Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Институт информационных технологий

Кафедра ИС

# ОТЧЁТ

по лабораторной работе №2

ИССЛЕДОВАНИЕ СКАНЕРА ПРИ АНАЛИЗЕ ПРОСТЫХ ЯЗЫКОВЫХ КОНСТРУКЦИЙ

Выполнил:

ст. гр. ИС/б-21-2-о

Мовенко К. М.

Проверил:

Карлусов В. Ю.

Севастополь

2024

# ЦЕЛЬ РАБОТЫ

Изучить принципы построения и программирования лексического анализатора для простых языковых конструкций. Получить навыки практического построения лексического анализатора (сканера) на основе теории конечных автоматов. Освоить приёмы составления регулярных выражений для описания лексем. Закрепить навыки построения минимального КА, осуществляющего сканирование текста программ.

# Задание

* 1. Согласно варианту задания составить регулярное выражение, описывающее КА, который осуществляет лексический анализ. Составить эскизные выражения для описания дизъюнктивных компонент. Построить управляющую таблицу МДКА, редактировать программу из предыдущей работы под нужды лексического анализа и текст диагностических сообщений, генерируемых в его ходе. Разработать схему кодирования лексем;
  2. Согласовать текст диагностических сообщений с преподавателем, сконструировать текст программы сканера на основании программы МДКА. Выявить и устранить синтаксические ошибки;
  3. Разработать тестовые примеры правильных и неправильных лексем языка, уточнить диагностические сообщения. Подготовить мероприятия по хронометрированию работы сканера. Продумать структуру выходной информации программы на предмет использования другими программными модулями компилятора;
  4. Тестировать и отладить программу. Экспериментально определить время работы сканера для качественно различных структур строк и характера ошибок;

# Вариант Задания

Вариант задания – 43.

**Тип идентификатора** − оканчивается символом "s", которому не предшествует цифра.

**Тип константы** – X. Шестнадцатеричное число. Формат 0x012…9a…f.

Таблица 1 – Фрагмент программы для анализа

|  |  |  |  |
| --- | --- | --- | --- |
| **Служебные слова** | **Разделители** | | **Фрагмент программы для анализа** |
| однолитерные | двулитерные |
| WAIT  SIGNAL | (  )  >  =  +  **−**  % | >=  <= | WAIT(S)  K=K+5.3 %  C=K+C+1 %  SIGNAL(S<=0xf) |

# ХОД РАБОТЫ

## Определение лексем и назначение им кодов

Всем лексемам были назначены коды относительно их классов и уникальности. Класс однолитерных лексем получил префикс 5, класс двулитерных − префикс 6.

* WAIT – 100
* SIGNAL – 200
* идентификатор, <iden> – 300
* константа, <data> – 400
* ( – 501
* ) – 502
* > – 503
* = – 504
* + – 505
* % – 506
* >= – 601
* <= – 602

## Построение регулярного выражения по варианту исходных данных

Общая структура регулярного выражения для описания КА, дополненного функциями сканера, примерно такова:

Обозначения указаны в методических указаниях.

**Служебные слова** являются самоопределяющимися цепочками, составленными из литер {W, A, I, T, S, G, N, L} – WAIT, SIGNAL.

Эскизно **идентификатор** (переменная) выглядит так:

*{Б* *˅ Б {Ц} Б} S,*

где Б – любая латинская буква, Ц – любая десятичная цифра, S – буква s.

Шестнадцатеричная **константа** описывается следующим образом:

*0x (ц ˅ б) {ц ˅ б},*

где ц – любая десятичная цифра, б – любая латинская буква в диапазоне a-f.

С учётом того, что конечный автомат должен отличать отдельные буквы и цифры на фоне других, имеем:

1. **Множество букв** , где б – латинская буква в диапазоне B-F, δ − буква латинского алфавита, не совпадающая с другими литерами;
2. **Множество цифр** , где ц – любая цифра в диапазоне 1-9;
3. **Множество «не буква, не цифра»** , где – литера, не принадлежащая алфавиту конечного автомата;
4. **Множество «не 16-ричная цифра»** .

Окончательно получаем дизъюнктивные члены выражения, описывающие конечный автомат:

1. WAIT L1 – первое служебное слово;
2. SIGNAL L1 – второе служебное слово;
3. {(W ˅ A ˅ I ˅ T ˅ S ˅ G ˅ N ˅ L ˅ X ˅ б ˅ δ) ˅ (W ˅ A ˅ I ˅ T ˅ S ˅ G ˅ N ˅ L ˅ X ˅ б ˅ δ){0 ˅ ц}(W ˅ A ˅ I ˅ T ˅ S ˅ G ˅ N ˅ L ˅ X ˅ б ˅ δ)}S L1 – переменная;
4. 0X {0 ˅ ц ˅ A ˅ б} (0 ˅ ц ˅ A ˅ б) L2 – константа;
5. – однолитерные и двулитерные разделители.

