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.

## Курсовая работа

**Теория языков программирования и методы трансляции**

титульный лист;

номер варианта и текст задания;

описание алгоритма решения задачи с иллюстрацией его на своём примере;

описание основных блоков программы;

текст программы;

результаты тестирования программы;

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.

При выполнении задания следует использовать средства объектно-ориентированного программирования. Выбор конкретного средства разработки оставляется за студентом. При наличии определённых требований к программному обеспечению студенту следует уведомлять об этом преподавателя.

Рекомендуется при разработке программного средства использовать материалы лабораторных работ (в зависимости от темы задания).

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.

## Вариант 8

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

Вход программы: алфавит языка, обязательная конечная подцепочка, цепочки для распознавания.

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

Подробно:

Язык задан своим алфавитом и обязательной конечной подцепочкой всех цепочек языка. В конечной подцепочке не должно находиться символов, не содержащихся в алфавите. В крайнем случае она может быть и пустой.

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 должен быть полностью определённым. Функция переходов ДКА может изображаться в виде таблицы или графа, вариант вида её представления выбирается разработчиком.

Наиболее простой способ построения такого ДКА состоит в том, чтобы сначала по описанию языка построить НКА (недетерминированный конечный автомат), а затем преобразовать его согласно рассмотренному в разделе 2.2.2 алгоритму. При выборе такого способа построения ДКА промежуточный результат в виде НКА необходимо также отображать на экране по просьбе пользователя.

По желанию автора допускаются и другие способы построения ДКА.

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.

Задан язык: алфавит {a,b,c}, обязательная конечная подцепочка ‘aaba’. Анализируем задание: язык будет состоять из цепочек любой длины, заканчивающихся на ‘aaba’, например {aaba, аaaba, baaba, caaba, aaaaba, …}. Тогда ДКА должен иметь вид M(Q,{a,b,с},δ,q0,F), множество состояний Q и заключительные состояния F определятся в процессе построения. Разберёмся с построением функции переходов δ. Очевидно, что пустая цепочка в языке не содержится (поскольку есть непустая обязательная конечная цепочка). Сначала определимся с минимальной цепочкой языка – это ‘aaba’, и построим для неё граф переходов.

q0

q1

q4

b

a

q2

q3

а

a

Если выбрать способ с предварительным построением НКА, то такой автомат выглядит очевидным образом. Сначала могут быть прочитаны любые символы алфавита в любом количестве, а затем конечная подцепочка:

q0

q1

q4

b

a

а,b,с

q2

q3

а

a

Недетерминированность автомата вызвана тем, что из начального состояния существует два перехода по одному символу алфавита (‘a’). Осталось преобразовать построенный автомат в детерминированный. Для этого построим таблицу переходов:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | ***вход*** | | | Исходную таблицу переходов отделим от остальной части жирной линией.  Для упрощения процесса будем создавать не все возможные новые состояния, которые могут получиться в результате сочетаний исходных состояний, а только те, которые реально возникают при построении. Сначала это единственное состояние q0q1 – занесём его в таблицу. Затем последовательно появятся q0q1q2, q0q3, q0q1q4. Все состояния исходного автомата, кроме q0, оказались недостижимыми. В таблице они выделены синим цветом. Удалим их. |
| ***состояние*** | | a | b | c |
| q0 | | {q0,q1} | {q0} | {q0} |
| q1 | | {q2} | – | – |
| q2 | | – | {q3} | – |
| q3 | | {q4} | – | – |
| q4 | | – | – | – |
| q0q1 | A | {q0q1q2} | {q0} | {q0} |
| q0q1q2 | B | {q0q1q2} | {q0q3} | {q0} |
| q0q3 | C | {q0q1q4} | {q0} | {q0} |
| q0q1q4 | D | {q0q1q2} | {q0} | {q0} |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ***вход*** | | | Новые состояния для удобства переобозначим A, B, C, D. Заключительными состояниями станут те, которые содержат q4. Здесь такое состояние одно – D. Новая таблица переходов представлена слева:  c  a  q0  A  D  b  a  b,с  B  C  a  a  b,c  a  b,c  b,c |
| ***состояние*** | a | b | c |
| q0 | {A} | {q0} | {q0} |
| A | {B} | {q0} | {q0} |
| B | {B} | {C} | {q0} |
| C | {D} | {q0} | {q0} |
| D | {B} | {q0} | {q0} |

Граф переходов построен по таблице:

Q={q0,A,B,С,D }, F={D}.

ДКА построен.