Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н. Э. Баумана (национальный исследовательский университет)

Факультет «Робототехника и комплексная автоматизация»

Кафедра «Системы автоматизированного проектирования»

**Отчет по лабораторной работе № 3**

По курсу «Программное обеспечение систем автоматизированного проектирования»

Выполнил: Студент Гусаров А. А.

Группа РК6-33Б

Проверил:

Дата \_\_\_\_\_\_\_\_\_\_\_\_

Подпись \_\_\_\_\_\_\_\_\_\_\_\_

Москва, 2020 г.

## Введение

Настоящий документ определяет техническое задание на разработку программы синтаксического анализатора (далее по тексту - программа ZEROONE) для распознавания записей любых двоичных наборов, где после каждого нуля стоит, по крайней мере, одна единица.

## Основания для разработки

Программа ZEROONE разрабатывается в рамках лабораторной работы по курсу "*Лингвистическое обеспечение САПР*" для практического изучения этапа синтаксического анализа в процедурах трансляции формальных языков.

## Назначение разработки

Программа ZEROONE предназначается для реализации грамматического разбора входного потока цифровых данных.

## Требования к программе

1. ***Требования к функциональным характеристикам***
   1. Программа ZEROONE должна обеспечивать синтаксический анализ двоичных наборов в произвольном бинарном выражении, которое передается символьной строкой из потока стандартного ввода.
   2. Программа ZEROONE должна предусматривать возможность обработки 2-х типов символов: 0 и 1
   3. Результат распознавания программой ZEROONE корректного двоичного набора должен отображаться в диагностические сообщения потока стандартного вывода.

## Требования к надежности

* 1. Программа ZEROONE не должна иметь каких-либо ограничений по числу цифр в анализируемом выражении, кроме внутренних ограничений инструментальных средств, использованных для ее реализации.

1. ***Требования к составу и параметрам технических средств***

Программа ZEROONE должна быть разработана исходя из возможности реализации на стандартном составе технических средств компьютеров любой архитектуры, после соответствующей трансляции исходного кода.

## Условия эксплуатации

* 1. Программа ZEROONE должна быть ориентирована на эксплуатацию в операционной среде OS UNIX любой версии.
  2. Программа ZEROONE должна быть реализована в виде выполняемого файла с именем zeroone, по которому она должна вызываться средствами любого командного процессора OS UNIX.
  3. Программа ZEROONE должна эксплуатироваться в интерактивном режиме, читая строку входного выражения из потока стандартного ввода и отображая результат ее обработки в потоке стандартного вывода.

## Требования к информационной и программной совместимости

* 1. Чтобы обеспечить выполнение требуемых технических характеристик, программа ZEROONE должна реализовывать синтаксический анализ любой входной строки бинарного выражения из потока стандартного ввода.
  2. Синтаксический анализатор программы ZEROONE должен обеспечивать грамматический разбор двоичных наборов с целью установить соответствие или несоответствие содержащей их строки потока стандартного ввода требуемому формату последовательности.
  3. Грамматику двоичных наборов синтаксического анализатора программы ZEROONE должны составлять следующие элементы:
* терминальные символы (терминалы), соответствующие структурным единицам (лексемам) входного двоичного выражения;
* начальный нетерминальный символ (начальный нетерминал), к которому приводится входное бинарное выражение;
* нетерминальные символы (нетерминалы), обозначающие допустимые варианты комбинации лексем во входном бинарном выражении;
* система продукций (правил вывода), обеспечивающая грамматический разбор входного двоичного выражения.
  1. Начальный нетерминал грамматики двоичных наборов синтаксического анализатора программы ZEROONE должен обозначаться именем input. Он должен выводиться из любой корректной последовательности.
  2. Система продукций (правил вывода) синтаксического анализатора программы ZEROONE должна обеспечивать грамматический разбор произвольной входной строки потока стандартного ввода путем приведения терминалов и нетерминалов к начальному нетерминалу методом LR(1) анализа снизу-вверх по дереву вывода. Приведение входной строки к начальному нетерминалу по правилам вывода грамматики двоичных наборов должно означать успех грамматического разбора. Альтернативный результат должен рассматриваться как синтаксическая ошибка в введенной строке.
  3. Для разработки синтаксического анализатора программы ZEROONE, необходимо использовать генератор синтаксических анализаторов (далее по тексту - YACC) из состава OS UNIX, инструментальные средства которого ориентированы на обработку файла спецификаций (далее по тексту, Yacc-файл) проектируемого синтаксического анализатора.
  4. При разработке синтаксического анализатора программы ZEROONE необходимо составить Yacc-файл, отражающий специфику грамматического разбора бинарного выражения, и сохранить его под именем zeroone.y в выбранном доступном рабочем каталоге файловой системы OS UNIX.
  5. Проектируемый Yacc-файл zeroone.y должен состоять из 3-х секций: деклараций, правил и функций. Разделителем секций должны быть символические пары %%, расположенные в начальных позициях содержащих их строк Yacc-файла. Каждая секция Yacc-файла zeroone.y должна содержать соответствующее число комментариев в формате, принятом для системы программирования C.
  6. Секция деклараций Yacc-файла zeroone.y должна включать:
* cпецификацию блока внешних описаний, ограниченную директивами %{ и %}, в котором необходимо декларировать и определить внешние целочисленные (типа int) переменные lengZero и maxLengZero для фиксации длины последовательности из нулей.
  1. В секции правил Yacc-файла zeroone.y должны быть приведены описания продукций приведения нетерминалов.
  2. Каждая продукция секции правил Yacc-файла zeroone.y должна быть задана в нотации, близкой к форме Бэкуса-Наура, где в левой части указывается приводимый нетерминал, а в правой - последовательность терминалов и/или нетерминалов, которые перечисляются через пробел. Для разделения частей продукции должен использоваться символ двоеточия (:). Каждую продукцию нужно начинать с новой строки и завершать либо символом точки с запятой (;), либо блоком действий в фигурных скобках.