Дизъюнктивные члены были подставлены в общее регулярное выражение.

## Построение функции переходов МДКА

Была построена разметка регулярного выражения (рисунок 1). Предостовные места для минимизации были отмечены жёлтым. Затем разметка была минимизирована (рисунок 2).

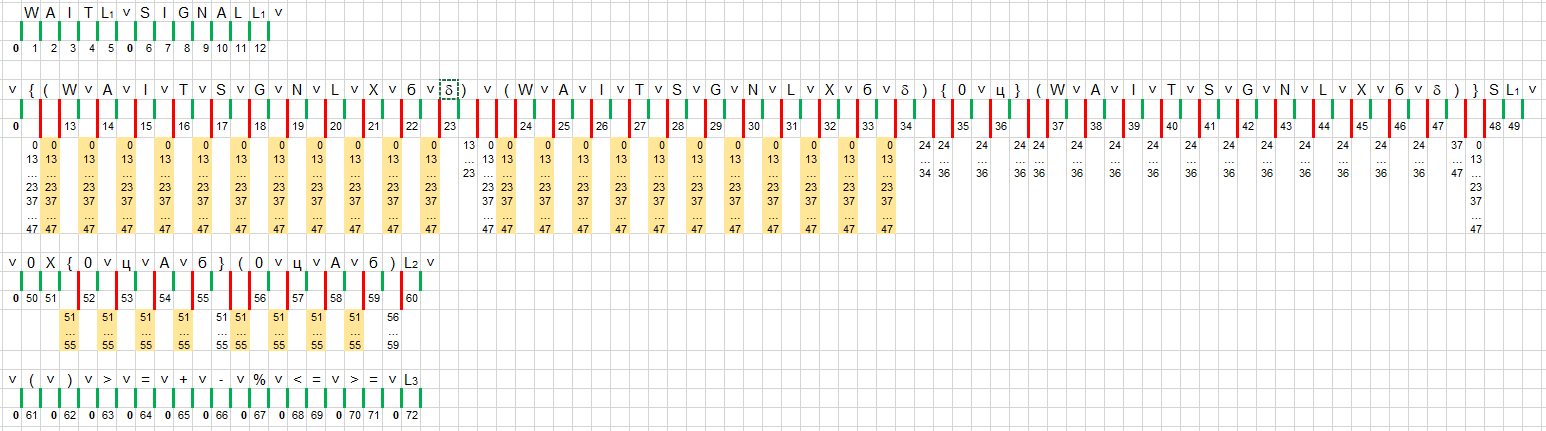


Рисунок 1 – Разметка регулярного выражения

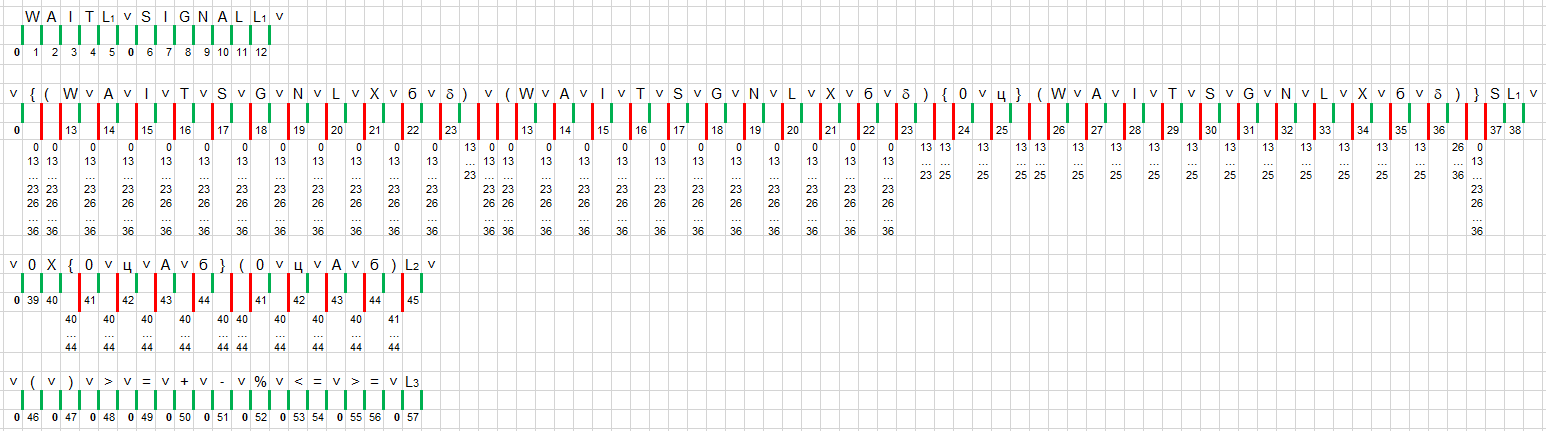


Рисунок 2 – Минимизация разметки

По разметке была построена таблица переходов конечного автомата (рисунок 3). Автомат недетерминированный, неоднозначные переходы были выделены жёлтым.

