

**Ex:**

$$S \rightarrow bSX \mid Y$$

$$X \rightarrow XC \mid bb$$

$$\bar{X} \rightarrow C\bar{X} \mid \lambda$$

$$Y \rightarrow b \mid bY$$

$$C \rightarrow ccC \mid CX \mid cc$$

	<b>First</b>	<b>Follow</b>
S	b	\$, b
X	b	\$, b, c
X'	c, λ	\$, b, c
Y	b	\$, b
C	c	\$, b, c

### 2.6.6 Bottom – Up Parsing

Bottom up parsers start from the sequence of terminal symbols and work their way back up to the start symbol by repeatedly replacing grammar rules' right hand sides by the corresponding non-terminal. This is the reverse of the derivation process, and is called "reduction".

**Ex:1** consider the grammar

$$S \rightarrow aABe$$

$$A \rightarrow Abc \mid b$$

$$B \rightarrow d$$

وهي عملية الاشتقاق من الاسفل الى الاعلى  
عكس Top down

The sentence **abcde** can be reduced to S by the following steps:

**Sol:**

$$abcde$$

$$aAbcde$$

$$aAde$$

$$aABe$$

$$S$$

$$\begin{aligned} S &\rightarrow aABe \\ S &\rightarrow aAbcBe \\ S &\rightarrow abcBe \\ S &\rightarrow abcde \end{aligned}$$

**Example:2** consider the grammar

$$S \rightarrow aABe$$

$$A \rightarrow Abc \mid bc$$

$$B \rightarrow dd$$

The sentence **abcddde** can be reduced to S by the following steps:

**Sol:**

$$abcddde$$

$$aAbcdde$$

$$aAdde$$

$$aABe$$

$$S$$

Definition: a handle is a substring that

- 1- Matches a right hand side of a production rule in the grammar
- 2- Whose reduction to the non-terminal on the left hand side of that grammar rule is a step along the reverse of a rightmost derivation?

There is a general style of bottom-up syntax analysis, known as **shift reduces parsing**.

**Example 1:**

parse the input **id +id \*id** for this grammar

- $E \rightarrow E+E$
- $E \rightarrow E * E$
- $E \rightarrow (E)$
- $E \rightarrow id$

يجب مراعاة اسبقية العمليات الرياضية اثناء الحل. واعتماد ال shift & reduce

STACK	INPUT	ACTION
(1) \$	$id_1 + id_2 * id_3 \$$	shift
(2) \$id <sub>1</sub>	$+ id_2 * id_3 \$$	reduce by $E \rightarrow id$
(3) \$E	$+ id_2 * id_3 \$$	shift
(4) \$E +	$id_2 * id_3 \$$	shift
(5) \$E + id <sub>2</sub>	$* id_3 \$$	reduce by $E \rightarrow id$
(6) \$E + E	$* id_3 \$$	shift
(7) \$E + E *	$id_3 \$$	shift
(8) \$E + E * id <sub>3</sub>	\$	reduce by $E \rightarrow id$
(9) \$E + E * E	\$	reduce by $E \rightarrow E * E$
(10) \$E + E	\$	reduce by $E \rightarrow E + E$
(11) \$E	\$	accept

**Ex 2:**

parse the input **id +\* id** for the same grammar

Stack	Input	Action
\$	id1 + * id2 \$	Shift
\$ id1	+ * id2 \$	Reduce by $E \rightarrow id$
\$ E	+ * id2 \$	Shift
\$ E+	*id2 \$	Shift
\$ E +*	id2 \$	Shift
\$ E +*id	\$	Reduce by $E \rightarrow id$
\$ E +*E	\$	<b>Not Accept</b>

H.W. : For this grammar

$E \rightarrow E+T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow id \mid (E)$

$E \rightarrow E+E \mid E * E \mid F$

Parse the input **id \* id + id**

### 2.6.7 Operator precedence parser

اسبقيات الرياضيات

في هذا الموضوع عند الحل يجب وضع الحروف او string المراد تحويلها بين عمليات رياضية معينة تعطى في السؤال

وكذلك اخذ بنظر الاعتبار عند التحويل اسبقيات الرياضيات. توجد طريقتين للحل في هذا الموضوع

1. الاعتيادية

2. باستخدام الـ Stack

Ex:- 1

Use operator precedence to check this sentence **id+id\*id** according to this grammar

$E \rightarrow E+E \mid E * E \mid id$

	id	+	*	\$
id		>	>	>
+	<	>	<	>
*	<	>	>	>
\$	<	<	<	

Sol:

في بداية الحل توضع في بداية الجملة المراد اعرابها بين \$ \$

\$ id+id\*id \$

\$ < id > + < id > \* < id > \$

\$ E + < id > \* < id > \$

\$ E + E \* < id > \$

\$ E + E \* E \$

\$ E + < E \* E > \$

\$ < E + E > \$

\$ E \$

➤ Stack implementation of operator precedence parser

Ex:- 2

Use Stack implementation of operator precedence parser to check this sentence id + id by this grammar:  $E \rightarrow E+E \mid E * E \mid id$

	id	+	*	\$
id		>	>	>
+	<	>	<	>
*	<	>	>	>
\$	<	<	<	

Sol:

Stack		Input
\$	<	id + id \$
\$ <id	>	+id \$
\$ <id >		+id \$
\$ <E+		+id \$
\$ <E+	<	id \$
\$ <E+ <id	>	\$
\$ <E+ <id >		\$
\$ <E+E >		\$
\$ E		\$ → accepted

H.w

Try input  $id * ( id \uparrow id ) - id / id$  with the following relations

	+	-	*	/	↑	id	(	)	\$
+	>	>	<	<	<	<	<	>	>
-	>	>	<	<	<	<	<	>	>
*	>	>	>	>	<	<	<	>	>
/	>	>	>	>	<	<	<	>	>
↑	>	>	>	>	<	<	<	>	>
id	>	>	>	>	>			>	>
(	<	<	<	<	<	<	<	=	
)	>	>	>	>	>			>	>
\$	<	<	<	<	<	<	<		