Министерство науки и высшего образования Российской Федерации

Новосибирский государственный технический университет

Кафедра вычислительной техники

Отчет по лабораторной работе № 3

по дисциплине «Теория формальных языков и компиляторов»

на тему «Синтаксис языков программирования. Формальные грамматики»

Студент: Шалаев И. А.

Вариант: 41111111

Группа: АВТ-709

Преподаватель: Малявко А.А.

Новосибирск, 2020

# Цель работы

Изучение основных понятий метаязыка формальных грамматик, свойств грамматик и нетерминальных символов, рекурсивности и однозначности грамматик, недостижимости, бесплодности, аннулируемости и рекурсивности нетерминальных символов, отношений предшествования и последования между символами, приобретение навыков эквивалентных преобразований формальных грамматик, освоение технологий разработки формальных грамматик.

# Постановка задачи

1. Изучить интерфейс пакета ВебТрансЛаб, предназначенный для работы с синтаксическими правилами.

2. На примерах 3LL1 и 4SLR1GrammarArithmExpr (грамматики Ga1 и Ga2 из учебника) изучить и освоить:

* ввод и редактирование правил грамматики;
* построение синтаксических акцепторов с целью вычисления свойств грамматики и ее символов – пункт меню «Построить» с выбором нужного шаблона;
* понятия множеств предшественников и множеств последователей для нетерминальных символов;
* понятие множеств выбора правил грамматики и их пересечений для правил с одинаковыми нетерминалами в левой части;
* просмотр свойств символов грамматики (следует достичь понимания того, почему те или иные символы грамматики имеют свой конкретный набор свойств – пункт меню «Показать» и подпункты «Правила грамматики», «Отношение предшествования», «Отношение последования» и «Множества выбора правил».

3. Изучить способы эквивалентных преобразований грамматики из примера 4SLR1GrammarArithmExpr для устранения пересечений множеств выбора правил с одинаковыми нетерминалами в левой части, т.е. превращения этой грамматики в LL(1)-грамматику.

4. Ориентируясь на свой вариант задания на курсовую работу, модифицировать (расширить) грамматики Ga1 и Ga2 до грамматик операторов присваивания, условного оператора и оператора цикла, добавляя новые нетерминалы и правила по технологии разработки «снизу вверх». Изучить свойства полученных грамматик и их символов.

5. Используя полученные навыки работы с грамматиками и программным обеспечением, начать поэтапную разработку грамматики по технологии «сверху вниз» для заданного варианта курсовой работы.

6. Оценить свои субъективные проблемы и трудности разработки формальных грамматик при использовании разных технологий («снизу вверх» и «сверху вниз»). Выбрать технологию для выполнения задания на курсовую работу (рекомендуется технология «сверху вниз»).

7. Подготовить, сдать и защитить отчет к лабораторной работе.

Вариант:

Идентификаторы: *$<пЦ><пБ> ($1ad, $74c, $0B…)*

Константы: *целые по основаниям 4, 10 и 16; вещественные; строковые*

Типы: *long[\_u] number char*

Оператор присваивания: *<И> = <В>;*

Условный оператор: *at <ЛВ> do <ОБ> [ else do <ОБ> ]*

Оператор цикла: *loop <ОБ> until <ЛВ>*

Оператор переключателя:

