Date: 07/08/2024

Experiment 3.1

AIM

To simulate FIRST and FOLLOW of a grammar

ALGORITHM

1. Start
2. Read grammar G with n non terminals and m terminals.
3. Each non terminal is represented as a number between 0 to n-1 and each terminal as a number between 0 to m-1.
4. Initialize a n*(m+1) 2D array of boolean values to store FIRST and FOLLOW of each non-terminal where the element at index (i,j) represents whether the terminal j is part of FIRST or FOLLOW of i. The index m+1 for terminals represents special characters 'ε' and '$' in FIRST and FOLLOW respectively.
5. Find FIRST of all non terminals by repeating the following till no changes occur to the FIRST set in a given iteration:
    1. For each production rule of form $X \rightarrow Y1Y2...Yk$:
        1. If the production of the form $X \rightarrow \varepsilon$:
            1. Add ε to FIRST(X).
            2. Continue to next iteration.
        2. Set nullable = true.
        3. For i from 1 to k:
            1. If Y_i is a terminal:
                1. Add Y_i to FIRST(X).
                2. Set nullable = false.
                3. Break.
            2. Add all terminals in FIRST(Y_i) to FIRST(X) except 'ε'.
            3. If $\varepsilon \notin$ FIRST(Y_i):
                1. Set nullable = false.
                2. Break.

4. If nullable, add 'ε' to FIRST(X).

6. Find FOLLOW of all non terminals by repeating the following till no changes occur to the FOLLOW set in a given iteration:
    1. Add '$' to FOLLOW(S) where S is the starting symbol.
    2. For each production rule of form $X \rightarrow Y1Y2...Yk$:
        1. For i from 1 to k:
            1. Set nullable = true.
            2. For j from i+1 to k:
                1. If Y_j is a terminal:
                    1. Add Y_j to FOLLOW(Y_i).
                    2. Set nullable = false.
                    3. Break.
                2. Add all terminals in FIRST(Y_j) to FOLLOW(Y_i) except 'ε'.
                3. If $\varepsilon \notin$ FIRST(Y_j):
                    1. Set nullable = false.
                    2. Break.
            3. If nullable, add all terminals in FOLLOW(X) to FOLLOW(Y_i) including '$'.
7. Display FIRST and FOLLOW of each non-terminal.
8. Stop

RESULT

Successfully found FIRST and FOLLOW of all non terminals in a given grammar