**Experiment No. 3: Write a program to check whether a string belong to grammar.**

**Theory:**

**Grammar Overview:**

* **Production Rules**: A grammar specifies a set of production rules that define how valid strings in a language are formed.
* **Terminals and Non-terminals**:
  + **Terminals** are the actual symbols (characters or tokens) used in the language.
  + **Non-terminals** represent groups or sequences of symbols and are used to derive terminals.

**String Checking Process:**

1. **Start with the Initial Symbol**:
   * The process begins with the initial symbol of the grammar, typically denoted by a non-terminal symbol.
2. **Apply Production Rules**:
   * Production rules are applied recursively, replacing non-terminals with symbols according to the defined rules.
3. **Generate a String**:
   * This process continues until a string of terminal symbols (actual characters) is produced.
4. **Compare with Input String**:
   * The generated string is then compared with the input string to check if they match.

**Example:**

Consider a simple grammar:

* **Initial Symbol**: S
* **Production Rules**:
  + S→aS
  + S→Sb
  + S→ab

Let's check if the string "aab" belongs to this grammar:

* **Start**:
  + Begin with the initial symbol S.
* **Apply Production Rules**:
  + S→aS (replace S with aS)
  + aS→aaS (replace S with aS again)
* **Generate String**:
  + Continue applying the rules: aaS→aab
* **Compare with Input String**:
  + The generated string "aab" matches the input string "aab". Hence, the string belongs to the grammar.

**String Rejection:**

If, during the process, no production rule can be applied or the generated string doesn't match the input string, then the string does not belong to the grammar.

**Conclusion:**

Checking whether a string belongs to a grammar involves recursively applying production rules to generate a string. If the generated string matches the input string, it belongs to the grammar. If not, it doesn't. This method is fundamental to the theory of formal languages and automata.

**C++ Code:**

#include <iostream>

#include <cctype>

bool isDigit(char c) {

return std::isdigit(c);

}

bool isOperator(char c) {

return c == '+' || c == '-' || c == '\*' || c == '/';

}

bool isExpression(const std::string& input) {

for (char c : input) {

// Check if each character is a digit, operator, or space

if (!(isDigit(c) || isOperator(c) || std::isspace(c))) {

return false;

}

}

return true;

}

int main() {

string input;

cout << "Enter an arithmetic expression: ";

getline(cin, input);

if (isExpression(input)) {

cout << "The string belongs to the grammar.\n";

} else {

cout << "The string does not belong to the grammar.\n";

}

return 0;

}

**OUTPUT:**

A screenshot of a math test

Description automatically generated

A screenshot of a computer

Description automatically generated