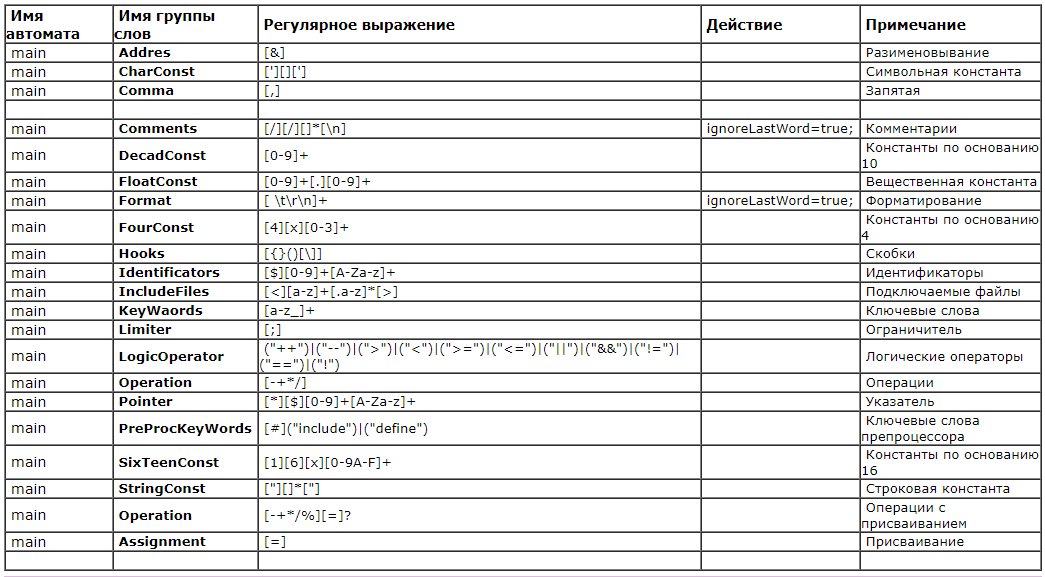
*select <В> case (<К> ) <ОБ> [break;] …[ case () <ОБ> ] end*

Формат псевдокода: *триады <Код><Оп><Оп>*

# Ход работы

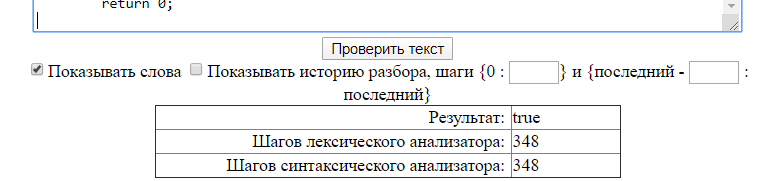
**Доработки:**

Были доработаны правила лексики. Обновлённые правила показаны на Рис. 1.



*Рис. 1 - Обновленные правила лексики*

Для тестирования новых правил использовался тот же код программы. Результат показан на Рис 2.

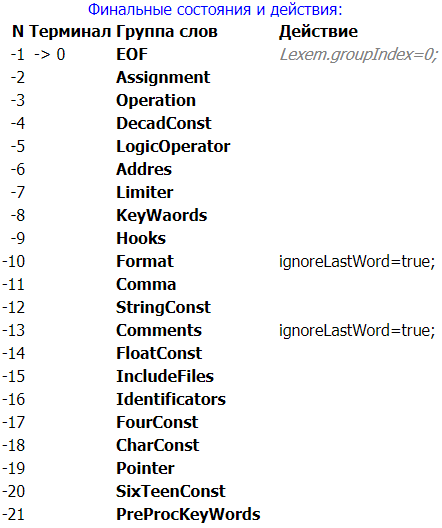


*Рис. 2 - Проверка обновленных правил лексики*

**Работа лексических автоматов по разбору заданного фрагмента:**

Управляющая таблица лексического анализатора и список состояний и переходов лексического анализатора представлены в Приложении 1.

Финальные состояния и действия представлены ниже.



*Рис. 3 - Финальные состояния и действия*

Заданный фрагмент: const long $1SIZE = 21;

Конечный автомат, заданный таблицей:

