COMSATS University Islamabad (CUI)

Lab Terminal

For

CC

By

Iqra Basharat(FA20-BCS-007)

Submitted To:

Sir Bilal

Bachelor of Science in Computer Science (2020-2024)

Question:01

2 functionailties Works.

Answer:

1. loadTransitionTable(string path)

This function reads a transition table from a file specified by the path parameter. The transition table is a matrix of integers that defines the transitions between different states based on the current state and the input character. The matrix is loaded into the rules variable, which is a list of lists (List<List<int>>). Each row in the matrix corresponds to a state, and each column corresponds to a character class (e.g., letters, digits, operators).

- Input:
  - path: The path to the transition table file.

- Functionality:
  - Reads the content of the transition table file.
  - Parses the content to populate the rules matrix.

2. getNextState(int iState, char cChar)

This function determines the next state in the lexical analysis process based on the current state (iState) and the current input character (cChar). The function contains a series of conditional statements that map characters to specific columns in the transition table.

- Inputs:
  - iState: The current state in the state machine.
  - cChar: The current input character.

- Functionality:
  - Checks the type of the input character (letter, digit, operator, etc.).
  - Retrieves the corresponding transition value from the rules matrix.
  - Returns the next state based on the transition value.

3. isKeyword(string sToken)

This function checks if a given token is a keyword. It compares the token to a predefined list of C# keywords. If the token matches any keyword, it returns true; otherwise, it returns false.

- Input:
  - sToken: The token to be checked.

- Functionality:
  - Compares the lowercase version of the token to a list of predefined keywords.
  - Returns true if the token is a keyword; otherwise, returns false.

4. Result(string txt, string tt = @"matrix.txt")

This is the main entry point for the lexical analysis. It takes the input text (txt) and optionally a path to the transition table file (tt). The function iterates through each character in the input text, updating the state and building tokens based on the defined state transitions.

- Inputs:
  - txt: The input text to be analyzed.
  - tt: The path to the transition table file (defaults to "matrix.txt").

- Functionality:
  - Calls loadTransitionTable(tt) to load the transition table.
  - Iterates through each character in the input text.
  - Updates the state machine based on the current character and state.
  - Builds tokens for identifiers, keywords, literals, and operators.
  - Filters out comments.
  - Returns a modified version of the input text with tokens replacing recognized language constructs.

Overall, these functions collectively form a lexical analyzer that tokenizes input code based on the provided rules and transition table, providing a structured representation of the source code. The lexical analysis is a critical step in the compilation process, serving as a foundation for subsequent phases such as parsing and semantic analysis.