

Bottom up parsing (Right most Derivations)

- \* Terminals  $\rightarrow$  Non-Terminals  
leaf node  $\rightarrow$  Root Node (Start Symbol)

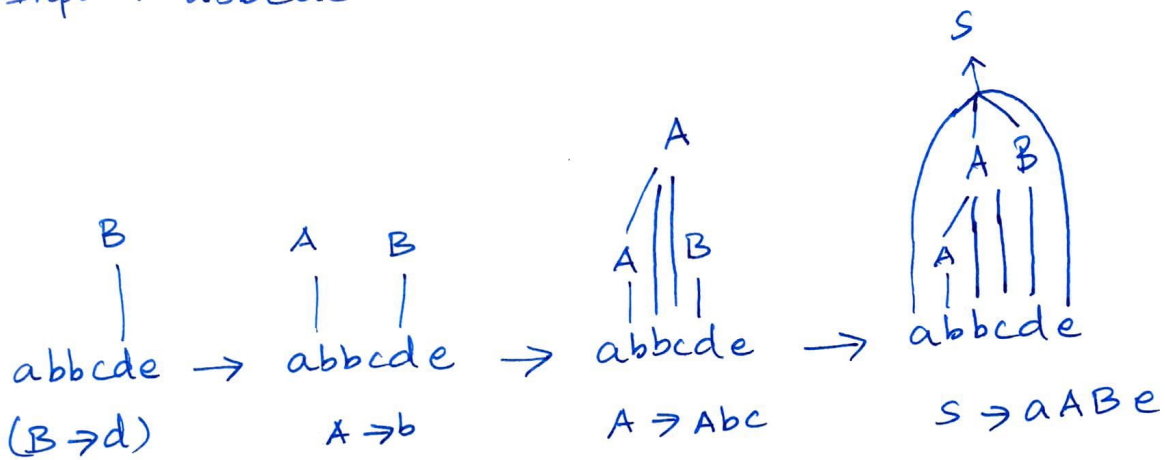
\* Example 1:

$S \rightarrow aABe$

$A \rightarrow Abc | b$

$B \rightarrow d$

Input: abcde

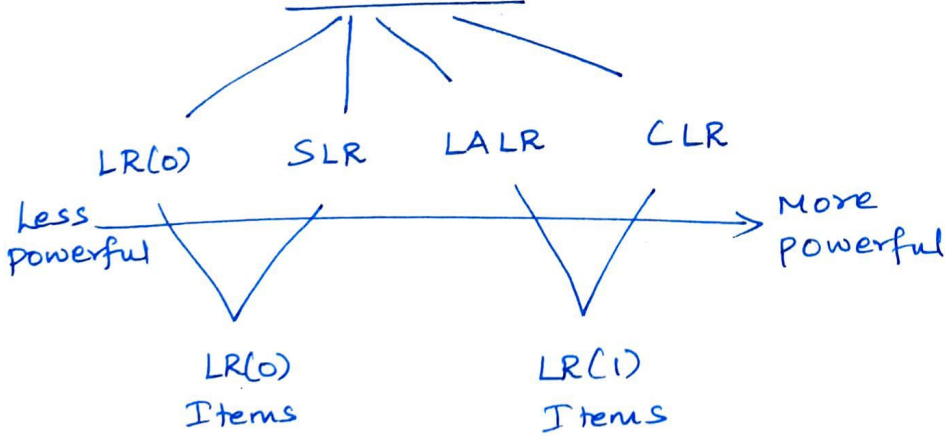


\* Types of Bottom up parsing

Non-Recursive Shift Reduce parser

Operator precedence parser

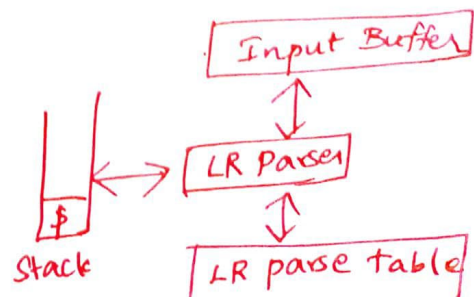
LR Parser



- SLR - Simple LR
- LALR - lookahead LR
- CLR - Canonical LR

LR(k)

- $\rightarrow$  Left to Right processing i/p
- $\rightarrow$  Right most Derivations
- $\rightarrow$  k (lookahead)