*KeyWaords (const):*

*Format (“ ”):*

*KeyWaords (long):*

() ( (

*Format (“ ”):*

*Identificators ($1SIZE):*

*Format (“ ”):*

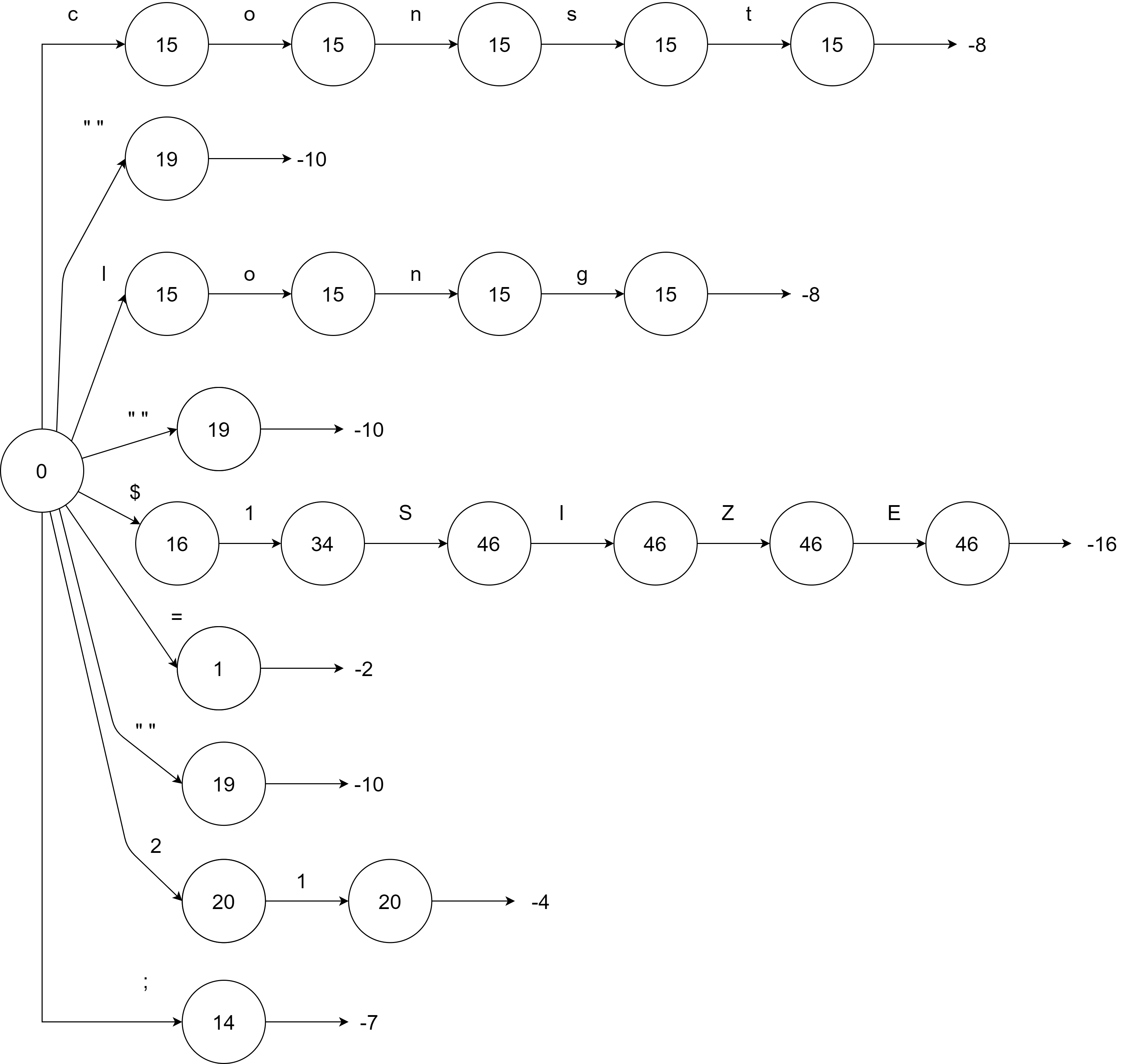
*Assignment (=):*

*Format (“ ”):*

*DeсadConst (21):*

*Limiter (;):*

Конечный автомат, заданный графом:



*Рис. 4 - Конечный автомат, заданный графом*

**Описание метаязыка формальных грамматик**

Формальной грамматикой называется совокупность , где:

