is in  $\text{Follow}(A)$ , add  $A \rightarrow \alpha$  to  $M[A, \$]$  as well

NON - TERMINAL	INPUT SYMBOL					
	id	+	*	(	)	\$
$E$	$E \rightarrow TE'$			$E \rightarrow TE'$		
$E'$		$E' \rightarrow +TE'$			$E' \rightarrow \epsilon$	$E' \rightarrow \epsilon$
$T$	$T \rightarrow FT'$			$T \rightarrow FT'$		
$T'$		$T' \rightarrow \epsilon$	$T' \rightarrow *FT'$		$T' \rightarrow \epsilon$	$T' \rightarrow \epsilon$
$F$	$F \rightarrow \text{id}$			$F \rightarrow (E)$		