МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

Факультет прикладной математики, информатики и механики

Кафедра математического обеспечения ЭВМ

Лексический анализатор

ЛАБОРАТОРНАЯ РАБОТА

02.03.02 Фундаментальная информатика и информационные технологии

Инженерия программного обеспечения

Студент Смехнев И. Ю.

Воронеж 2024

1. **Постановка задачи**
   Цель работы: разработка программы, реализующей конечный автомат для выделения лексем из строк, состоящих из цифр 1, 2, 3 в произвольной последовательности. Программа должна идентифицировать и извлекать подстроки, состоящие из неубывающих цифр. Строки проверяются на соответствие заданной грамматике. Программа должна считывать строки из файла и выводить результаты проверки.

2. **Разработка программы**

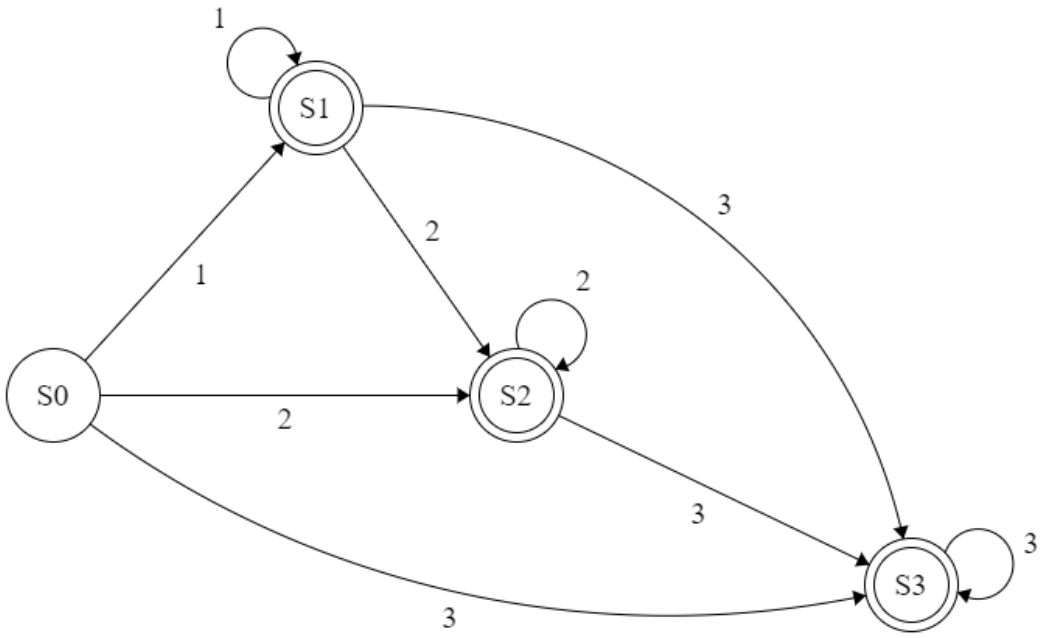
   2.1. Модель и метод решения задачи

   Для решения задачи используется модель конечного автомата. Это абстрактная машина, которая на основе считываемого символа переходит из одного состояния в другое. Автомат имеет три состояния:

   - S0: начальное состояние, принимает цифры 1, 2, 3;
   - S1: состояние, принимающее цифры 1, 2, 3, в случае отсутствия входящего символа является конечным состоянием;
   - S2: состояние, принимающее цифры 2, 3, в случае отсутствия входящего символа является конечным состоянием;
   - S3: состояние, принимающее цифру 3, в случае отсутствия входящего символа является конечным состоянием;
   - ERROR: ошибочное состояние, в которое программа переходит при несоответствии символов заданной грамматике.

   Программа реализована с использованием структуры данных для хранения состояний и функции для перехода между состояниями на основе текущего символа строки.

