



ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
УПРАВЛЕНИЕ ДИСТАНЦИОННОГО ОБУЧЕНИЯ И ПОВЫШЕНИЯ
КВАЛИФИКАЦИИ

Кафедра «Программное обеспечение вычислительной
техники и автоматизированных систем»

Построение простейшего лексиче- ского анализатора с применением flex

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по специальностям

230105-«Программное обеспечение вычислительной
техники и автоматизированных систем»

010503-«Математическое обеспечение и администри-
рование информационных систем»

Автор

Коледов Леонид Викторович

Ростов-на-Дону, 2013



Аннотация

Данная разработка может быть использована в качестве основного учебного материала по дисциплинам: «Теория языков программирования и методы трансляции» и «Теория вычислительных процессов»

Автор

Коледов Леонид Викторович, к. ф.-м. н., доцент,
профессор кафедры

Область научных интересов

Информационные технологии, системы искусственного интеллекта





Оглавление

ЛАБОРАТОРНАЯ РАБОТА 4.....4

[СПЕЦИФИКАЦИЯ](#)



ЛАБОРАТОРНАЯ РАБОТА 4

Задание 1.

Собрать и откомпилировать описанный ниже лексический анализатор. Провести его исполнение в режиме отладки.

Идентификатор может быть представлен следующим образом:

```
letter (letter | digit)*
```

Можно написать программу распознавания символов. Для идентификатора она будет иметь следующий вид:

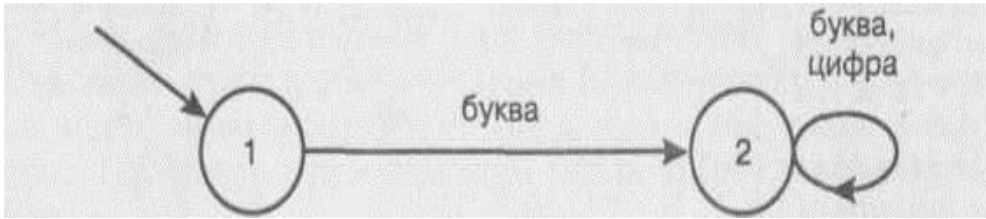
```
#include <stdio.h>      "  
#include <ctype.h>  
main ()  
{char in;  
in = getchar();  
if (isalpha(in))  
in = getchar();  
else error();  
while (isalpha(in) || isdigit(in))  
in = getchar();  
}
```

Здесь in - значение только что считанного знака; функции isalpha() и isdigit() осуществляют проверку аргумента на предмет принадлежности к буквам и цифрам, соответственно;

error () выполняет некоторые операции при возникновении ошибки.

Написать код довольно просто: проверять поступающие символы и использовать цикл while для реализации оператора *.

Конечный автомат распознавателя:



При создании распознавателей удобно использовать представление конечных автоматов. Например, следующая программа осуществляет распознавание идентификаторов.

```

#include <stdio.h>
#include <ctype.h>

int main()
{int state;
char in,
state = 1;
in = getchar();

while (isalpha (in) || isdigit (in) )
{switch (state)

{case 1: if (isalpha(in))
state = 2;
else error();
break;

case 2: state = 2;
break;
} in = getchar();
}

return (state == 2);
}

```

Цикл while обеспечивает прекращение работы программы



Информатика и вычислительная техника

при считывании любого знака, который не является буквой или цифрой. Оператор switch имеет по элементу для каждого состояния автомата, причем в каждом элементе представлены все возможные переходы из данного состояния. Во втором элементе оператора switch уже не нужно проверять вход, поскольку (из-за условия цикла while) переход к нему невозможен, если последний считанный знак не является буквой или цифрой. Присвоение этому состоянию значения 2 не является обязательным - оно просто делает переход явным.

Задание 2.

Решите задачу, номер которой сравним по модулю 8 с двузначным числом, образованному последними двумя цифрами номера Вашей зачетки.

Задача 1.

Токен задается следующим регулярным выражением:

$d \cdot ded^*$

Сконструируйте конечный автомат распознавателя.

Написать программу распознавания токенов.

%%%%%%%%%%
%%%%%%%%%

Задача 2.

Токен задается следующим регулярным выражением:

$(a|b)xx^*(g|h)$

Сконструируйте конечный автомат распознавателя.

Написать программу распознавания токенов.

%%%%%%%%%%
%%%%%%%%%

Задача 3.

Токен задается следующим регулярным выражением:



$(a|c)xx^*(f|h)$

Сконструируйте конечный автомат распознавателя.

Написать программу распознавания токенов.

%%%%%%%%%

Задача 4.

Токен задается следующим регулярным выражением:

$(ab^*)xy^*(g|h^*)$

Сконструируйте конечный автомат распознавателя.

Написать программу распознавания токенов.

%%%%%%%%%