

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 № 5

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 ZEROorONE) для распознавания записей двоичных наборов, где число единиц нечетно, а число нулей делится на 3.

Основания для разработки

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

Назначение разработки

Программа ZEROorONE предназначена для реализации грамматического разбора входного потока цифровых данных.

Требования к программе

1. Требования к функциональным характеристикам

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

2. Требования к надежности

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

3. Требования к составу и параметрам технических средств

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

4. Условия эксплуатации

- 4.1. Программа ZEROorONE должна быть ориентирована на эксплуатацию в операционной среде OS UNIX любой версии.
- 4.2. Программа ZEROorONE должна быть реализована в виде выполняемого файла с именем zeroorone, по которому она должна вызываться средствами любого командного процессора OS UNIX.
- 4.3. Программа ZEROorONE должна эксплуатироваться в интерактивном режиме, читая строку входного выражения из потока стандартного ввода и отображая результат ее обработки в потоке стандартного вывода.

5. Требования к информационной и программной совместимости

- 5.1. Чтобы обеспечить выполнение требуемых технических характеристик, программа ZEROorONE должна реализовывать синтаксический анализ любой входной строки бинарного выражения из потока стандартного ввода.
- 5.2. Синтаксический анализатор программы ZEROorONE должен обеспечивать грамматический разбор двоичных наборов с целью установить соответствие или несоответствие содержащей их строки потока стандартного ввода требуемому формату последовательности.
- 5.3. Грамматику двоичных наборов синтаксического анализатора программы ZEROorONE 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.
- 5.4. Начальный нетерминал грамматики двоичных наборов синтаксического анализатора программы ZEROorONE должен обозначаться именем input. Он должен выводиться из любой корректной последовательности.
- 5.5. Система продукций (правил вывода) синтаксического анализатора программы ZEROorONE должна обеспечивать грамматический разбор произвольной входной строки потока стандартного ввода по правилам регулярной грамматики. 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.
- 5.6. Для разработки синтаксического анализатора программы ZEROorONE, 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.
- 5.7. При разработке синтаксического анализатора программы ZEROorONE необходимо составить Yacc-файл, отражающий специфику грамматического

разбора бинарного выражения, и сохранить его под именем zeroorone.y в выбранном доступном рабочем каталоге файловой системы OS UNIX.

5.8. Проектируемый Yacc-файл zeroorone.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 zeroorone.y должна содержать соответствующее число комментариев в формате, принятом для системы программирования C.

5.9. Секция деклараций Yacc-файла zeroorone.y должна включать:

- спецификацию блока внешних описаний, ограниченную директивами `%{` и `%}`, в котором необходимо подключить библиотеку `stdio.h` и предопределить функции `yylex` и `yerror`.

5.10. В секции правил Yacc-файла zeroorone.y должны быть приведены описания продукций приведения нетерминалов.

5.11. Каждая продукция секции правил Yacc-файла zeroorone.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 программы ZEROorONE целесообразно разделяться на следующие 3 стадии:

- составить Yacc-файл zeroorone.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 zeroorone.y командой `yacc`, следующим образом:

\$ yacc zeroorone.y

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

\$ cc -o zeroorone y.tab.c

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

Код программы

Для левостороннего вывода:

```
%{
    #include<stdio.h>
    int yylex();
    int yyerror(char* s);
}%

%%

S: A '\n' { return 0; };

A:   '1'
     |A2 '0'
     |B '1'
     ;

B:   B2 '0'
     | A '1'
     ;

A2: B2 '1'
     | A1 '0'
     ;

A1: B1 '1'
     | A '0'
     ;

B2: B1 '0'
     | A2 '1'
     ;

B1: '0'
     |B '0'
     | A1 '1'
     ;

%%

int yylex()
{
    return (getchar());
}

int yyerror(char *s)
{
    return 1;
}
```

```

int main(int argc, char *argv[])
{
    int ret;

    if((ret = yyparse() ) == 0)
        puts("Correct");
    else
        puts("Error");
    return (ret);
}

```

Для правостороннего вывода:

```

%{
    #include <stdio.h>
    int yylex();
    int yyerror(char* s);
}%

%%
A: '0' E
  |'1' B
  ;
B: '0' F
  |'1' A
  |'\n' { return 0; }
  ;
C: '0' A
  |'1' D
  ;
D: '0' B
  |'1' C
  ;
E: '0' C
  |'1' F
  ;
F: '0' D
  |'1' E
  ;
%%

int yylex()
{
    return (getchar());
}

int yyerror(char* s)
{
    return 1;
}

int main(int argc, char* argv[])
{

```

```

int ret;

if((ret = yyparse() ) == 0)
    puts("Correct");
else
    puts("Error");
return (ret);
}

```

Порядок контроля и приемки

1. Для проверки функционирования программы ZEROorONE должны быть предложены контрольные примеры, предусматривающие стандартный ввод корректных и некорректных двоичных наборов.
2. Для приемки программы ZEROorONE должен быть организован вызов выполняемого файла zeroorone 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 ZEROorONE рекомендуется использовать литературные источники, перечисленные ниже.

1. Рейуорд-Смит В.Дж. Теория формальных языков. Вводный курс, М.: 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)