|  |  |
| --- | --- |
| **Изображение выглядит как текст, эмблема, герб, нашивка  Автоматически созданное описание** | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Робототехника и комплексная автоматизация (РК)

КАФЕДРА Системы автоматизированного проектирования (РК-6)

**ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

**по курсу «Программное обеспечение систем автоматизированного проектирования»**

Студент Долженко Анастасия Тимофеевна

Группа РК6-32Б

Тип задания Лабораторная работа №7

Вариант A22(W)

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Долженко А.Т.**

*подпись, дата фамилия, и.о.*

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Князева С.В.**

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Москва, 2023 г**.*

**Задание к лабораторной работе**

**Постановка задачи**

Разработать конечный автомат для распознавания во входном потоке текстовой информации слов из латинских букв, где гласные буквы расположены в алфавитном порядке. Все слова должны передаваться конечному автомату строками потока стандартного ввода. Результаты их распознавания должны отображаться через поток стандартного вывода. Программная реализация конечного автомата должна формироваться генератором LEX по правилам с предусловиями.

**Требования к программе**

При разработке программы необходимо использовать генератор лексических анализаторов (далее по тексту - LEX), который ориентирован на обработку файла спецификаций (далее - Lex-файл). При разработке программы необходимо составить Lex-файл, отражающий специфику лексического анализа целых чисел, и сохранить его под именем lab7.l в выбранном рабочем каталоге. Lex-файл lab7.l программы должен содержать 2 раздела:

• Раздел деклараций

• Раздел правил

которые должны разделять символьные пары %%.

В разделе деклараций должны быть объявлены метки предусловий, для этого в первой строке Lex-файла должна быть использована директива %START, после которой через пробел объявляются метки предусловий: AA EE II OO UU YY CONS ERROR.

В разделе правил Lex-файла программы должны быть введены правила, которые обеспечивают распознавание необходимых лексем с помощью шаблонов регулярных выражений с предусловиями и их функциональную обработку посредством блоков действий. Правила раздела правил Lex-файла программы должны включать шаблоны регулярных выражений, которые обеспечивают распознавание следующих лексем:

* Слово из латинских букв, где все гласные буквы расположены в алфавитном порядке
* Любой набор символов, неудовлетворяющий искомому условию
* Пустая строка

Правила раздела правил Lex-файла программы должны обеспечивать выполнение следующих действий при распознавании лексем:

• Отображение в поток стандартного вывода диагностического сообщения, показывающего верность введённых данных

Правила раздела правил Lex-файла программы должны реализовывать конечный автомат посредством меток предусловий: для этого после обнаружения во входном потоке буквы должна устанавливаться соответствующая метка предусловия.

Метки предусловий устанавливаются оператором BEGIN. Когда оператор вызывается после регулярного выражения, не содержащего меток предусловия, указанная метка добавляется к существующим, а если после регулярного выражения, содержащего метки предусловия, то новая метка снимает раннее установленные.

При обнаружении символа, который недопустим при установленной метке предусловия, устанавливается метка ERROR, после которой игнорируются все символы вплоть до конца строки (обозначаемого символом ’\n’), затем выводится уведомление об ошибке, и процесс лексического анализа начинается сначала.

**Стадии и этапы разработки**

В процессе разработки программы средствами LEX необходимо выполнить следующие этапы:

* получить исходный С-код программы в файле lab7.c, обработав Lex- файл lab7.l командой lex следующим образом:

$ lex -t lab7.l > lab7.c

* построить по исходному коду lab7.c выполняемый файл lab программы, вызвав C-компилятор OS UNIX и подключив стандартную библиотеку объектных модулей LEX libl.a следующим образом:

$ cc -o lab7 lab7.c -ll

**Код программы**

%START AA EE II OO UU YY CONS ERROR

%%

<CONS,AA,EE,II,OO,UU,YY>\n {printf("Correct\n"); BEGIN 0;};

<AA>[aAb-df-hj-np-tv-xzB-DF-HJ-NP-TV-XZ]+ ;

<EE>[eEb-df-hj-np-tv-xzB-DF-HJ-NP-TV-XZ]+ ;

<II>[iIb-df-hj-np-tv-xzB-DF-HJ-NP-TV-XZ]+ ;

<OO>[oOb-df-hj-np-tv-xzB-DF-HJ-NP-TV-XZ]+ ;

<UU>[uUb-df-hj-np-tv-xzB-DF-HJ-NP-TV-XZ]+ ;

<YY>[yYb-df-hj-np-tv-xzB-DF-HJ-NP-TV-XZ]+ ;

<CONS>[b-df-hj-np-tv-xzB-DF-HJ-NP-TV-XZ]+ ;

<CONS>[aA] BEGIN AA;

<CONS,AA>[eE] BEGIN EE;

<CONS,AA,EE>[iI] BEGIN II;

<CONS,AA,EE,II>[oO] BEGIN OO;

<CONS,AA,EE,II,OO>[uU] BEGIN UU;

<CONS,AA,EE,II,OO,UU>[yY] BEGIN YY;

<CONS>\n {printf("Error\n"); BEGIN 0;};

<CONS,AA,EE,II,OO,UU,YY>. BEGIN ERROR;

<ERROR>.\*\n {printf("Error\n"); BEGIN 0;};

[b-df-hj-np-tv-xzB-DF-HJ-NP-TV-XZ] BEGIN CONS;

[aA] BEGIN AA;

[eE] BEGIN EE;

[iI] BEGIN II;

[oO] BEGIN OO;

[uU] BEGIN UU;

[yY] BEGIN YY;

\n {BEGIN 0; printf("Empty line\n");};

. BEGIN ERROR;

%%

int yywrap(){return 1;}

int main(){printf("Enter the line:\n"); yylex();}

**Результаты работы программы**

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

**Список литературы**

1. Кристиан К. Введение в операционную систему UNIX, M.: Финансы и статистика, 1985.
2. Тихомиров В.П., Давидов М.И. Операционная система ДЕМОС: инструментальные средства программирования, М.: Финансы и статистика, 1988.
3. SCO XENIX, Development System, LEX Programmer Guide, SCO Inc., 1986. ( Имеется русский перевод: Генератор лексических анализаторов LEX. Руководство Программиста )
4. Конспекты лекций
5. http://bigor.bmstu.ru