

LL(1) Parsing Table & Parsing

2 de junho de 2024 14:08

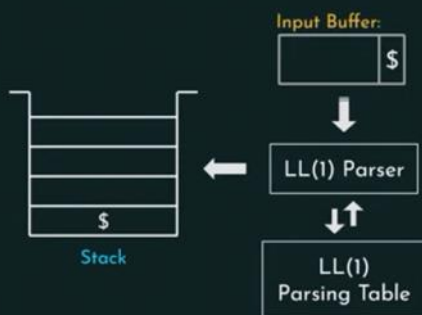
LL(1) Parsing Table

LL(1) Parsing Table						
	id	()	*	+	\$
E	$E \rightarrow TE'$	$E \rightarrow TE'$				
E'			$E' \rightarrow \epsilon$		$E' \rightarrow +TE'$	$E' \rightarrow \epsilon$
T	$T \rightarrow FT'$	$T \rightarrow FT'$				
T'			$T' \rightarrow \epsilon$	$T' \rightarrow *FT'$	$T' \rightarrow \epsilon$	$T' \rightarrow \epsilon$
F	$F \rightarrow id$	$F \rightarrow (E)$				

35 Compiler Design

LL(1) Parser:

— "1" look-ahead symbol.
— Use **Left most derivation**.
— Scan from **Left to right**.



Top down approach:



$S \Rightarrow aABe$
 $\Rightarrow aAbcBe$
 $\Rightarrow aabcBe$
 $\Rightarrow aabcde$

Decision:
Which production to use.

(Left most Derivation)

	FIRST	FOLLOW
$E \rightarrow TE'$	{id, (}	{\$,)}
$E' \rightarrow +TE' \mid \epsilon$	{+, ϵ }	{\$,)}
$T \rightarrow FT'$	{id, (}	{+, \$,)}
$T' \rightarrow *FT' \mid \epsilon$	{*, ϵ }	{+, \$,)}
$F \rightarrow id \mid (E)$	{id, (}	{*, +, \$,)}

	id	()	*	+	\$
E	$E \rightarrow TE'$	$E \rightarrow TE'$				
E'			$E' \rightarrow \epsilon$		$E' \rightarrow +TE'$	$E' \rightarrow \epsilon$
T	$T \rightarrow FT'$	$T \rightarrow FT'$				
T'			$T' \rightarrow \epsilon$	$T' \rightarrow *FT'$	$T' \rightarrow \epsilon$	$T' \rightarrow \epsilon$
F	$F \rightarrow id$	$F \rightarrow (E)$				

Rules:

1. All the ϵ -productions are placed under FOLLOW sets
2. Remaining productions are placed under the FIRSTs

LOOKAHEAD(A \rightarrow a) = FIRST(a) se a não deriva ϵ .
LOOKAHEAD(A \rightarrow a) = FIRST(a) U FOLLOW(A) caso contrário.

$E \rightarrow F + E \mid F$
 $F \rightarrow \epsilon$
 $F \rightarrow (E)$

LOOKAHEAD(E \rightarrow F + E) = FIRST(F + E) = { ϵ , (, + }
LOOKAHEAD(E \rightarrow F) = FIRST(F) U FOLLOW(E) = { ϵ , (, \$,) }
LOOKAHEAD(F \rightarrow ϵ) = FIRST(ϵ) U FOLLOW(F) = { ϵ , +, \$,) }
LOOKAHEAD(F \rightarrow (E)) = FIRST((E)) = { (}

FIRST FOLLOW		
$S \rightarrow aABb$	a	\$
$A \rightarrow c \mid \epsilon$	c	, \$,)
$B \rightarrow d \mid \epsilon$	d	, \$,)

	FIRST	FOLLOW
$S \rightarrow aABb$	a	$\$$
$A \rightarrow c \mid \epsilon$	c, ϵ	d, b
$B \rightarrow d \mid \epsilon$	d, ϵ	b

-	a	b	c	d	\$
S	$S \rightarrow aABb$				
A		$A \rightarrow \epsilon$	$A \rightarrow c$	$A \rightarrow \epsilon$	
B		$B \rightarrow \epsilon$		$B \rightarrow d$	

How will the parse tree be generated for this particular input string?