   Ниже представлена диаграмма конечного автомата:



2.2. Описание структуры данных

Структура программы основана на перечисленном типе State, описывающем состояния конечного автомата:

- S0 – начальное состояние
- S1 – состояния обработки цифр
- S2 – состояния обработки цифр
- S3 – состояния обработки цифр
- ERROR – состояние ошибки

Входными данными для обработки являются строки, считываемые из файла. Каждая строка проверяется на соответствие допустимой лексике.

2.3. Алгоритм решения задачи

1) Программа считывает строку из файла
2) Строка обрабатывается посимвольно, для каждого символа совершается переход из одного состояния в другое в соответствии с таблицей переходов:
    - В состоянии S0 программа ожидает символ цифры 1-3. После обработки:
        - Цифра ‘1’ – переход в состояние S1
        - Цифра ‘2’ – переход в состояние S2
        - Цифра ‘3’ – переход в состояние S3
        - Любой другой символ – переход в состояние ERROR
    - В состоянии S1 программа ожидает символ цифры 1-3. После обработки:
        - Цифра ‘1’ – остаётся в состояние S1
        - Цифра ‘2’ – переход в состояние S2
        - Цифра ‘3’ – переход в состояние S3
        - Любой другой символ – переход в состояние ERROR
    - В состоянии S2 программа ожидает символ цифры 2, 3. После обработки:
        - Цифра ‘2’ –остаётся в состояние S2
        - Цифра ‘3’ – переход в состояние S3
        - Любой другой символ – переход в состояние ERROR
    - В состоянии S2 программа ожидает символ цифру 3. После обработки:
        - Цифра ‘3’ – переход в состояние S3
        - Любой другой символ – переход в состояние ERROR
3) Если в ходе обработки символов программа перешла в состояние ERROR, то строка считается недопустимой.
4) Если после обработки всех символов программа находится в состоянии S3, строка считается допустимой.
5) Вывод результата.

3. Руководство программиста

Программа состоит из основной функции main(), функции changeState(), в которой происходит переход между состояниями конечного автомата, и функции checkStates(), которая проверяет входящую строку на корректность.

Программа написана на языке программирования C++. Она использует стандартные библиотеки:

- iostream для консольного ввода и вывода;
- fstream для файлового ввода и вывода;
- string для работы со строками.

4. Руководство пользователя

Для работы программы пользователю необходимо подготовить файл file.txt, в котором должны содержаться строки. Данные строки будут считаны с файла программой и проверены на соответствие заданной грамматике. Программа будет выводить переходы по состояниям, а также результат проверки для каждой строки.

5. Тестирование программы и его результаты

Входные данные:

```
1222333
13
2233
21
012
1a2b
```

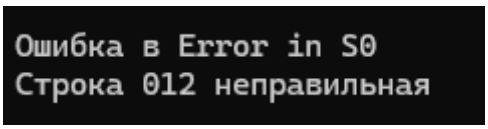
Выходные данные:

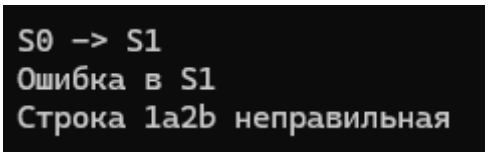
```
S0 -> S1
S1 -> S2
S2 -> S2
S2 -> S2
S2 -> S3
S3 -> S3
S3 -> S3
Строка 1222333 правильная
```

```
S0 -> S1
S1 -> S3
Строка 13 правильная
```

```
S0 -> S2
S2 -> S2
S2 -> S3
S3 -> S3
Строка 2233 правильная
```

```
S0 -> S2
Ошибка в S2
Строка 21 неправильная
```





6. **Результаты работы программы и их анализ**
   Программа успешно считала и распознала строки, соответствующие заданной грамматике, и корректно обработала недопустимые строки.

7. **Выводы**
   Разработанная программа успешно решает задачу распознавания строк, состоящих из кавычек и восьмеричных констант. Программа выполнена в виде конечного автомата, что делает ее легко расширяемой для более сложных грамматик. Тестирование показало корректность работы программы для различных тестовых входных данных.