Problem

Modify the algorithm for context-free language recognition in the proof of Theorem 7.16 to give a polynomial time algorithm that produces a parse tree for a string, given the string and a CFG, if that grammar generates the string.

THEOREM **7.16**

Every context-free language is a member of P.

Step-by-step solution

Step 1 of 3

Parse tree is a tree of any grammar $G=(V,T,R,S)$ where start symbol *S* is root, non-terminals are interior nodes and terminals are leaf nodes. Each terminal string which is generated by some grammar *G* has parse tree.

Comment

Step 2 of 3

Here, consider the CFG *G* where,

• *V* is variables set.

• *T* is terminals or leaf nodes.

• *R* is rule of production.

• *S* is starting symbol.

Comment

Step 3 of 3

The Theorem 7.16 states that every context free language is a member of P. There is an algorithm for the Theorem 7.16. Modify the algorithm to produce a polynomial time algorithm that produces a parse tree for a string using the CFG.

The algorithm is as follows:

$D=$"On input $s=s_1s_2....s_n$ and CFG G
    ParseTree $\leftarrow$ (Initial S tree, Beginning of the string s)
    CurrentState $\leftarrow$ POP the ParseTree
    repeat *for* $i=1,2,3,4....n$
        if parsing is successful for CurrentState
            return TREE(CurrentState)
        else
            //Expand the CurrentState
            if no nodes to expand
                return reject
            else
                //Apply the rules to find the next node
                temp $\leftarrow$ Apply the CFG rules on CurrentState
                PUSH temp into the ParseTree
        if ParseTree is empty
            return reject
        else
            // Move to the next part of the input
            CurrentState $\leftarrow$ NEXT(ParseTree)
    return ParseTree

The above algorithm takes the string and apply the rules on it. For each part of the string *s*, apply the rules and find the next node (i.e., state). Each new state is added to the tree. If the input is processed completely then it returns the parse tree.