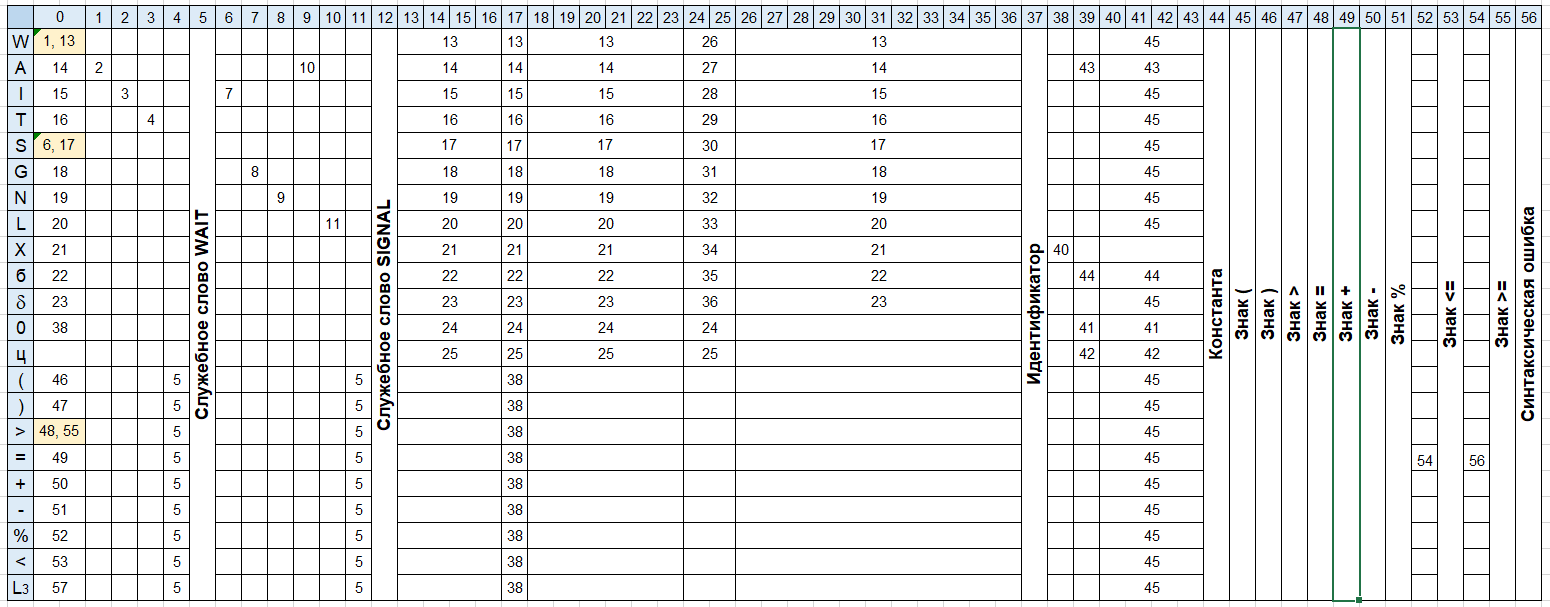


Рисунок 3 – Таблица переходов НДКА

Для преобразования НДКА в ДКА были построены графы фрагментов КА, относящихся к служебным словам (рисунок 4, рисунок 5).

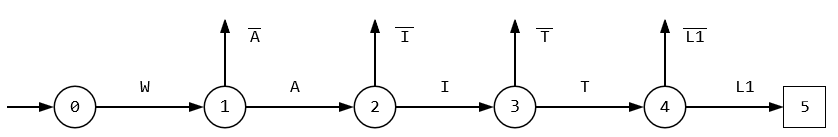


Рисунок 4 – Фрагмент КА, относящийся к разбору служебного слова WAIT

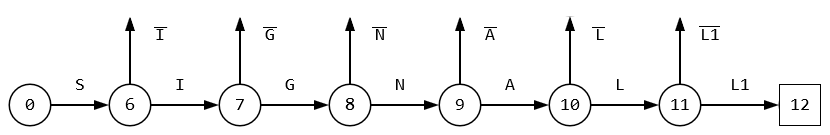


Рисунок 5 – Фрагмент КА, относящийся к разбору служебного слова SIGNAL

С помощью графов таблица переходов была дополнена таким образом, чтобы исключить недетерминированность автомата (рисунок 6).

