Ques. Consider the grammar

$S' \rightarrow CC$

$C \rightarrow aC \mid d$

(i) Construct LR(1) set of items for the above grammar

(ii) Construct LR(1) parsing table

(iii) Parse the string aadd with the help of LR(1) table.

Solution

$S \rightarrow CC$

$C \rightarrow aC \mid d$

Step-1 First make the augmented grammar by adding production $S' \rightarrow S$.

The purpose of this grammar is to indicate the acceptance of input. That is when parser is about to reduce $S' \rightarrow S$ it reaches to acceptance state.

Step-2 Now the augmented grammar becomes

$$\left.\begin{array}{l} S' \rightarrow S \\ S \rightarrow CC \\ C \rightarrow aC \\ C \rightarrow d \end{array}\right\} \text{augmented grammar}$$

Now place (•) at first position of R.H.S in each production

$S' \to \cdot S \ , \ \underline{\$} \longrightarrow$ lookahead symbol which is calculated in LR(1) & which makes it different from LR(0).

$S \to \cdot CC$

$C \to \cdot aC$

$C \to \cdot d$

* Now it is noted that lookahead of starting production i.e. production which we add to make the augmented grammar is \$ by default.

$S' \to \cdot S \ , \ \underline{\$}$

$S \to \cdot CC \ , \ \$$

$C \to \cdot aC \ , \ -$

$C \to \cdot d \quad -$

For calculating lookahead for other production we compare $S' \to \cdot S \ , \ \underline{\$}$ with

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

we get $A = S' \ , \ X = S \ , \ \beta = \epsilon \ , \ a = \$$

Now calculate $FIRST(\beta a) = FIRST(\epsilon \$)$

$= FIRST(\$)$

$= \{\$\}$

Scanned by CamScanner

$S' \rightarrow \cdot S, \underline{\$}$ ← by default

$S \rightarrow \cdot CC, \underline{\$}$ ← Calculated by using above production

$C \rightarrow \cdot aC, \underline{\quad}$

$C \rightarrow \cdot d, \underline{\quad}$

Now, again do the same process for calculation of lookahead for production $C \rightarrow \cdot aC$

$$S \rightarrow \cdot CC, \$$$

Compare with

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

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

$$\begin{aligned} FIRST(\beta a) &= FIRST(C \cdot \$) \\ &= FIRST(C) \\ &= \{a, d\} \end{aligned}$$

← using rule for calculation of FIRST from grammar.

For every product which start with C copy this lookahead i.e.

$$C \rightarrow \cdot aC, \underline{a/d}$$

$$C \rightarrow \cdot d, \underline{a/d}$$

Now, the complete picture becomes

$$\boxed{\begin{aligned} &S' \rightarrow \cdot S, \underline{\$} \\ &S \rightarrow \cdot CC, \underline{\$} \\ &C \rightarrow \cdot aC, \underline{a/d} \\ &C \rightarrow \cdot d, \underline{a/d} \end{aligned}}$$

State I_0

Scanned by CamScanner

~~Step~~-

Now apply goto on state I_0.

goto (I_0, S) [means shift the dot one place to right where (•) is followed by S]

$$\boxed{S' \rightarrow S\cdot, \$}\quad I_1$$

goto (I_0, C) [since (•) is followed by non terminal after applying goto then repeat the production of non-terminal mentioned in I_0.]

$S \rightarrow C \cdot C, \$$

$C \rightarrow \cdot aC, \$$

$C \rightarrow \cdot d, \$$

(by default since it is first production)

for calculation of lookahead for this production

Compare $S \rightarrow C \cdot C, \$$ with

$A \rightarrow \alpha \cdot X\beta, a$, we get

$A = S, X = C, \beta = \epsilon, a = \$$

$FIRST(\beta a) = FIRST(\epsilon \$)$

$= FIRST(\$)$

$= \{\$\}$

Now copy this lookahead in all production with derive from C. therefore

$$\boxed{\begin{array}{l} S \rightarrow C \cdot C, \$ \\ C \rightarrow \cdot aC, \$ \\ C \rightarrow \cdot d, \$ \end{array}}\quad I_2$$

goto (I_0, a)

$C \to a \cdot C$, $\underline{a|d}$ — This lookahead copy from the I_0.

