Ministerul Educaţiei și Cercetării al Republicii Moldova
Universitatea Tehnică a Moldovei

Facultatea Calculatoare, Informatică și Microelectronică

Course: Formal Languages & Finite Automata

Laboratory work 3: Lexer & Scanner

Elaborated:

st. gr. FAF-221 Clepa Rodion

Verified:

asist. univ. Dumitru Crețu

Chişinău – 2024

Objectives

1. Understand what lexical analysis [1] is.
2. Get familiar with the inner workings of a lexer/scanner/tokenizer.
3. Implement a sample lexer and show how it works.

Implementation

I decided to do sampler lexer for arithmetic equation.

To check the correctness of the sequence of parentheses and arithmetic operations, I created a dictionary that describes which group can move into other groups.

```
transations = {
    "open_parenthesis": ["numbers", "open_parenthesis"],
    "math_operation": ["numbers", "open_parenthesis"],
    "close_parenthesis": ["math_operation", "close_parenthesis"],
    "numbers": ["numbers", "close_parenthesis", "math_operation"],
    "START": ["open_parenthesis", "numbers"]
}
```

Also I listed all elements in specific groups

```
data = {
    "open_parenthesis":["(", "["],
    "close_parenthesis":[")", "]"],
    "math_operation": ["+", "-", "*", "/", "%", "^"],
    "numbers": ["0", "1", "2", "3", "4", "5", "6", "7", "8", "9"]
}
```

```
def check_equation(equation):
      category_mapping = ["START"]
      for symbol in equation:
```

Through function I am transmitting my equation for test, and with **for** I am going through by symbol. category_mapping is used to hold previouses category's names.

```
        for category in data:
            if symbol in data[category]:
                current_category = category
                break
        if current_category in transations[category_mapping[-1]]:
            category_mapping.append(current_category)
```

With a **for,** I am getting current category, and after with an **if** checking if rules are respected.

To handle a correct order of parenthesis, I am using stack. When I am meet open parenthesis **(** I append to stack close parenthesis **),** and after meeting close parenthesis in equation I checking last element if it is the same type and it is not empty I am popping last element, and otherwise return False.

Conclusions

In conclusion, this laboratory work focused on understanding lexical analysis and implementing a sample lexer for arithmetic equations. The objectives included grasping the concept of lexical analysis, becoming familiar with the inner workings of a lexer/scanner/tokenizer, and implementing a functional lexer.