|  |  |
| --- | --- |
| **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 №5

Вариант T23

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

*подпись, дата фамилия, и.о.*

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Волосатова Т.М.**

*подпись, дата фамилия, и.о.*

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

*Москва, 2023 г**.*

**Задание к лабораторной работе**

**Постановка задачи**

Разработать транслятор для преобразования записей линейных алгебраических неравенств (ЛАН) с целочисленными коэффициентами в эквивалентный формат линейных алгебраических уравнений (ЛАУ). При этом все переменные ЛАН и ЛАУ должны быть заданы одинаковыми латинскими буквами с различными числовыми индексами. Входные записи ЛАН должны передаваться транслятору строками стандартного ввода. Результаты их трансляции в ЛАУ должны отображать строки стандартного вывода.

**Требования к программе**

1. Требования к функциональным характеристикам

1.1. Программа должна в интерактивном режиме распознавать и преобразовывать записи ЛАН в эквивалентный формат ЛАУ.

1.2. ЛАН должны передаваться строками стандартного ввода без разделителей и иных символов, кроме знаков ‘-‘, ’+’, ‘(’, ’)’, ‘>’, ‘<’, ‘>=’, ‘<=’, ‘!=’ описания переменных и десятичных цифр.

1.3. Программа обнаруживает соответствие или несоответствие символьной

2. Условия эксплуатации

2.1. Программа должна быть ориентирована на эксплуатацию в среде OS UNIX.

2.2. Программа должна эксплуатироваться в интерактивном режиме, читая строки из потока стандартного ввода и отображая результаты их обработки в потоке стандартного вывода.

3. Требования к информационной и программной совместимости

3.1. Чтобы обеспечить выполнение требуемых технических характеристик, программа должна реализовывать синтаксический и лексический анализ любой входной строки, содержащей символьные наборы, из потока стандартного ввода.

3.2. Лексический анализатор программы должен обеспечивать распознавание лексем, соответствующих формату ЛАН.

3.3. Лексический анализатор программы должен обеспечивать выделение следующих типов лексем:

* DIGIT [1-9][0-9]\* - натуральные числа
* SIGN (>|<|>=|<=|!=) – знаки неравенства
* VAR [A-Za-z] – переменная без индекса
* LBRACE [(] – круглая скобка
* RBRACE [)] – круглая скобка
* PN\_TOKEN [+-] – знаки + и -

Макроопределения значений типов лексем следует сосредоточить в заголовочном файле y.tab.h, который должен формироваться при разработке синтаксичеcкого анализатора программы ПУ.

3.4. Для сохранения значений лексем следует использовать внешнюю (extern) целочисленную (типа int) переменную yylval.

3.5. Для разработки лексического анализатора программы необходимо использовать генератор лексических анализаторов (далее по тексту - LEX) OS UNIX, инструментальные средства которого должны быть ориентированы на обработку файла спецификаций проектируемого лексического анализатора (далее по тексту - Lex-файл).

3.6. Проектируемый Lex-файл должен состоять из 2-х разделов: раздел деклараций и раздел правил. Разделы Lex-файла должны отделяться символической парой %%.

3.7. В разделе деклараций Lex-файла необходимо специфицировать блок внешних описаний, ограничив его директивами %{ и %}, в котором нужно объявить внешнюю целочисленную переменную yylval для хранения значений лексем, включить заголовочные файл y.tab.h для макроопределений терминалов и stdlib.h для функции преобразования строки в число (atoi) с помощью директивы "#include" препроцессора системы программирования C.

3.8. В разделе правил Lex-файла должны быть введены правила, которые обеспечивают распознавание лексем, перечисленных в п. 4.3, с помощью шаблонов регулярных выражений и необходимую функциональную обработку переменных в блоках действий правил.

3.9. Блоки действий правил Lex-файла LDE.l должны обеспечивать: сохранение значений коэффициентов, изменение индексов переменных и знака выражения, возврат типа распознанной лексемы оператором return, аргумент которого должен соответствовать макроопределению лексемы из заголовочного файла y.tab.h;

3.10. Вызов лексического анализатора должен осуществляться путем обращения к функции yylex() в синтаксическом анализаторе программы. Связь между синтаксическим и лексическим анализаторами должна осуществляться по типу лексемы, определяемому кодом возврата функции yylex(), и по значению лексемы, который передается через внешнюю переменную yylval.

3.11. Синтаксический анализатор программы должен обеспечивать трансляцию потока лексем от лексического анализатора в необходимый формат.

3.12. Для выполнения трансляции разбора синтаксический анализатор программы должен реализовывать однозначную контекстно-свободную грамматику, которая ориентирована на построчную обработку ЛАН.

3.13. Грамматику ЛАН синтаксического анализатора программы должны составлять следующие элементы: начальный нетерминал (input), нетерминал (exp), нетерминал (line), терминалы (лексемы, выделенные лексическим анализатором), система продукций (правил вывода), которые должны обеспечивать вывод входной строки стандартного ввода путем приведения терминалов и нетерминалов к аксиоме грамматики.

**Стадии и этапы разработки**

При разработке лексического анализатора программы lab5 необходимо выполнить следующие этапы:

* получить исходный код синтаксического анализатора в файле y.tab.c и заголовочный файл макроопределений типов лексем для лексического анализатора в файле y.tab.h, обработав Yacc-файл lab5\_6.y командой yacc, следующим образом: $ yacc -d lab5\_6.y
* получить исходный код лексического анализатора в файле lex.yy.c, обработав Lex-файл lab5\_6.l командой lex следующим образом: $ lex lab5\_6.l
* Для сборки выполняемого файла lab5 из объектных модулей lex.yy.o, y.tab.o, lab5.o, полученных на предыдущих стадиях, необходимо применить редактор связей OS UNIX, реализовав его вызов следующей командой:  
  $ cc y.tab.с lex.yy.с

**Результаты тестирования**

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**Содержимое файла спецификации lex**

%{

#include <stdio.h>

#include <stdlib.h>

#include "y.tab.h"

%}

DIGIT [1-9][0-9]\*

SIGN (>|<|>=|<=|!=)

VAR [A-Za-z]

LBRACE [(]

RBRACE [)]

PN\_TOKEN [+-]

%%

{DIGIT} {yylval.digit = atoi(yytext); return DIGIT;}

{VAR} {yylval.var\_ = \*yytext; return VAR;}

{SIGN} {return SIGN;}

{PN\_TOKEN} {yylval.pn\_ = \*yytext; return PN\_TOKEN;}

{LBRACE} {yylval.lbrace = \*yytext; return LBRACE;}

{RBRACE} {yylval.rbrace = \*yytext; return RBRACE;}

\n {return '\n';};

. {};

%%

int yywrap() {return 1;}

**Источники**

1. Родионов C.В., Волосатова Т.М. “Автоматизация проектирования лексических анализаторов”
2. Тихомиров В.П., Давидов М.И. Операционная система ДЕМОС: инструментальные средства программирования, М.: Финансы и статистика, 1988
3. База и Генератор Образовательных Ресурсов (bigor.bmstu.ru)
4. Генератор программ синтаксического анализа yacc. Производственно- внедренческий кооператив "И Н Т Е Р Ф Е Й С": Демос/P 2.1, 1988