

Отчет по лабораторной работе № 11 по курсу “Фундаментальная информатика”

Студент группы М80-109Б-22 Малько Павел Алексеевич, № по списку 9

Контакты e-mail: malko.pavel07@gmail.com

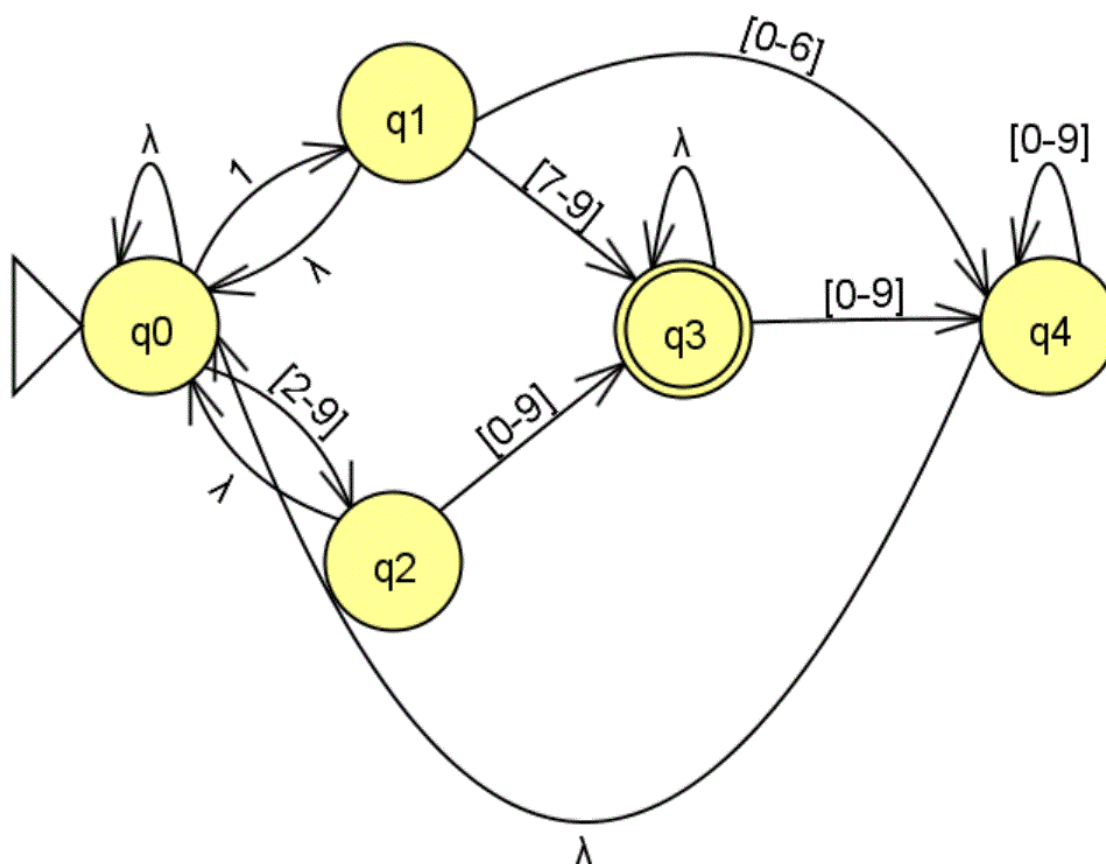
Работа выполнена: «20» декабря 2022г.

Преподаватель: каф. 806 Сысоев Максим Алексеевич

Отчет сдан « » _____ 20__ г., итоговая оценка _____

Подпись преподавателя _____

- Тема:** Обработка последовательности литер входного текстового файла. Простейшие приёмы лексического анализа. Диаграммы состояний и переходов.
- Цель работы:** Составить программу для анализа и обработки введённого текста в соответствии с заданием
- Задание (вариант № 32):** В введённом тексте выделить все числа по модулю от 17 до 99, распечатать их на русском языке.
- Оборудование (студента):**
Процессор *amd ryzen 5500u* с ОП 7851 Мб, НМД 1024 Гб. Монитор 1920x1080
- Программное обеспечение (студента):**
Операционная система семейства: *linux*, наименование: *kali*, версия 2022.3
интерпретатор команд: *bash* версия 0.16.1.
Система программирования -- версия --, редактор текстов *emacs* версия 25.2.2
Утилиты операционной системы --
Прикладные системы и программы --
Местонахождение и имена файлов программ и данных на домашнем компьютере --
- Идея, метод, алгоритм** решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)



Для решения предлагается использовать идею конечных автоматов, которые задаются своими состояниями и

условиями для переходов между ними. Такие автоматы можно программировать, используя перечисления enum для обозначения множества состояний(используется typedef псевдоним State для удобства) а также встроенная конструкция switch..case языка C.

На вход такому автомату подаётся символ в цикле, на каждой итерации программа помнит текущее состояние и таким образом реализуется этот КА. Данный автомат обнаруживает подходящее число, и при помощи вспомогательной процедуры программа печатает значение числа. Далее автомат применяется снова для поиска очередного числа...

7. Сценарий выполнения работы

1. Понять работу получения символов из потока ввода, конструкций switch..case(а особенно специфического синтаксиса), перечисления enum
2. Реализовать работу вспомогательной процедуры для печати числа, убедиться в её работоспособности
3. Построить КА для обнаружения чисел
4. Реализовать поведение данного КА в программе

Unit-тесты

```
void write_num(int prev, int current) {  
    assert( (prev<=9) && (prev>=1) && (current>=0) && (current<=9) );  
    if (prev==1) {
```

Нужно проверять, что на вход процедуре печати подаются именно подходящие числа(процедура эта вызывается в финальном состоянии конечного автомата и таким образом данный тест позволит убедиться в корректной работе самого КА)

8. Дневник отладки должен содержать дату и время сеансов отладки и основные события (ошибки в сценарии и программе, нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание
-	-	-	-	-	-	-

9. Замечания автора по существу работы

Кроме использования идеи конечных автоматов, есть подход, который я считаю более простым. Нужно завести массив-буфер из 4 элементов и проверять, что два крайних элемента не являются цифрами, а два средних образуют подходящее число.

Однако, более сложное поведение, уверен, более эффективно реализуемо при помощи КА.

10. Выводы

Откровенно говоря, без конечных автоматов данная работа не кажется интересной. С другой стороны, использование именно конечных автоматов здесь не совсем наглядно в том смысле, что есть в данном случае возможны чуть более простые реализации. Причём в случае реализации при помощи массива-буфера даже не страдает эффективность, так как в буфере 4 элемента и ровно столько же состояний у итогового конечного автомата.

В то же время, как я узнал во время работы, у конечных автоматов есть довольно интересное применение в разработке игр – к примеру, с помощью абстрактных конечных автоматов можно реализовывать поведение искусственных интеллектов персонажей(<https://gamedevelopment.tutsplus.com/tutorials/finite-state-machines-theory-and-implementation--gamedev-11867>).

Возможно, следует увеличить сложность задания, чтобы польза и применимость именно Конечных Автоматов была более наглядна.

Недочёты при выполнении задания могут быть устранены следующим образом: --

Подпись студента _____