These productions repeat as in first production (.) is followed by non-terminal C therefore repeat production derive by C in state I_0

$C \to \cdot aC$; $\underline{a|d}$

$C \to \cdot d$, $\underline{a|d}$

This lookahead copy from above production because we copy lookahead for all production derive by same non-terminal.

$$\boxed{\begin{array}{l} C \to a \cdot C, \ \underline{a|d} \\ C \to \cdot aC, \ \underline{a|d} \\ C \to \cdot d, \ a|d \end{array}} \quad I_3$$

goto (I_0, d)

$$\boxed{C \to d \cdot, \ \underline{a|d}} \quad I_4$$

$\therefore$ After applying goto on state I_0 we get I_1, I_2, I_3 & I_4

similarly apply goto on state I_1, I_2, I_3 & I_4 & calculate new states. as

Note: since goto is not applicable on state I_1 as $(\cdot)$ is already at last in R.H.S. therefore.

goto (I_2, C)

$$\boxed{S \rightarrow CC\cdot, \$} \quad I_5$$

This lookahead copy from state I_2.

goto (I_2, a)

$$\boxed{\begin{array}{l} C \rightarrow a\cdot C, \$ \\ C \rightarrow \cdot aC, \$ \\ C \rightarrow \cdot d, \$ \end{array}} \quad I_6$$

This situation again becomes that $(\cdot)$ is followed by non-terminal after applying goto therefore repeat production of non-terminal from I_0.

As these production derives by non-terminal 'C' therefore write or copy lookahead of first production.

goto (I_2, d)

$$\boxed{C \rightarrow d\cdot, \$} \quad I_7$$

This is new state because this lookahead is different from lookahead in I_4.

After applying goto on state I_2 we get I_5, I_6 & I_7.

→ Now apply goto on state I_3.

goto (I_3, C)

$$\boxed{C \rightarrow aC\cdot, a/d} \quad I_8$$

goto (I_8, a)

$$\left\{\begin{array}{ll} C \to a \cdot C, & \underline{a/d} \\ C \to \cdot aC, & \underline{\cancel{\$}\,a/d} \\ C \to \cdot d, & a/d \end{array}\right\}$$

This state is I_3 repeated as state I_3 so need not give as new state.

goto (I_8, d)

$C \to d\cdot, \; a/d \} \; I_4$ [this is also repeated as state I_4]

After goto on state I_3 we get new state I_8 & old state I_3 & I_4.

Now again apply goto on state I_4 but in this state $(\cdot)$ is already at last place in R.H.S. so need of goto on state I_4 similarly in state I_5 therefore apply goto on state I_6

goto (I_6, C)

$\boxed{C \to aC\cdot, \; \underline{\$}}$ I_9 (New state)

goto (I_6, a)

$$\left.\begin{array}{ll} C \to a \cdot C, & \underline{\$} \\ C \to \cdot aC, & \underline{\$} \\ C \to \cdot d, & \underline{\$} \end{array}\right\} \leftarrow \text{old state } (I_6)$$

