Om Sai Ram

LR(K) parser:-

The canonical set of items is the parsing technique in which a lookahead symbol is generated while constructing set of items. Hence cannonical the collection of set of items is referred as LR(1).

1 indicates one lookahead symbol in the set of items.

construction of cannonical set of items along with the lookahead:-

① For the Grammar G initially add $S' \to \cdot S$ in the set of items.

② For each set of items ~~Ifound~~ I_i in C and for each grammar symbol X (may be terminal or non-terminal) add closure (I_i, X).

This process should be repeated by applying $goto(I_i, X)$ for each X in I_i such that $goto(I_i, X)$ is not empty and not in C.

The set of items has to constructed until no more set of items can be added to C.

③ The closure function can be computed as follows

① For each item $A \to \alpha \cdot X \beta a$ and rule $X \to \gamma$ and $b \in First(\beta a)$ such that $X \to \cdot \gamma$ and b is not in I then add

$X \to \cdot \gamma, b$ to I.

④ Similarly the goto function can be computed as:

For each item $[A \to \alpha \cdot X \beta, a]$ is in I and rule $[A \to \alpha X \cdot \beta, a]$ is not in goto items then add $[A \to \alpha X \cdot \beta, a]$ to goto items.

Construction of cannonical LR parsing table:-

I/P:- An Augmented grammar G'

O/P:- The cannonical LR parsing table.

Algorithm:-

1. Initially construct set of items $C = \{I_0, I_1, I_2 \cdots I_n\}$ where C is a collection of set of LR(1) items for the i/p Grammar G'.

2. The parsing actions are based on each Item I_i.

   a. if $[A \to \alpha \cdot a\beta, b]$ is in I_i and $goto(I_i, a) = I_j$ then create an entry in the action table $action[I_i, a] = \underset{\text{shift } j}{\underbrace{S_j}}$

   b. if there is a production $[A \to \alpha \cdot, a]$ in I_i then in the action table $action[I_i, a] =$ reduce by $A \to \alpha$.
      Here A should not be S'.

   c. If there is a production $S' \to S \cdot, \$$ in I_i then $action[i, \$] = accept$.

3. The goto part of the LR table can be filled as
The goto transitions for state i is considered for non-terminals only.
   If $goto(I_i, A) = I_j$ then $goto[I_i, A] = j$.

4. All the entries not defined by rule 2 and 3 are considered to be "error".

① $S \to CC$
② $C \to aC$ construct LR(1) set of items
③ $C \to d$

→ we will initially add $S' \to \bullet S, \$$ as the first rule in I_0.

$S' \to \bullet S, \$$ with

$[A \to \alpha \bullet X\beta, a]$

$A = S', \alpha = \varepsilon, X = S, \beta = \varepsilon, a = \$$

If there is a production $X \to \gamma$, b then add $X \to \bullet\gamma, b$.

$\therefore S \to \bullet CC$ $\quad b \in FIRST(\beta)$

$b \in FIRST(\varepsilon\$)$ as $\varepsilon\$ = \$$

$b \in FIRST(\$)$.

$b = \{\$\}$

$S \to \bullet CC, \$$ will be adde to I_0

Now $S \to \bullet CC, \$$ is in I_0 we will match it with $A \to \alpha \bullet X\beta, a$

$S \to \bullet CC, \$$

$A \to \alpha \bullet X\beta, a$

$A = S, \alpha = \varepsilon, X = C, \beta = C, a = \$$

If there is a production $X \to \gamma, b$ then add $X \to \bullet\gamma, b$.

$C \to \bullet aC$ $\quad b \in FIRST(\beta a)$

$b \in FIRST(C\$)$

$C \to \bullet d$ $\quad b \in FIRST(C)$

$b = \{a, d\}$

$C \to \bullet aC$, a or d will be added in I_0

Hence I_0:

$S' \to \bullet S, \$$
$S \to \bullet CC, \$$
$C \to \bullet aC, a|d$
$C \to \bullet d, a|d$

Now apply goto on I_0

$S' \to \bullet S, \$$
$S \to \bullet CC, \$$
$C \to \bullet aC, a|d$
$C \to \bullet d, a|d$

Hence goto (I_0, S)

$\boxed{I_1 : S' \to S\bullet, \$}$

Now apply goto on C in I_0

$S \to C \bullet C, \$$ add in I_2.

$A \to \alpha \bullet X\beta, a$ $\quad \alpha = C, X = C, \beta = \varepsilon, a = \$$

$X \to \bullet\gamma, b$ where $b \in FIRST(\beta a)$

$C \to \bullet aC$ $\quad b \in FIRST(\varepsilon\$)$

$C \to \bullet d$ $\quad b \in FIRST(\$)$

Hence $C \to \bullet aC, \$$
$C \to \bullet d, \$$ will be added to I_2

Hence I_2: goto (I_0, C)

$S \to C \bullet C, \$$
$C \to \bullet aC, \$$
$C \to \bullet d, \$$

goto (I_0, a)

$C \to a \bullet C, a|d$
$C \to \bullet aC$
$C \to \bullet d$.

