|  |  |
| --- | --- |
|  | **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 дисциплине: «Программное обеспечение систем автоматизированного проектирования»

Студент Камалов Антон Павлович

Группа РК6-35Б

Тип задания Лабораторная работа №5

Вариант T23 (LEX)

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Камалов А.П.**

*подпись, дата фамилия, и.о.*

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Родионов С.В.**

*подпись, дата фамилия, и.о.*

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

*Москва, 2023 г.*

**Оглавление**

[Задание 3](#_Toc1)

[Введение 3](#_Toc2)

[Основания для разработки 3](#_Toc3)

[Назначение разработки 3](#_Toc4)

[Требования к программе 4](#_Toc5)

[1. Требования к функциональным характеристикам 4](#_Toc6)

[2. Требования к надежности 4](#_Toc7)

[3. Требования к составу и параметрам технических средств 4](#_Toc8)

[4. Условия эксплуатации 4](#_Toc9)

[5. Требования к информационной и программной совместимости 4](#_Toc10)

[Стадии и этапы разработки 6](#_Toc11)

[Порядок контроля и приема 7](#_Toc12)

[Список литературы 7](#_Toc13)

[Приложение №1. Результаты разработки 7](#_Toc14)

[Приложение №2. Содержимое файла спецификации 9](#_Toc15)

# Задание

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.

# Введение

Настоящий документ определяет техническое задание на разработку лексического анализатора транслятора (далее по тексту - программа T23\_L) для распознания заданных двоичных наборов. Программа T23\_L разрабатывается в рамках лабораторной работы по курсу «Программное обеспечение систем автоматизированного проектирования» для практического изучения этапов лексического и синтаксического анализа в процедурах трансляции формальных языков.

# Основания для разработки

Программа T23\_L разрабатывается в рамках лабораторной работы по курсу «Программное обеспечение систем автоматизированного проектирования» для практического изучения этапов лексического и синтаксического анализа в процедурах трансляции формальных языков.

# Назначение разработки

Программа T23\_L предназначается для для преобразования записей ЛАН с целочисленными коэффициентами в эквивалентный формат ЛАУ.

# Требования к программе

## Требования к функциональным характеристикам

Программа T23\_L должна в интерактивном режиме распознавать элементы линейных алгебраических неравенств.

## Требования к надежности

Программа T23\_L не должна иметь каких-либо ограничений по числу символов в анализируемой строке, кроме внутренних ограничений инструментальных средств, использованных для её реализации.

## Требования к составу и параметрам технических средств

Программа T23\_L должна быть разработана исходя из возможности реализации на стандартном составе технических средств компьютеров любой архитектуры после соответствующей трансляции исходного кода.

## Условия эксплуатации

4.1. Программа T23\_L должна быть ориентирована на эксплуатацию в среде OS UNIX.

4.2. Программа T23\_L должна быть реализована в виде выполняемого файла с именем T23\_L, по которому она должна вызываться средствами любого командного процессора OS UNIX.

4.3. Программа T23\_L должна эксплуатироваться в интерактивном режиме, читая строки из потока стандартного ввода и отображая результаты их обработки в потоке стандартного вывода.

## Требования к информационной и программной совместимости

5.1. Чтобы обеспечить выполнение требуемых технических характеристик, программа T23 должна реализовывать синтаксический и лексический анализ любой входной строки, содержащей символьные наборы, из потока стандартного ввода.

5.2. Лексический анализатор программы T23 должен обеспечивать распознавание лексем, соответствующих формату ЛАН.

5.3. Лексический анализатор программы T23 должен обеспечивать выделение следующих типов лексем:

• X - переменная с индексом

• GLT – знаки ‘<’, ‘<=’, ‘>’, ‘>=’

• SIGN – знак переменной

• COEF – свободный коэффициент

Макроопределения значений типов лексем следует сосредоточить в заголовочном файле y.tab.h, который должен формироваться при разработке синтаксичеcкого анализатора программы T23.

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

5.5. Для разработки лексического анализатора программы T23, 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).

5.6. При разработке лексического анализатора программы T23 необходимо составить Lex-файл, отражающий специфику лексического анализа ЛАУ, и сохранить его под именем t23.l в выбранном рабочем каталоге файловой системы OS UNIX.

5.7. Проектируемый Lex-файл t23.l должен состоять из 2-х разделов: раздел деклараций и раздел правил. Разделы Lex-файла должны отделяться символической парой %%.

5.8. В разделе деклараций Lex-файла t23.l 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.

5.9. В разделе правил Lex-файла t23.l должны быть введены правила, которые обеспечивают распознавание лексем, перечисленных в п. 5.3, с помощью шаблонов регулярных выражений и необходимую функциональную обработку переменных в блоках действий правил.

5.10. Блоки действий правил Lex-файла t23.l должны обеспечивать:

• сохранение значений коэффициентов, переменных и их индексов во внешней переменной yylval, которые возвращаются функцией atoi, преобразовывающей yytext в число;

• возврат типа распознанной лексемы оператором return, аргумент которого должен соответствовать макроопределению лексемы из заголовочного файла y.tab.h;

Каждый блок действий должен быть ограничен парой фигурных скобок, внутри которых допустимо использовать любые конструкции и вызовы библиотечных функций системы программирования C. Например, блок действий для коэффициента рекомендуется оформить следующим образом:

([1-9][0-9]\*)|(0) { yylval.ival = atoi(yytext); return COEF; }

5.11. Вызов лексического анализатора должен осуществляться путем обращения к функции yylex() в синтаксическом анализаторе программы T23. 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

# Стадии и этапы разработки

Процесс разработки программы T23 целесообразно разделить на следующие 3 стадии:

* составить lex-файл в выбранном рабочем каталоге файловой системы OS

UNIX, используя любой текстовый редактор;

* получить исходный код лексического анализатора в файле lex.yy.c текущего каталога файловой системы OS UNIX, обработав lex-файл командой lex следующим образом:

**$ lex t23.l**

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

**$ cc lex.yy.c -o T23\_L**

Результаты разработки программы T23\_L должны содержать описание грамматики и файл спецификаций для генератора синтаксических анализаторов YACC.

