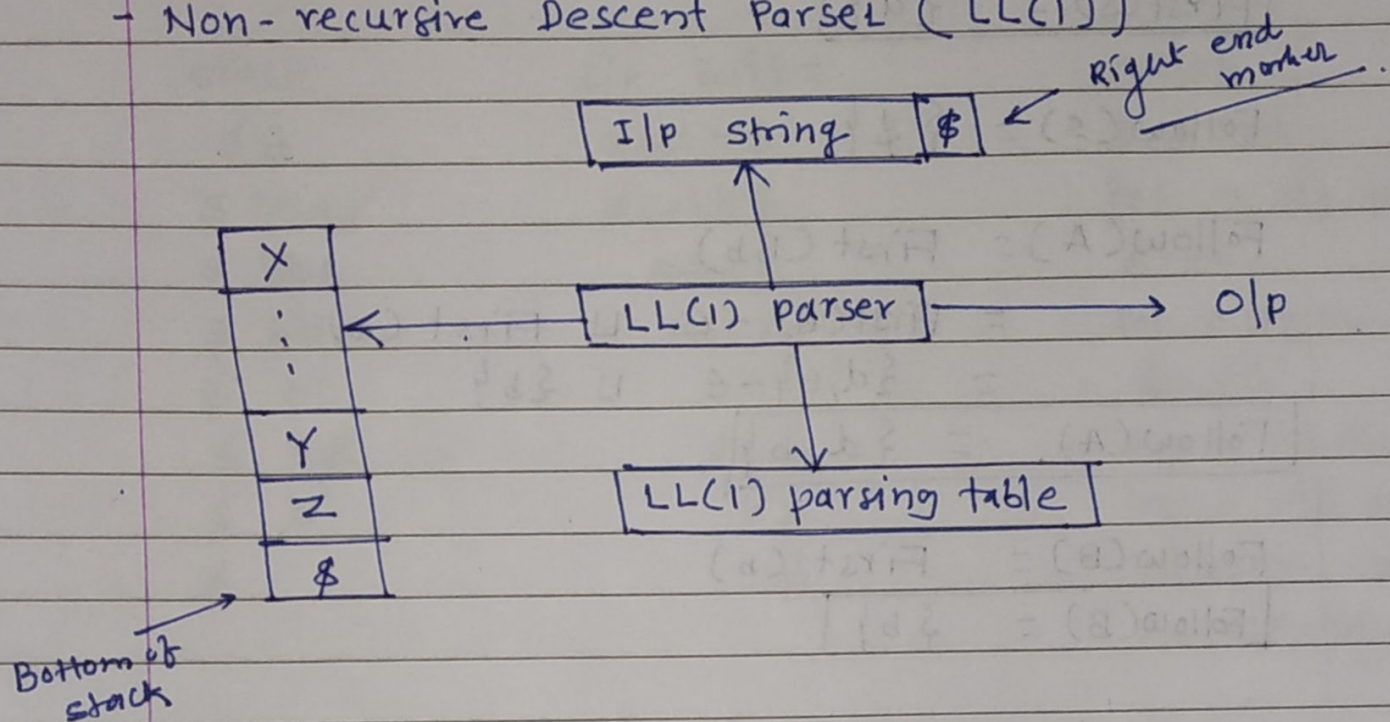


Predictive Parser.

- Non-recursive Descent Parser (LL(1))



- 1) Design LL(1) parser and derive the string "acdb" for LL(1) grammar.

$$S \rightarrow aABb$$

$$A \rightarrow c \mid \epsilon$$

$$B \rightarrow d \mid \epsilon$$

$$\text{First}(B) = \text{First}(d)$$

$$= \{d\}$$

$$\text{First}(B) = \text{First}(\epsilon)$$

$$= \{\epsilon\}$$

$$\therefore \boxed{\text{First}(B) = \{d, \epsilon\}}$$

$$\text{First}(A) = \text{First}(c)$$

$$= \{c\}$$

$$\text{First}(A) = \text{First}(\epsilon)$$

$$= \{\epsilon\}$$

$$\boxed{\text{First}(A) = \{c, \epsilon\}}$$

$$\text{First}(S) = \text{First}(aABb)$$

$$\boxed{\text{First}(S) = \{a\}}$$

$$\text{Follow}(S) = \{\$ \}$$

$$\text{Follow}(A) = \text{First}(Bb)$$

$$= \text{First}(B) - \epsilon \cup \text{First}(b)$$

$$= \{d, \epsilon\} - \epsilon \cup \{b\}$$

$$\boxed{\text{Follow}(A) = \{d, b\}}$$

$$\text{Follow}(B) = \text{First}(b)$$

$$\boxed{\text{Follow}(B) = \{b\}}$$

Parsing Table

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

Parsing Table does not contain multiple entries,
Thus grammar is LL(1)

Derivation of string "acdb"

Page No.

Date

stack

I/p Buffer

Action

\$S

acdb\$

$S \rightarrow aABb$

\$bBA~~a~~

~~a~~cdb\$

Pop a off the stack

\$bBA

cd b\$

$A \rightarrow c$

\$bB~~a~~

~~a~~cd b\$

Pop c off the stack

\$bB

db\$

$B \rightarrow d$

\$b~~a~~

~~a~~db\$

pop d off the stack

\$~~a~~

~~a~~b\$

pop b off the stack

\$

\$

Successful completion of parsing.