алфавит терминальных символов, это есть конечное множество всех слов языка, порождаемого данной грамматикой. Понятие «терминальный» здесь обозначает неразложимость таких символов с точки зрения синтаксических правил;

алфавит нетерминальных символов, это есть конечное множество названий синтаксических конструкций, например: <предложение>, <выражение>, <список аргументов>, <условный оператор>, <тело функции>. Нетерминальные символы используются только в метаязыке, на котором описывает язык программирования;

начальный терминальный символ, это есть один из нетерминальных символов. Этим символом обычно обозначается наиболее общая синтаксическая конструкция, например: <правильная программа>;

система правил подстановки, это есть конечное множество пар цепочек вида α : β, причем цепочка α (левая часть правила) должна содержать хотя бы один нетерминальный символ;

**Свойства грамматик:**

1. **Рекурсивность**

Нетерминальный символ Х называется рекурсивным, если из него могут быть выведены цепочки, содержащие сам этот символ, т. е.

Х ⇒ μ Х η,

где μ и η – произвольные цепочки символов.

Грамматика называется рекурсивной, если рекурсивен хотя бы один нетерминальный символ, и не рекурсивной в противном случае.

1. **Однозначность**

Грамматика называется однозначной, если любое правильное предложение порождаемого ею языка имеет единственное дерево грамматического разбора, и неоднозначной в противном случае.

**Свойства символов грамматик:**

1. **Аннулируемость**

Нетерминальный символ называется аннулируемым, если из него может быть выведена пустая цепочка символов. В противном случае нетерминальный символ называется неаннулируемым.

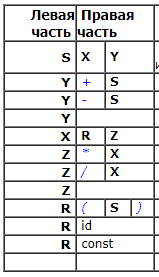
1. **Недостижимость**

Символ называется недостижимым, если он не появляется ни в одной цепочке символов, выводимой из начального нетерминала грамматики.

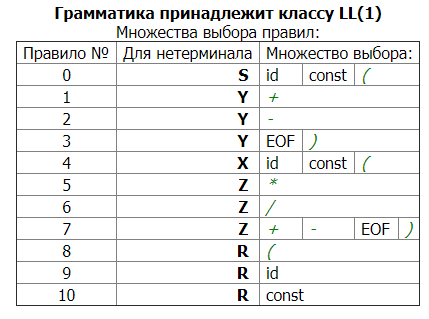
1. **Бесплодность**

Нетерминальный символ называется бесплодным, если из него не может быть выведена ни одна цепочка, состоящая только из терминальных символов.

**Пример из файла 3LL1.xml.**



*Рис. 5 - Правила грамматики*



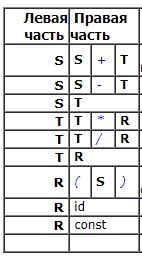
*Рис. 6 - Множество выбора правил языка*



*Рис. 7 - Множество предшественников и множество последователей*

В системе правил представленной в примере не содержится недостижимых и бесплодных грамматик, но, содержится два аннулируемых правила (Z и Y). Множества выбора для всех нетерминалов не пересекаются. Язык принадлежит классу LL(1) и пригоден для корректного построения и работы синтаксического акцептора.

