



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных Технологий

Кафедра Вычислительной техники

ОТЧЕТ О ВЫПОЛНЕНИИ ПРАКТИЧЕСКОЙ РАБОТЫ

№3

«Лексический анализатор на базе конечного автомата входного
языка»

по дисциплине

«Теория формальных языков»

Выполнил студент группы ИКБО-04-22

Кликушин В.И.

Принял старший преподаватель

Боронников А.С.

Практическая работа
выполнена

«_ 04 _»__09____2023 г.

«Зачтено»

«__»____2023 г.

Москва 2023

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ	3
2 КОД ПРОГРАММЫ	4
3 НАБОР ТЕСТОВ	7
4 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ.....	8
5 ВЫВОДЫ.....	9
6 ИСПОЛЬЗУЕМЫЕ ИСТОЧНИКИ.....	10

1 ПОСТАНОВКА ЗАДАЧИ

Написать на любом языке программирования (или доработать листинг 1) лексический анализатор на базе конечного автомата входного языка, описанного диаграммой состояний рис. 1.

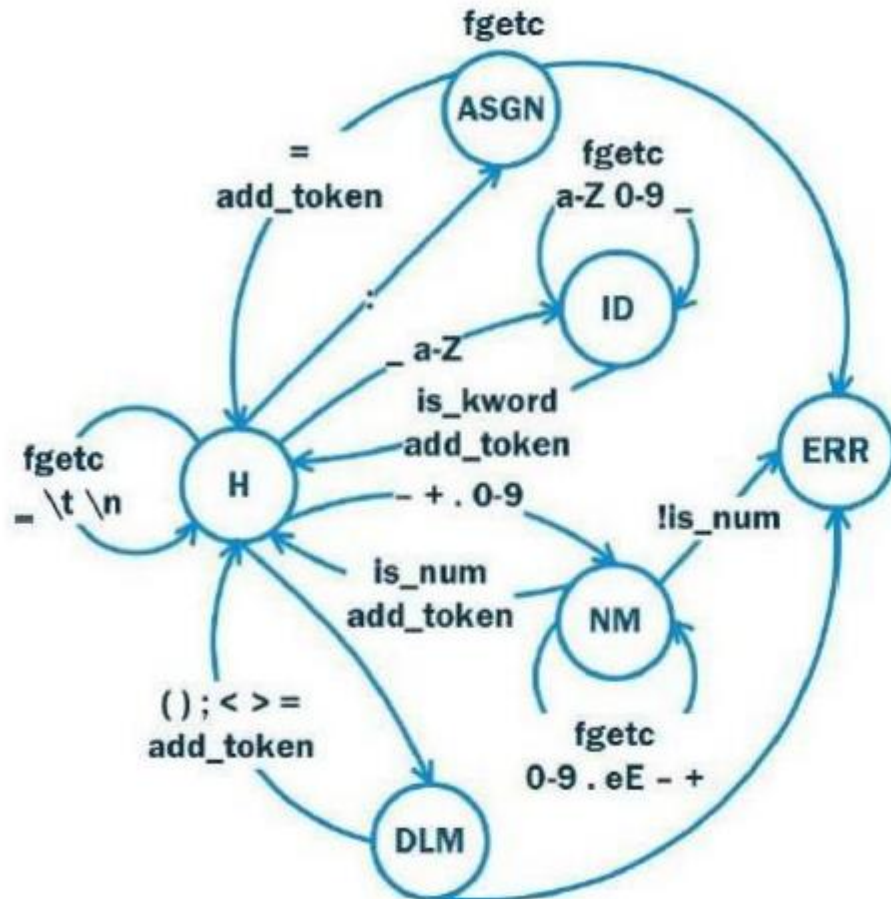


Рис. 1. Диаграмма состояний лексического анализатора

2 КОД ПРОГРАММЫ

В качестве языка программирования был выбран Python.

```
with open('lex_test.txt',encoding='utf-8') as file:
    data = file.read()
    data+='\n'
    status = "H"
    n = 0
    letter = data[n]
    key_words = list()
    idents = list()
    numbers = list()
    operations = list()
    seps = list()
    mistakes = list()
    while (n!=len(data)):
        match status:
            case "H":
                flag_err = "OK"
                while letter in ("\n"," ","\t"):
                    n+=1
                    try:
                        letter = data[n]
                    except:
                        print(f"Ключевые слова: {key_words}")
                        print(f"Идентификаторы: {idents}")
                        print(f"Числа: {numbers}")
                        print(f"Операторы присваивания: {operations}")
                        print(f"Разделители: {seps}")
                        print(f"Исключения: {mistakes}")
                        exit()
                if letter.isalpha() or letter=='_':
                    status = "ID"
                elif letter.isdigit() or letter in '+-.':
                    status = "NM"
                elif letter == ":":
                    status = "ASGN"
                else:
                    status = "DLM"
            case "ID":
                flag_err = "OK"
                id = letter
                n+=1
                letter=data[n]
                while letter.isalpha() or letter.isdigit() or letter == '_':
                    id+=letter
                    n+=1
                    letter = data[n]
                if id in ('for','do'):
```

```

        key_words.append(id)
    else:
        idents.append(id)
    status = "H"
case "NM":
    flag_err = "OK"
    num = letter
    n+=1
    letter = data[n]
    while letter.isdigit() or letter in '._+-eElLfF':
        num+=letter
        n+=1
        letter = data[n]
    c =
num.count('f')+num.count('F')+num.count('l')+num.count('L')
    if c == 1:
        if num[-1] in 'fFlL':
            try:
                float(num[:-1])
            except:
                flag_err = "NM"
                status = "ERR"
