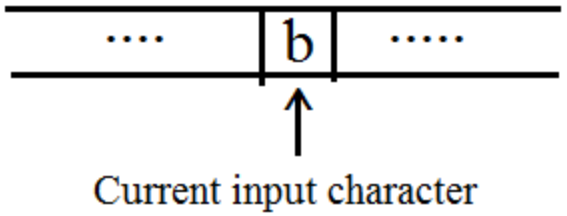
Top-Down and LL(1) Parsing

Parse Table

Applicable Productions

For leftmost variable A and current input token $b \in \Sigma \cup \{\$\}$,
the applicable productions are

- all productions $A \to \alpha$ s.t. $b \in \mathbf{FIRST}(\alpha)$, and
- all productions $A \to \alpha$ s.t. $\epsilon \in \mathbf{FIRST}(\alpha)$ and $b \in \mathbf{FOLLOW}(A)$.



Let's construct the parse table for the grammar we've been looking at.

$$
\begin{aligned}
S &\rightarrow XSa \mid Yc \\
X &\rightarrow aY \mid YY \\
Y &\rightarrow bSa \mid cX \mid \epsilon
\end{aligned}
$$

$\text{FIRST}(XSa) = \{a, b, c\}$ $\qquad$ $\text{FOLLOW}(S) = \{a, \$\}$

$\text{FIRST}(Yc) = \{b, c\}$ $\qquad$ $\text{FOLLOW}(X) = \{a, b, c\}$

$\text{FIRST}(YY) = \{b, c, \epsilon\}$ $\qquad$ $\text{FOLLOW}(Y) = \{a, b, c\}$

LEFTMOST VARIABLE	CURRENT INPUT TOKEN			
	a	b	c	\$
S	$S \rightarrow XSa$	$S \rightarrow XSa \mid Yc$	$S \rightarrow XSa \mid Yc$	none
X	$X \rightarrow aY \mid YY$	$X \rightarrow YY$	$X \rightarrow YY$	none
Y	$Y \rightarrow \epsilon$	$Y \rightarrow bSa \mid \epsilon$	$Y \rightarrow cX \mid \epsilon$	none

LEFTMOST VARIABLE	CURRENT INPUT TOKEN			
	a	b	c	\$
S	$S \to XSa$	$S \to XSa \mid Yc$	$S \to XSa \mid Yc$	none
X	$X \to aY \mid YY$	$X \to YY$	$X \to YY$	none
Y	$Y \to \epsilon$	$Y \to bSa \mid \epsilon$	$Y \to cX \mid \epsilon$	none

Input string: *aca*

Note: There is no need to add ***S'→S$*** *always. If we add this then an entry in the parse table with S' should exist!*

STACK	CURRENT INPUT	PRODUCTION TO APPLY
$S\$$	$aca\$$	$S \to XSa$
$XSa\$$	$aca\$$	$X \to aY$ (backtrack $X \to YY$)
$aYSa\$$	$aca\$$	match a
$YSa\$$	$ca\$$	$Y \to \epsilon$ (backtrack $Y \to cX$)
$Sa\$$	$ca\$$	$S \to Yc$ (backtrack $S \to XSa$)
$Yca\$$	$ca\$$	$Y \to \epsilon$ (backtrack $Y \to cX$)
$ca\$$	$ca\$$	match c
$a\$$	$a\$$	match a
\$	\$	successful parse

LL(1) Grammar

- A grammar without left recursion is said to be LL(1), if there is atmost one production in every entry of the parsing table.

- So the previous grammar $\begin{array}{l} S \rightarrow XSa \mid Yc \\ X \rightarrow aY \mid YY \\ Y \rightarrow bSa \mid cX \mid \epsilon \end{array}$ is not LL(1).

$$
\begin{array}{lcl}
E & \rightarrow & T\ E' \\
E' & \rightarrow & +\ T\ E' \mid \epsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & *\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \mathbf{id}
\end{array} \tag{4.28}
$$

Variable	FIRST
F	(, id
T	(, id
E	(, id
E’	+, ε
T’	*, ε

$$\text{FOLLOW}(E) = \text{FOLLOW}(E') = \{), \$\}.$$

$$\text{FOLLOW}(T) = \text{FOLLOW}(T') = \{+,), \$\}.$$

$$\text{FOLLOW}(F) = \{+, *,), \$\}.$$

$$
\begin{array}{lcl}
E & \rightarrow & T\ E' \\
E' & \rightarrow & +\ T\ E' \mid \epsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & *\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \mathbf{id}
\end{array} \qquad (4.28)
$$

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	$F \rightarrow \mathbf{id}$			$F \rightarrow (E)$		

Parse Table for Grammar (4.28)

So, the grammar is LL(1).

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

MATCHED	STACK	INPUT	ACTION
	$E\$$	$\mathbf{id} + \mathbf{id} * \mathbf{id}\$$	
	$TE'\$$	$\mathbf{id} + \mathbf{id} * \mathbf{id}\$$	output $E \to TE'$
	$FT'E'\$$	$\mathbf{id} + \mathbf{id} * \mathbf{id}\$$	output $T \to FT'$
	$\mathbf{id}\ T'E'\$$	$\mathbf{id} + \mathbf{id} * \mathbf{id}\$$	output $F \to \mathbf{id}$
id	$T'E'\$$	$+\,\mathbf{id} * \mathbf{id}\$$	match id
id	$E'\$$	$+\,\mathbf{id} * \mathbf{id}\$$	output $T' \to \epsilon$
id	$+\,TE'\$$	$+\,\mathbf{id} * \mathbf{id}\$$	output $E' \to +\,TE'$
id +	$TE'\$$	$\mathbf{id} * \mathbf{id}\$$	match +
id +	$FT'E'\$$	$\mathbf{id} * \mathbf{id}\$$	output $T \to FT'$
id +	$\mathbf{id}\ T'E'\$$	$\mathbf{id} * \mathbf{id}\$$	output $F \to \mathbf{id}$
id + id	$T'E'\$$	$*\,\mathbf{id}\$$	match id
id + id	$*\,FT'E'\$$	$*\,\mathbf{id}\$$	output $T' \to *\,FT'$
id + id *	$FT'E'\$$	$\mathbf{id}\$$	match *
id + id *	$\mathbf{id}\ T'E'\$$	$\mathbf{id}\$$	output $F \to \mathbf{id}$
id + id * id	$T'E'\$$	$\$$	match id
id + id * id	$E'\$$	$\$$	output $T' \to \epsilon$
id + id * id	$\$$	$\$$	output $E' \to \epsilon$

Moves made by LL(1) parser on input **id + id * id**

$$E \underset{lm}{\Rightarrow} TE' \underset{lm}{\Rightarrow} FT'E' \underset{lm}{\Rightarrow} \mathbf{id}\,T'E' \underset{lm}{\Rightarrow} \mathbf{id}\,E' \underset{lm}{\Rightarrow} \mathbf{id} + TE' \underset{lm}{\Rightarrow} \cdots \underset{lm}{\Rightarrow} \mathbf{id} + \mathbf{id} * \mathbf{id}$$