Выполнение задания включает разработку программного средства, тестирование его на наборе данных и написание отчёта по работе.

Отчёт должен содержать:

1)      титульный лист;

2)      номер варианта и текст задания;

3)      описание алгоритма решения задачи с иллюстрацией его на своём примере;

4)      описание основных блоков программы;

5)      текст программы;

6)      результаты тестирования программы;

7)      распечатку файла результатов – 2–3 примера.

Высылаемый на проверку преподавателю архив должен содержать отчёт и все файлы программы – все модули, exe-модуль, файлы данных, файлы результатов.

Внимание!!!  
Необходимо предусмотреть обработку ошибок.           
Никакие входные данные не должны нарушать работу программы!! Программа не должна «зависать» или прекращать выполнение по неизвестной причине – обязательна выдача соответствующей диагностики.

При выполнении задания следует использовать средства объектно-ориентированного программирования. Выбор конкретного средства разработки оставляется за студентом.

Программа должна управляться посредством меню, в котором должны присутствовать следующие пункты: "Автор", "Тема" (с полной информацией о разработчике и теме задания), "Данные" (выбор способа задания исходных данных – чтение из файла или ввод с клавиатуры), "Расчёты", "Запись результатов в файл" – и другие, определяемые конкретным заданием. При вводе данных с клавиатуры необходимо использовать соответствующую форму, а также предусмотреть возможность вызова справки с примером формата данных. При чтении из файла – должна открываться своя папка. Все результаты расчётов должны отображаться на экране и выводиться в файл (по требованию пользователя). При введении автором каких-либо ограничений (размер алфавита и т.п.) они должны быть описаны в пояснительной записке и в соответствующем пункте меню.

Задание

Написать программу для автоматического построения детерминированного конечного автомата (ДКА) по словесному описанию языка.

Вход программы: алфавит языка, обязательная начальная подцепочка, кратность длины всех цепочек языка, цепочки для распознавания.

Выход: построенный ДКА (все 5 элементов), результат проверки цепочек.

Подробно:

Язык задан своим алфавитом, обязательной начальной цепочкой и указанием кратности длины всех цепочек языка. В начальной цепочке не должно находиться символов, не содержащихся в алфавите. В крайнем случае она может быть и пустой. Кратность длины цепочек задаётся любым натуральным числом. Кратность, равная единице, равносильна отсутствию ограничений на длину цепочек.

Программа должна:

1.            по предложенному описанию регулярного языка строить ДКА, распознающий этот язык, в том виде, как он рассматривался в теории, раздел 2.2.2;

2.            с помощью построенного ДКА проверять вводимые пользователем цепочки на их принадлежность этому языку.

ДКА может быть полностью или неполностью определённым – по желанию разработчика. Функция переходов ДКА может изображаться в виде таблицы и графа, вариант вида отображения должен выбираться в меню.

После построения ДКА пользователь может вводить произвольные цепочки для проверки их на принадлежность исходному языку. Разбор цепочек автоматом следует поэтапно отображать на экране в виде последовательной смены конфигураций в соответствии с лабораторной работой №2.

Рассмотрим пример построения ДКА.

|  |
| --- |
|  |
|  | C:\Учеба\4 семестр - январь - июль 2019 (9 семестр)\Теория языков программирования и методы трансляции (ДВ 4.1)\course798\pages\img\p01.png |

Задан язык: алфавит {a,b,c}, обязательная начальная цепочка ‘aab’, длина цепочек кратна 2. Анализируем задание: язык будет состоять из цепочек чётной длины, начинающихся с ‘aab’, например {aaba, aabb, aabc, aabaaa, aabaab, aabccc,…}. Тогда ДКА должен иметь вид M(Q,{a,b,с},,q0,F), множество состояний Q и заключительные состояния F определятся в процессе построения. Разберёмся с построением функции переходов. Очевидно, что пустая цепочка в языке не содержится (поскольку есть непустая обязательная начальная цепочка). Сначала определимся с минимальной цепочкой языка и построим для неё граф переходов. Это ‘aab’ и еще один (любой) символ алфавита.

После этой цепочки допускается её наращивание парами любых символов алфавита. Возврат из последнего состояния в q3 по любому символу алфавита позволит добавлять такие пары символов. Функция переходов δ построена:

C:\Учеба\4 семестр - январь - июль 2019 (9 семестр)\Теория языков программирования и методы трансляции (ДВ 4.1)\course798\pages\img\p02.pngQ={q0,q1,q2,q3,q4}, F={q4}.

В табличном виде она представляется очевидным образом.