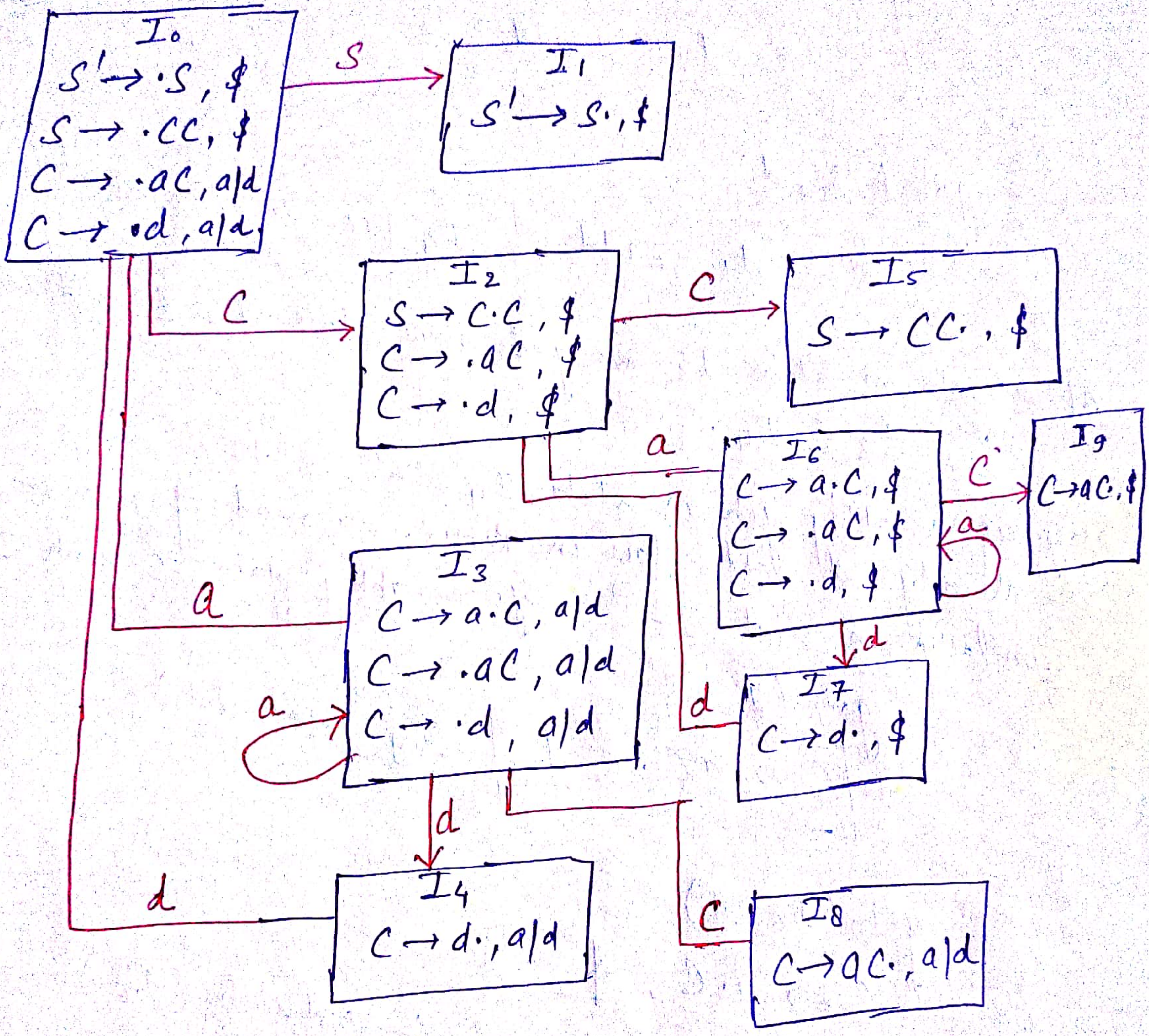
goto (I_6, d)

$C \to d\cdot, \; \$ \} \leftarrow$ old state (I_7)

i.e., after applying goto on state I_6 we get new state I_9 & old state I_6 & I_7.

Scanned by CamScanner

Now goto on state I_7, I_8 & I_9 produce no states because there is no scope of shifting of (•) in R.H.S. therefore after completion of step-2.



The GOTO graph for Grammar

This is the answer of LR(1) items of the given grammar.