**Пример из файла 4SLR1grammarArithmExpr.xml.**



*Рис. 8 - Правила грамматики*



*Рис. 9 - Множество выбора правил*



*Рис. 10 - Множество предшественников*

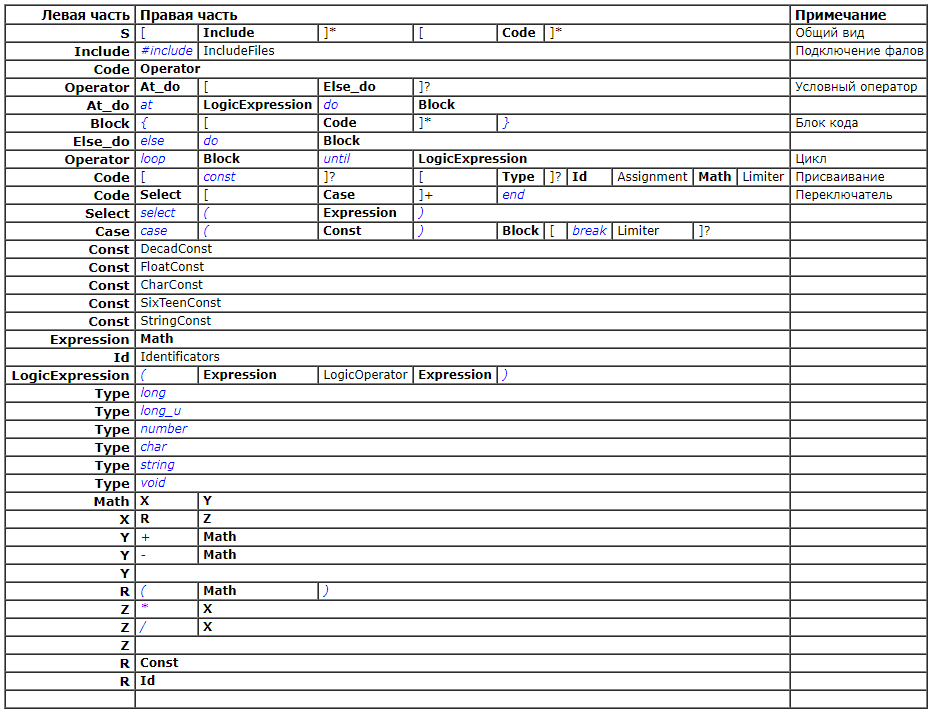


*Рис. 11 - Множество предшественников*

В данном примере во множестве выбора данной грамматике содержатся пересечения, поэтому грамматика не является LL(1), ее работа будет некорректна.

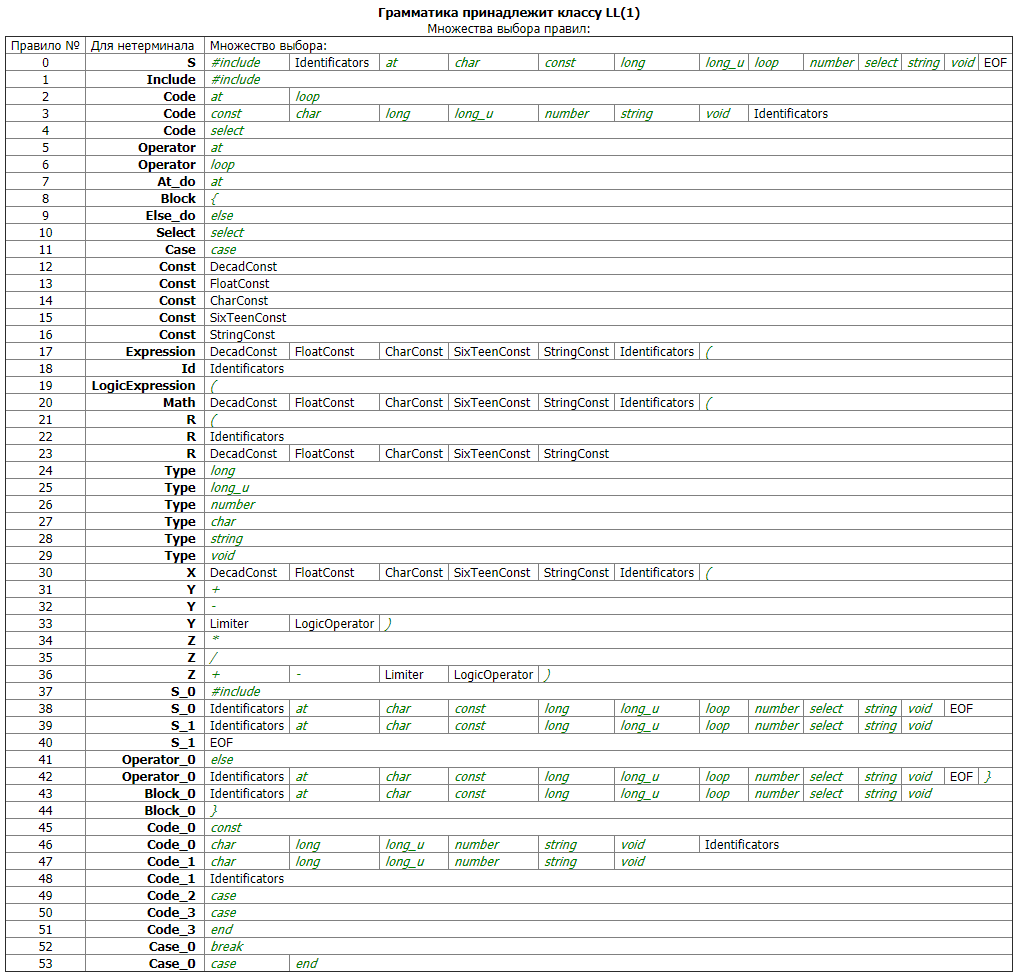
**Разработка грамматики программируемого языка**