## Стадии и этапы разработки

Процесс разработки про граммы ZEROONE целесообразно разделяться на следующие 3 стадии:

* составить Yacc-файл zeroone.y в выбранном рабочем каталоге файловой системы OS UNIX, используя любой текстовый редактор, например, xedit, ориентированный на работу в операционной среде X Window System, или joe, для редактирования в консольном режиме;
* получить исходный код синтаксического анализатора в файле y.tab.c текущего каталога файловой системы OS UNIX, обработав Yacc-файл zeroone.y командой yacc, следующим образом:

# $ yacc zeroone.y

* сформировать выполняемый модуль в файле zeroone текущего каталога файловой системы OS UNIX, компилируя исходный код синтаксического анализатора в файле y.tab.c следующей командой:

# $ cc -o zeroone y.tab.c

# Результаты разработки программы ZEROONE необходимо оформить в виде проектного документа, содержащего описание двоичных наборов и файла спецификаций для генератора синтаксических анализаторов YACC.

## Код программы

%{

#include <stdio.h>

void yyerror(char const \*);

int yylex(void);

int lengZero;

int maxLengZero;

%}

%token ZERO

%token ONE

%%

input: { printf("Enter the line: \n"); }

| input line

;

line: '\n' { printf("Empty line!\n"); }

| error '\n' { yyerrok; }

| expr '\n'

{

if ((maxLengZero > 1) || ($1 == 0))

yyerror("syntax error");

}

;

expr: ZERO { lengZero = 1; maxLengZero = 1; $$ = 0; }

| ONE { lengZero = 0; maxLengZero = 0; $$ = 1; }

| expr ZERO

{

++lengZero;

if (lengZero > maxLengZero)

maxLengZero = lengZero;

$$ = 0;

}

| expr ONE

{

lengZero = 0;

$$ = 1;

}

;

%%

int yylex(void)

{

int c;

c = getchar();

if(c == '0')

return ZERO;

if(c == '1')

return ONE;

if (c == EOF)

return 0;

return c;

}

void yyerror(char const \*s)

{

fprintf(stderr, "%s\n", s);

}

int main()

{

yyparse();

return 0;

}

## Порядок контроля и приемки

1. Для проверки функционирования программы ZEROONE должны быть предложены контрольные примеры, предусматривающие стандартный ввод корректных и некорректных двоичных наборов.
2. Для приемки программы ZEROONE должен быть организован вызов выполняемого файла zeroone в консольном режиме работы OS UNIX или режиме эмуляции терминала операционной среды X Window System.

## Приложение

При разработке синтаксических анализаторов в OS UNIX для проектирования программы ZEROONE рекомендуется использовать литературные источники, перечисленные ниже.

1. Рейуорд-Смит В.Дж. Теория формальных языков. Вводный курс, M.: Радио и связь, 1988.
2. Тихомиров В.П., Давидов М.И. Операционная система ДЕМОС: инструментальные средства программирования, М.: Финансы и статистика, 1988.
3. SCO XENIX, Development System, YACC Programmer Guide, SCO Inc., 1986. ( Имеется русский перевод: Генератор синтаксических анализаторов YACC. Руководство Программиста )