$C \to a \bullet C, a|d$. $\quad \alpha = a, X = C, \beta = \varepsilon, a = a|d$

$A \to \alpha \bullet X\beta, a$

$C \to \bullet aC$ $\quad b \in first(\beta a)$

$C \to \bullet d$ $\quad FIRST(\varepsilon . a$ or $d)$

I_3:- goto(I_0, a)

$$\boxed{\begin{array}{l} C \to a \cdot C, a/d \\ C \to \cdot aC, a/d \\ C \to \cdot d, a/d \end{array}}$$

I_4: goto(I_0, d)

$$\boxed{C \to d \cdot a/d}$$

I_5: goto(I_2, C)

$$\boxed{S \to CC \cdot, \$}$$

I_6: goto(I_2, a)

$C \to a \cdot C, \$$

$A \to \alpha \cdot X \beta, a$

$A = C, \alpha = a, X = C,$

$\beta = \varepsilon, a = \$$

$X \to \cdot \gamma$

$C \to \cdot aC$ and $C \to \cdot d$

$b \in FIRST(\beta a)$

$b \in FIRST(\varepsilon \cdot \$)$

$b = (\$)$

$$\boxed{\begin{array}{l} I_6 : \text{goto}(I_2, a) \\ C \to a \cdot C, \$ \\ C \to \cdot aC, \$ \\ C \to \cdot d, \$ \end{array}}$$

$$\boxed{\begin{array}{l} I_7 : \text{goto}(I_2, d) \\ C \to d \cdot, \$ \end{array}}$$

$$\boxed{\begin{array}{l} I_8 :\text{- goto}(I_3, C) \\ C \to aC \cdot, a/d \end{array}}$$

$$\boxed{\begin{array}{l} I_9 :\text{- goto}(I_6, C) \\ C \to aC \cdot, \$ \end{array}}$$

Now we can construct LR parsing table.

Construction of canonical LR parsing Table:-

Now consider I_0 in which there is a rule matching with $[A \to \alpha \cdot a\beta, b]$ as $C \to \cdot aC$, a/d and if the goto is applied on 'a' then we get the state I_3.

Hence we will create entry action$[0, a]$ = shift 3.

Similarly, in I_0

$C \to \cdot d$ a/d

$A \to \alpha \cdot a\beta, b$

$A = C, \alpha = \varepsilon, a = d, \beta = \varepsilon, b = a/d$

goto$(I_0, d) = I_4$.

Hence action$[0, d]$ = shift 4

For state I_4.

$C \to d \cdot, a/d \quad A = C, \alpha = d, a = a/d$

$A \to \alpha \cdot, a$

action$[4, a]$ = reduce by $C \to d$ i.e; rule 3

$S' \to S \cdot, \$$ in I_1

So we will create action$[1, \$]$ = accept.

The goto table can be filled by using the goto functions

$I_0 \xrightarrow{S} I_1$ goto$(I_0, S) = I_1$ Hence goto $[0, S] = 1$

$I_0 \xrightarrow{C} I_2$ goto$(I_0, C) = I_2$

$I_2 \xrightarrow{C} I_5$ goto$(I_2, C) = I_5$

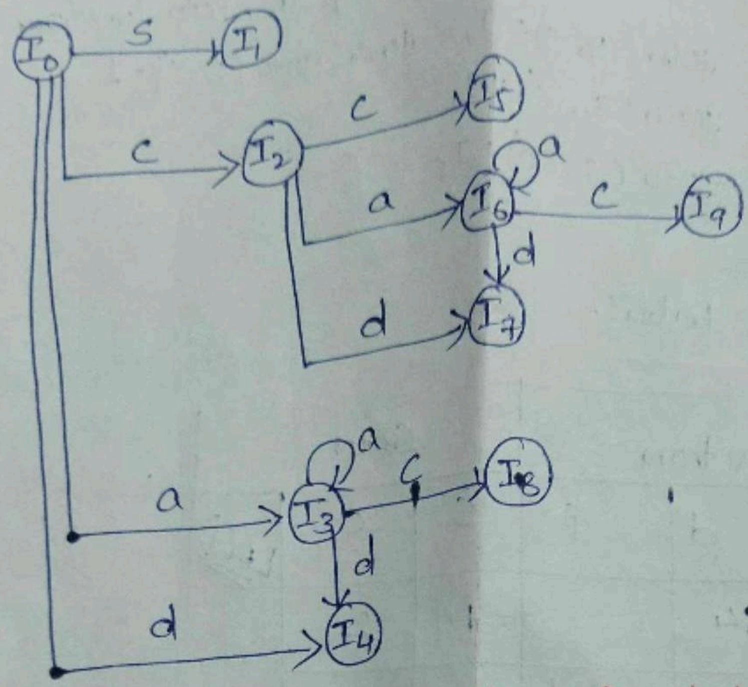
LR(1) parsing table:-

	Action			Goto	
	a	d	$	S	C
0	S3	S4		1	2
1			Accept		
2	S6	S7			5
3	S3	S4			8
4	r3	r3			
5			r1		
6	S6	S7			9
7			r3		
8	r2	r2			
9.			r2		

LR(1)

remaining blank entries in the table are considered as syntatical error.

DFA:-



parsing the i/p using LR(1) parsing table:-

Stack	input buffer	Action table	Goto table	parsing Action
$0	aadd$	action[0,a]=S3		shift
$0a3	add$	action[3,a]=S3		shift
$0a3a3	dd$	action[3,d]=S4		shift
$0a3a3d4	d$	action[4,d]=r3	[3,C]=8	Reduce by C→d
$0a3a3C8	d$	action[8,d]=r2	[3,C]=8	Reduce by C→aC
$0a3C8	d$	action[8,d]=r2	[0,C]=2	Reduce by C→aC
$0C2	d$	action[2,d]=S7		shift
$0C2d7	$	action[7,$]=r3	[2,C]=5	Reduce by C→d
$0C2C5	$	action[5,$]=r1	[0,S]=1	Reduce by S→CC
$0S1	$	accept		