Была частично разработана грамматика для языка, в соответствии с заданным вариантом курсовой работы. Она показана на Рис. 12.

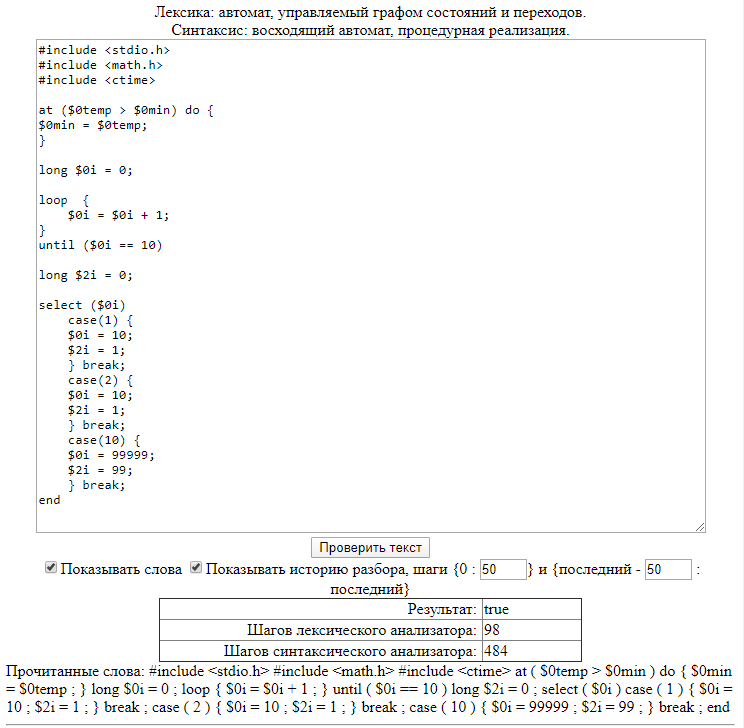


*Рис. 12 - Частично разработанная грамматика*

Были описаны грамматики операторов присваивания, условного оператора и оператора цикла. Множества выбора для всех нетерминалов не пересекаются. (Рис. 13) Описанная грамматика принадлежит классу LL(1).

  
*Рис. 13 - Множества выбора правил*

Проверка правильности, разработанной грамматики показана на Рис. 14.