        else:
            flag_err = "NM"
            status = "ERR"
    elif c > 2:
        flag_err = "NM"
        status = "ERR"
    elif c==0:
        try:
            float(num)
        except:
            flag_err = "NM"
            status = "ERR"
    if flag_err == "OK":
        numbers.append(num)
        status = "H"
case "ASGN":
    flag_err = "OK"
    op = letter
    n+=1
    letter = data[n]
    op+=letter
    if op == ':=':
        operations.append(op)
        status = "H"
    else:
        flag_err = "ASGN"
        status = "ERR"
n+=1

```

```

        letter=data[n]
    case "DLM":
        flag_err = "OK"
        if letter in ('(',')',';','<','>','='):
            seps.append(letter)
            n+=1
            letter = data[n]
            status = "H"
        else:
            status = "ERR"
    case "ERR":
        if flag_err == "ASGN":
            mistakes.append(op)
        if flag_err == "NM":
            mistakes.append(num)
        if flag_err == "DLM":
            mistakes.append(letter)
        status = "H"

```

3 НАБОР ТЕСТОВ

```
lex_test.txt
1  for (i:=2; i < 10; i := i) do
2  a:=i
3  b := 123821.12
4  c :: 0.12e+6l
5  d:=12e-1f
6  e1137ajahdaj := 12e
7  d := 90
8  f :/ 75.05ee
9  ____:=+12.e+0
10 X:=++777
```

Рисунок 2 – Содержание файла теста №1

```
lex_test.txt X lexical_analyzer.py
lex_test.txt
1  a:=-4646e53
2  b:=.2525
3  c:=-+353
4  d:=1252E5L
5  e:=0.0+e9f
```

Рисунок 3 – Содержание файла теста №2

```
lex_test.txt
1  __:=4
2  __a:=5
3  for(k:=__>6;k:=k=kkk;cdf:=+5.15<-10.084e5) do X:=777777
```

Рисунок 4 – Содержание файла теста №3

4 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

```
Ключевые слова: ['for', 'do']
Идентификаторы: ['i', 'i', 'i', 'i', 'a', 'i', 'b', 'c', 'd', 'e1137ajahdaj', 'd', 'f', '___', 'X']
Числа: ['2', '10', '123821.12', '0.12e+61', '12e-1f', '90', '+12.e+0']
Операторы присваивания: [':=', ':=', ':=', ':=', ':=', ':=', ':=', ':=', ':=']
Разделители: ['(', ';', '<', ';', ')']
Исключения: ['::', '12e', ':/', '75.05ee', '++777']
PS C:\python_projects>
```

Рисунок 5 – Результат теста №1

```
Ключевые слова: []
Идентификаторы: ['a', 'b', 'c', 'd', 'e']
Числа: ['-4646e53', '.2525', '1252E5L']
Операторы присваивания: [':=', ':=', ':=', ':=', ':=']
Разделители: []
Исключения: ['-+353', '0.0+e9f']
PS C:\python_projects>
```

Рисунок 6 – Результат теста №2

```
Ключевые слова: ['for', 'do']
Идентификаторы: ['__', '__a', 'k', '__', 'k', 'k', 'kkk', 'cdf', 'X']
Числа: ['4', '5', '6', '+5.15', '-10.084e5', '777777']
Операторы присваивания: [':=', ':=', ':=', ':=', ':=', ':=']
Разделители: ['(', '>', ';', '=', ';', '<', ')']
Исключения: []
PS C:\python_projects>
```

Рисунок 7 – Результат теста №3

5 ВЫВОДЫ

В ходе выполнения двух практических работ были получены теоретические и практические навыки принципа работы и написания лексического анализатора.

6 ИСПОЛЬЗУЕМЫЕ ИСТОЧНИКИ

1. Теория формальных языков. Лекция 4 – СДО (online-edu.mirea.ru)
2. Алгоритмы и методы – обратная польская запись: [Электронный ресурс]. URL: <https://www.interface.ru/home.asp?artid=1492> (дата обращения 04.09.2023)