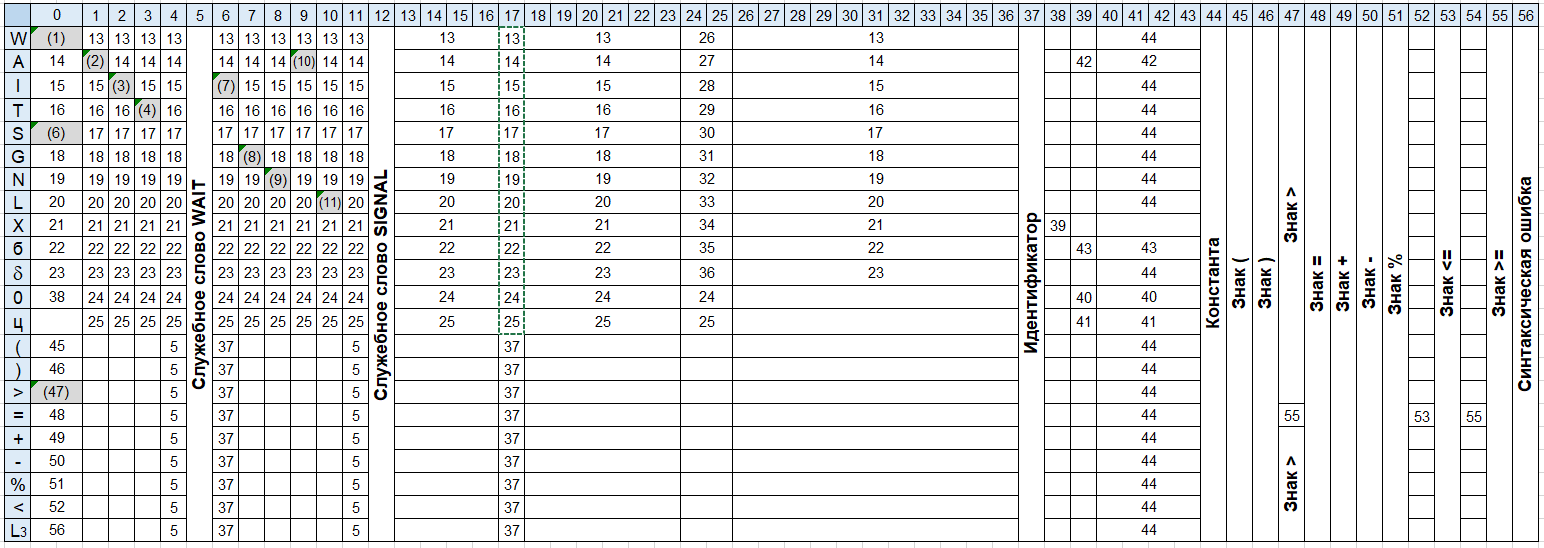


Рисунок 6 – Таблица переходов ДКА

Таблица переходов была минимизирована. Для этого…

Для случаев, когда образ функции переходов КА пуст, были введены коды ошибок, которые в ходе работы сканера помогут генерировать осмысленные диагностические сообщения:

1. Некорректный фрагмент цепочки. Состояние 0, строки, соответствующие литерам {ц}, код 801;
2. Ошибка в служебном слове. Состояния {1, 2, 3, 6, 7, 8, 9}, строки, соответствующие L1, код 802;
3. Ошибка в написании имени переменной. Состояния {11, 13, 14}, строки, соответствующие L1 и литерам {0, ц}, код 803;
4. Ошибочная константа. Состояния {15, 16}, строки, соответствующие множеству L2. Состояние 17, литера X. Код 804;
5. Ошибка записи двулитерных операторов. Состояния {18, 19}, все символы кроме ‘=’, код 805.

Также для упрощения подготовки тестовых данных с позиции их наглядности в алфавит входных символов конечного автомата был введён пробел, пополняющий множество L1.

## Разработка программы сканера

.

# ВЫВОД

В ход работы были изучены принципы построения и программирования лексического анализатора для простых языковых конструкций. Получены навыки практического построения лексического анализатора (сканера) на основе теории конечных автоматов.