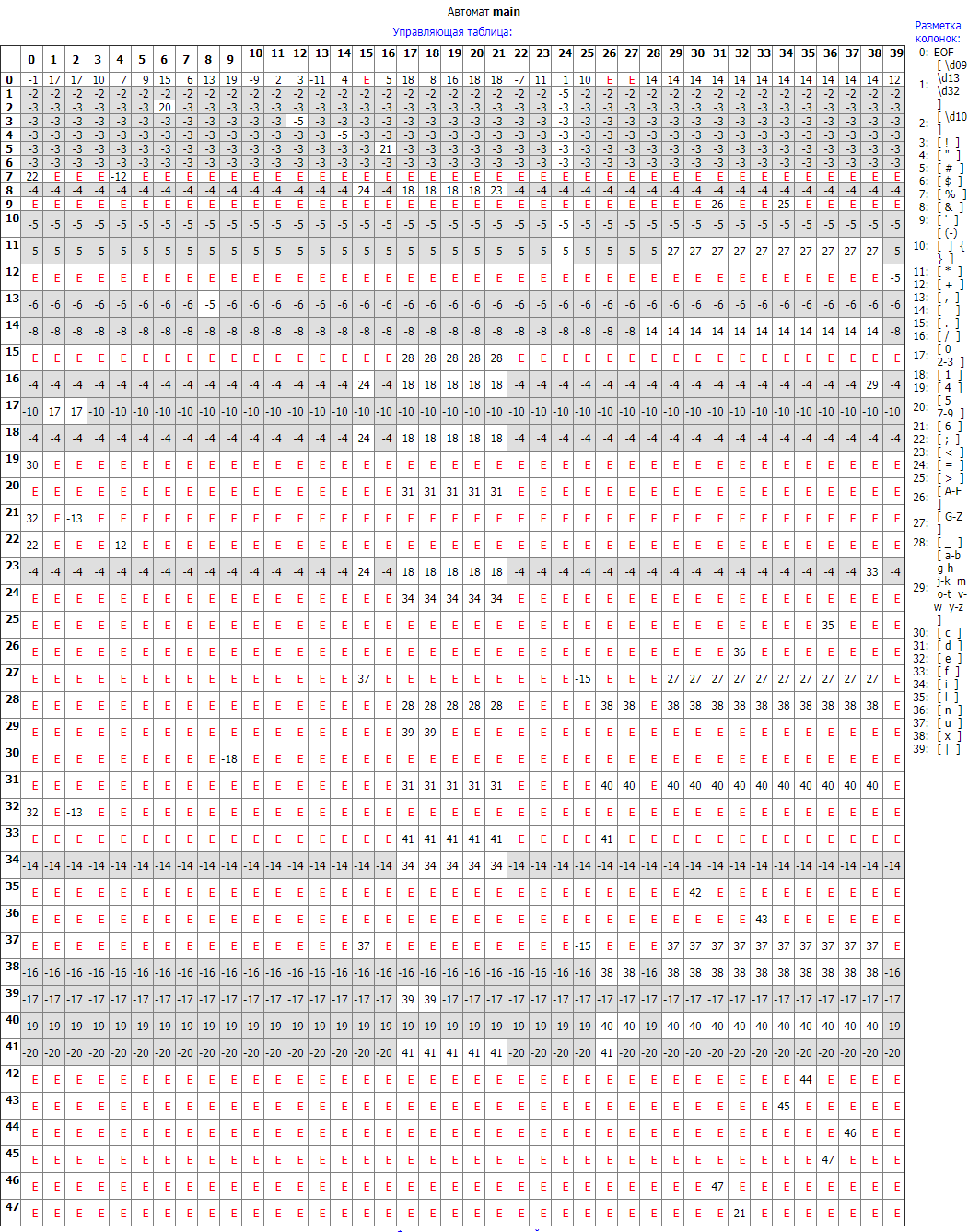
*Рис. 14 - Проверка правильности грамматики*

# Вывод

В ходе лабораторной работы были изучены основные понятия метаязыка формальных грамматик, свойств грамматик и нетерминальных символов. Была частично разработана грамматика для языка, в соответствии с заданным вариантом курсовой работы*.*

# Приложение 1

Управляющая таблица лексического анализатора:



# Приложение 1

Список состояний и переходов лексического анализатора:

