Problem

Let G be the following grammar:

$$S \rightarrow T\dashv$$
$$T \rightarrow T\mathtt{a}T\mathtt{b} \mid T\mathtt{b}T\mathtt{a} \mid \varepsilon$$

a. Show that $L(G) = \{w\dashv|$ w contains equal numbers of a's and b's}. Use a proof by induction on the length of w.

b. Use the *DK*-test to show that G is a DCFG.

c. Describe a DPDA that recognizes L(G).

Step-by-step solution

Step 1 of 7

Consider the following grammar:

$S \rightarrow T \dashv$
$T \rightarrow TaTb \mid TbTa \mid \varepsilon$

It is required to show that $L(G) = \{w \dashv | \; w\}$

Comment

Step 2 of 7

• Here, it can be prove by the induction that the string w contains an equal number of a 's and b's. Induction can be done in two ways either from the left side of the induction or from the right side of the induction.

• In order to prove $L(G) = \{w \dashv | \; w\}$, first ε handles occurs in reduction as it is indicated by the underscores in the leftmost reduction of the string $(\;)(\;)\dashv$.

Comment

Step 3 of 7

• It is being known that $T \rightarrow \varepsilon$ and thus $T \dashv \rightarrow \varepsilon$ and $S \rightarrow \varepsilon$.

Consider the string w contains $aaabbb$. Each and every string which is being generated by the grammar *G* is in *L*.

$aaabbb \rightarrowtail aaSbb \rightarrowtail aSb \rightarrowtail S \rightarrowtail T \dashv$
$w \in T \dashv \in S \in L(G)$

Similarly in the second case, $w \dashv \in T \in S$

• Hence, it is being proved that any language G contains equal number of a's and b's in string w and $w \dashv$

Comment

Step 4 of 7

Using the *DK*-test it is to be shown that the grammar belongs to *DCFG*.

DK test depend upon one simple and surprising fact. *DK* is used for accepting an input z only if these two conditions are met.

• z is the prefix of the valid string $v = zw$.

- z should be end with a handle of v.

Comment

Step 5 of 7

Consider the grammar $L(G)=\{w \dashv w\}$ which contains an equal number of a's and b's.

- So, if the grammar contains an equal number of a's and b's then the grammar $a^n b^n$ becomes $a^n b^n$. In case of *DCFG* all the handles are being forced. So, if zw is a one of the valid string with the prefix z that will end for handling zw'.
- In case of *DCFG* all the handles are being forced. So, if zw is a one of the valid string with the prefix z that will end for handling zw'. For all this property *DK* must be capable of handling all the accept states. But, the condition is that outgoing path should not be there.

Comment

Step 6 of 7

- Here, the construction of *DK* test is done for concluding that *G* is deterministic if the property of accept state is satisfied. For the construction of the *DFA*, first *NFA* need to be constructed. After the construction it is found that no outgoing path is there from the start state.
- Here, $aaabbb$ is a valid string and it handles zw so after the reduction of the rule it becomes: $T \rightarrow z.w$. Grammar must contain the following rules:

$$\begin{aligned} &S_1 \rightarrow z_1 S_2 w_1 \\ &S2 \rightarrow z_2 S_3 w_2 \\ &\quad\vdots \\ &S_l \rightarrow z_l T w_l \\ &T \rightarrow zw \end{aligned}$$

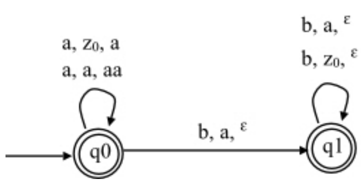
K contains the path from the start state to the end state. First the input $z = xu$ is being read after that transition is done and shifting of the string is done. After the shifting it is being found that there is no outgoing path from the start state.

Hence with the help of *DK* test it is proved that grammar belong to *DCFG*.

Comment

Step 7 of 7

The DPDA that recognizes the L(G) is as follows:



Comment