**Implement a C program to eliminate left factoring from a given CFG**?

#include <stdio.h>

#include <string.h>

#define MAX 100

void eliminateLeftFactoring(char grammar[][MAX], int n) {

for (int i = 0; i < n; i++) {

char nonTerminal = grammar[i][0];

char commonPrefix[MAX], remaining[MAX][MAX];

int commonLength = 0, remainingCount = 0;

// Find the longest common prefix

char \*production = strtok(grammar[i], " ");

production = strtok(NULL, " ");

strcpy(commonPrefix, production);

commonLength = strlen(production);

while (production != NULL) {

for (int j = 0; j < commonLength; j++) {

if (commonPrefix[j] != production[j]) {

commonPrefix[j] = '\0';

commonLength = j;

break;

}

}

production = strtok(NULL, " ");

}

if (commonLength > 0) {

printf("%c -> ", nonTerminal);

printf("%.\*s%c'\n", commonLength, commonPrefix, nonTerminal);

production = strtok(grammar[i], " ");

production = strtok(NULL, " ");

while (production != NULL) {

if (strncmp(commonPrefix, production, commonLength) == 0) {

strcpy(remaining[remainingCount++], &production[commonLength]);

}

production = strtok(NULL, " ");

}

printf("%c' -> ", nonTerminal);

for (int j = 0; j < remainingCount; j++) {

if (strlen(remaining[j]) == 0) {

printf("ε ");

} else {

printf("%s ", remaining[j]);

}

}

printf("\n");

} else {

printf("%s\n", grammar[i]);

}

}

}

int main() {

int n;

printf("Enter the number of productions: ");

scanf("%d", &n);

char grammar[MAX][MAX];

printf("Enter the productions (e.g., E E+T | E-T | T): \n");

for (int i = 0; i < n; i++) {

scanf(" %[^\n]", grammar[i]);

}

printf("The grammar after eliminating left factoring:\n");

eliminateLeftFactoring(grammar, n);

return 0;

}