# LR(0) parser

Q.  $S \rightarrow AA$   
 $A \rightarrow aA|b$

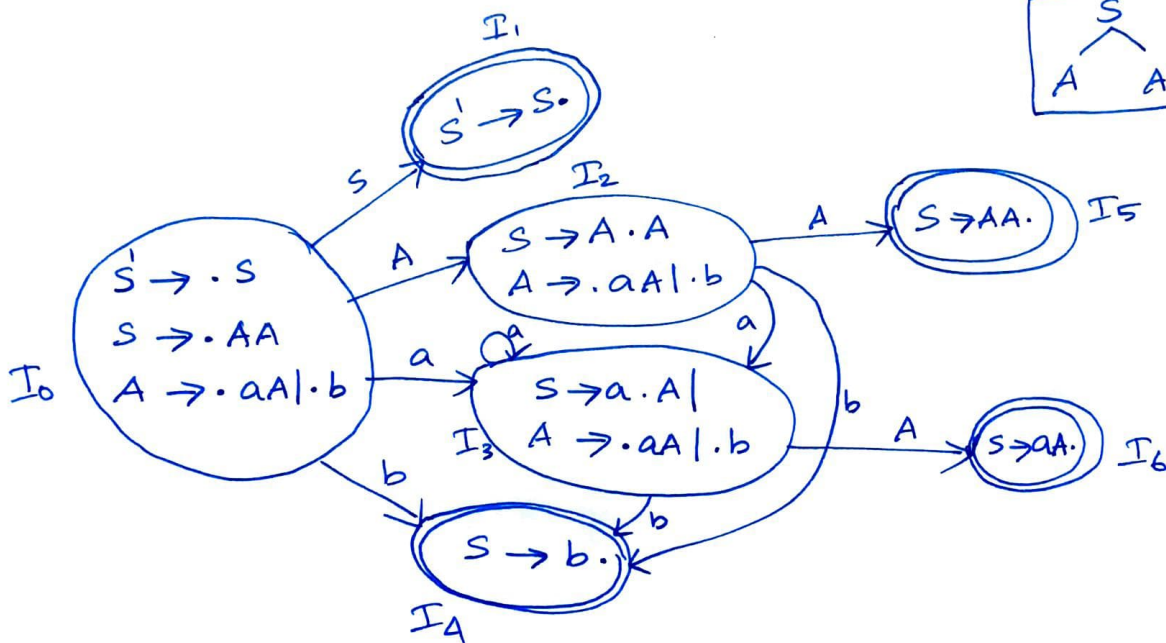
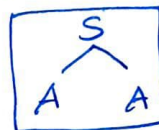
## Step 1: Augmented Grammar

$S' \rightarrow S$   
 $S \rightarrow AA$   
 $A \rightarrow aA|b$

## Step 2: Canonical LR(0) items

LR(0) items  $\rightarrow$  productions  $\rightarrow$

- $S \rightarrow \underline{\cdot}AA$  (yet to read)
- $S \rightarrow A \cdot A$  (want to read A)
- $S \rightarrow AA \cdot$  (Completing read).



### No. The production

- 1  $S \rightarrow AA$
- 2  $A \rightarrow aA$
- 3  $A \rightarrow b$

## Step 3: parse Table

	Goto (Non-Termi)		Action (Terminals)		
	A	S	a	b	\$
I <sub>0</sub>	2	1	S <sub>3</sub>	S <sub>4</sub>	
I <sub>1</sub>					Accept
I <sub>2</sub>	5		S <sub>3</sub>	S <sub>4</sub>	
I <sub>3</sub>	6		S <sub>3</sub>	S <sub>4</sub>	
I <sub>4</sub>			r <sub>3</sub>	r <sub>3</sub>	r <sub>3</sub>
I <sub>5</sub>			r <sub>1</sub>	r <sub>1</sub>	r <sub>1</sub>
I <sub>6</sub>			r <sub>2</sub>	r <sub>2</sub>	r <sub>2</sub>

state  $\xrightarrow{T}$  shift  
state  $\xrightarrow{N-T}$  Goto  
Final state  $\rightarrow$  Red

"aabb"

STACK	INPUT	ACTION
\$ <u>0</u>	<u>a</u> abb\$	
\$0a <u>3</u>	<u>a</u> bb\$	shift a <sub>3</sub>
\$0a3a <u>3</u>	<u>b</u> bb\$	shift a <sub>3</sub>
\$0a3a3 <u>b</u> <sub>4</sub>	<u>b</u> \$	shift b <sub>4</sub>
\$0a3 <span style="border: 1px solid black; padding: 2px;">a3A6</span>	<u>b</u> \$	Reduce (R <sub>3</sub> ) 3 <sup>rd</sup> production (3A → b)
\$0a3A <u>6</u>	<u>b</u> \$	Reduce (R <sub>2</sub> ) 2. A → aA
\$0A <u>2</u>	<u>b</u> \$	Reduce (R <sub>2</sub> )
\$0A2 <span style="border: 1px solid black; padding: 2px;">b4</span>	<u>\$</u>	shift b <sub>4</sub>
\$0A2A <u>5</u>	<u>\$</u>	Reduce (R <sub>3</sub> ) (3. A → b)
\$0 <u>S</u> <sub>1</sub>	<u>\$</u>	Reduce (R <sub>1</sub> ) (1. S → AA)

Accept.

Step 5: parse Tree.

aabb