# Порядок контроля и приема

1. Для проверки функционирования программы T23\_L должны быть предложены контрольные примеры, предусматривающие стандартный ввод корректных и некорректных неравенств.
2. Для приёмки программы T23\_L должен быть организован вызов выполняемого файла T23\_L в консольном режиме работы OS UNIX.

# Список литературы

1. Родионов C.В., Волосатова Т.М. Автоматизация проектирования лексических анализаторов. Москва: Издательство МГТУ им. Н. Э. Баумана, 2005.

2. Волосатова Т.М., Родионов С.В. Лингвистическое обеспечение САПР. Режим доступа: http://bigor.bmstu.ru/?cnt/?doc=LO-SAPR/base.cou

## Приложение №1. Результаты разработки

**Описание структуры программы**

Секция объявлений:

* stdio.h - для работы с функциями стандартного ввода/вывода;
* free\_coef – для хранения свободного коэффициента;
* var\_index, char\_var, – массивы, хранящие индексы и символы переменных, а также коэффициенты при переменных;
* len\_var\_index, len\_char\_var, len\_a – длины массивов var\_index, char\_var и a соответственно;
* token X SIGN GLT COEF - терминалы, обозначающие переменные, знаки ‘+’ или ‘-‘, знаки ‘>’, ‘<’, ‘>=’, ‘<=’ и свободные коэффициенты.

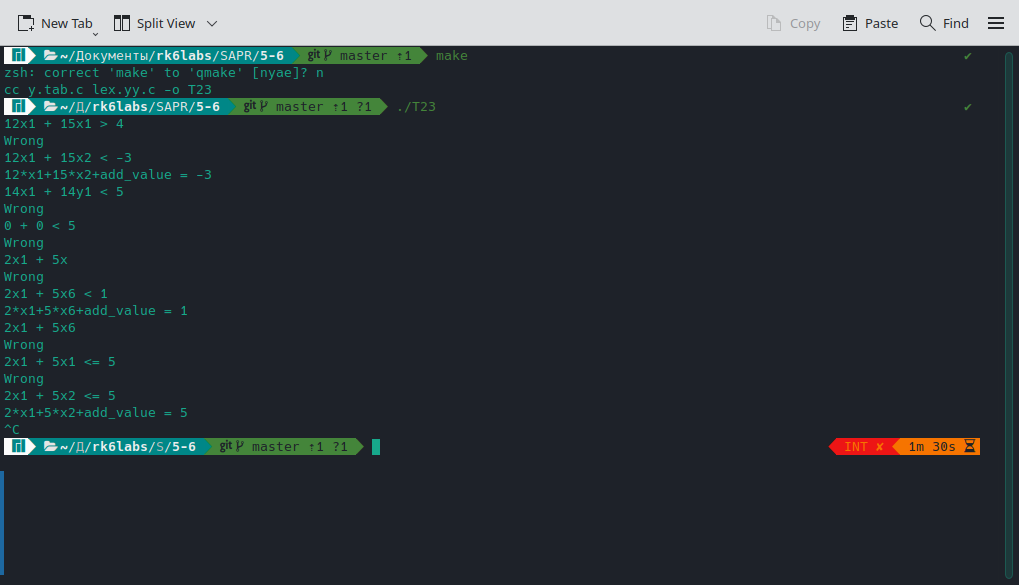
Секция правил:

* input - описывает входную строку вместе с приглашением ко вводу;
* line - содержит различные варианты входной строки и действия при этих вариантах. Для строки, не соответствующей правилу, будет сброшен флаг ошибки, чтобы программа предложила повторный ввод. Для верной строки будет выведено соответствующее сообщение;
* ineq – описывает корректный вид неравенства;
* exp - правило, рекурсивно описывающее требуемую левую часть ЛАН.

Секция программ:

* int yylex() - функция, реализующая лексический анализатор. Возвращает номер нетерминала, если он является нужной буквой, или символ в другом случае;
* int yyerror() - функция, вызываемая при неправильном вводе. Выводит на экран сообщение об ошибке;
* int main() - основная функция программы. Вызывает функцию yyparse(), которая реализует синтаксический анализатор;
* int check\_letters() – функция, проверяющая нахождение в неравенстве переменной с символом, отличающимся от остальных;
* int check\_indexes() – функция, проверяющая нахождение в неравенстве переменной с символом, индекс которого повторяется;
* void out() – функция, выводящая левую часть ЛАУ;

**Результаты лексического и синтаксического анализа**



## Приложение №2. Содержимое файла спецификации

%{

#include <stdio.h>

#include "y.tab.h"

%}

%%

\n return '\n';

[ \t] ;

[A-Za-z]([1-9][0-9]\*)|(0) { yylval.var.letter = yytext[0]; yylval.var.index = atoi(yytext+1); return X; }

(<=?)|(>=?) { return GLT; }

[\+\-] { yylval.cval = yytext[0]; return SIGN; }

([1-9][0-9]\*)|(0) { yylval.ival = atoi(yytext); return COEF; }

. ;