ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ КЕМЕРОВСКОЙ ОБЛАСТИ

**ГБОУ СПО ЮРГИНСКИЙ ТЕХНОЛОГИЧЕСКИЙ КОЛЛЕДЖ**

Отделение ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ И РЕМОНТ РЭТ

УТВЕРЖДАЮ

Заведующий методическим

кабинетом

\_\_\_\_\_\_\_\_\_\_\_\_ Г.П.Вдовина

\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

**Решение задач по теории конечных автоматов**

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению практических работ №2,3

|  |  |
| --- | --- |
| МДК 01.02 | Математический аппарат для построения компьютерных сетей |
| Специальность | 230111 Компьютерные сети |

2013

Методические указания составлены в соответствии с Федеральным государственным образовательным стандартом среднего профессионального образования по специальности 230111 Компьютерные сети, утверждённым приказом Минобрнауки России от 16.04.2008г. и на основании рабочей программы по ПМ 01 Участие в проектировании сетевой инфраструктуры, утвержденной 1 сентября 2012г.

ОДОБРЕНО ЦК специальных

дисциплин отделения ТОиРРЭТ

Протокол №\_\_\_\_ от\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_ г.

Председатель ЦК специальных

дисциплин отделения ТОиРРЭТ \_\_\_\_\_\_\_\_\_\_\_\_\_ Т.П.Каретникова

СОСТАВИТЕЛЬ

Преподаватель

специальных дисциплин ГБОУ СПО ЮТК \_\_\_\_\_\_\_\_\_\_\_ М.В.Поликарпочкин

РЕЦЕНЗЕНТЫ

Преподаватель

специальных дисциплин ГБОУ СПО ЮТК \_\_\_\_\_\_\_\_\_\_\_\_\_А.И. Обжорин

Заведующий

лабораторией стандартизации \_\_\_\_\_\_\_\_\_\_\_\_\_ Е.Н.Соловьева

Зарегистрировано в методическом кабинете \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

Заведующий

методическим кабинетом \_\_\_\_\_\_\_\_\_\_\_\_\_ Г.П. Вдовина

СОДЕРЖАНИЕ

Оглавление

[1 Теоретические сведения 4](#_Toc367130745)

[1.1. Основные понятия и определения 4](#_Toc367130746)

[1.2. Формальное описание КА 7](#_Toc367130747)

[1.3. КА – модель цифровых автоматов 8](#_Toc367130748)

[1.4. КА-распознаватели 9](#_Toc367130749)

[1.5. Формальное описание КА-распознавателя 11](#_Toc367130750)

[1.6. Примеры КА-распознавателей 14](#_Toc367130751)

[1.7. Задачи 17](#_Toc367130752)

[1.8. КА-преобразователи 19](#_Toc367130753)

[1.8.1. Классификация автоматов 19](#_Toc367130754)

[1.8.2. Абстрактный синтез автоматов 19](#_Toc367130755)

[1.8.3. Примеры КА–преобразователей и их синтез 20](#_Toc367130756)

[1.8.4. Задачи 25](#_Toc367130757)

[2 Содержание отчета 27](#_Toc367130758)

[3 Список источников 27](#_Toc367130759)

[Основная литература 27](#_Toc367130760)

[Интернет-ресурсы 27](#_Toc367130761)

[ПРИЛОЖЕНИЕ А Пример оформления титульного листа для отчета по практической работе 28](#_Toc367130762)

# Теоретические сведения

## *1.1. Основные понятия и определения*

На протяжении последних десятилетий велись и ведутся интенсивные работы по созданию и использованию различных систем и устройств для переработки дискретной информации. Преобразователи дискретной информации широко применяются в качестве различного рода технических автоматов, вычислительных устройств и их функциональных блоков, устройств управления роботами, управляющих объектами по заданному алгоритму и т.д. Широкий класс таких преобразователей объединяется под общим названием *автомат*. При одном из подходов к определению термина автомат его истолковывают как математическую модель реальных преобразователей дискретной информации. Функционирование его состоит в том, что последовательность z1, z2… символов конечного или в общем случае бесконечного алфавита Z, поступающая на вход, вызывает на его выходе определенную последовательность ω1, ω2, … символов того же или другого алфавита. Таким образом, самой общей математической моделью преобразователя дискретной информации (ПДИ) является последовательностная функция φ: Z\*→W\*, отображающая множество Z\* всех последовательностей символов алфавита Z в другое множество W\* последовательностей символов алфавита W (рис. 1.1).



Рис.1.1. Общая функциональная Рис.1.2. Формальная……… модель ПДИ модель ПДИ

Такая интерпретация позволяет схематично представить преобразователь как устройство, реализующее отображение одного множества на другое (рис. 1.2).

Отображение φ называется *алфавитным (автоматным)* отображением или *алфавитным оператором*. Результат преобразования *вход => выход* (рис. 1.2) зачастую зависит не только от того, какая информация в данный момент появилась на входе, но и от того, что происходило раньше, от предыстории преобразования. Чтобы как-то запоминать предыстории и отличать одну от другой, преобразователь должен иметь память. Для этого в модель (рис. 1.2) вводят алфавит состояний Q={q1, q2,…,qm} и такой преобразователь (рис. 1.3) называют *конечным автоматом (КА)*, если множество входных сигналов Z и множество возможных состояний Q конечны. Преимущество конечности числа состояний заключается в том, что устройство можно реализовать, имея ограниченные ресурсы, либо аппаратно (в “железе”), либо в виде простой программы, принимающей решение на основе ограниченного количества данных или текущей команды машинного кода. Таким образом, конечные автоматы включают набор состояний и переходов между ними, зависящих от входных данных.



Рис.1.3. Конечный автомат Рис.1.4. КА, моделирующий

переключатель

*Конечный автомат* – это устройство, работающее в дискретные моменты времени (такты). На вход его в каждом такте поступает один из возможных сигналов, а на выходе вырабатывается выходной сигнал, являющийся функцией его текущего состояния (q) и поступившего входного сигнала (z), то есть ω=λ(q,z). Внутреннее состояние автомата также меняется, и новое состояние q=(t+1) является функцией δ тех же двух аргументов: q(t+1)= δ(q,z).

Понятие текущего состояния q играет важную роль в работе КА, так как определяет его будущее поведение – реакцию на последующие входные сигналы.

Простейшим нетривиальным конечным автоматом является переключатель «включено-выключено». Это устройство помнит свое состояние, и от этого состояния зависит результат нажатия кнопки. Из состояния «выключено» нажатие кнопки переводит переключатель в состояние «включено», и наоборот.

Конечно-автоматная модель переключателя представлена на рис.1.4

Как и для всех конечных автоматов, состояния обозначены кружками, а переходы между ними – дугами. Дуги отмечаются «входными символами», задающими внешние воздействия на систему (устройство). В примере это *Нажатие*, что означает нажатие на кнопку переключателя. Стрелки указывают, что всякий раз при *Нажатии* система переходит из одного состояния в другое. Одно из состояний является так называемым *«начальным состоянием»* (в нашем случае – «выкл.»), в котором система находится изначально. На рисунке оно отмечено словом *Начало* и стрелкой, указывающей на это состояние.

Часто необходимо выделять одно или несколько *«заключительных»* или *«допускающих»* состояний. Если в результате реализации некоторой последовательности входных воздействий автомат попадает в одно из них, то такую последовательность можно считать в определенном смысле «хорошей». Например, состояние «вкл.» (рис.1.4) можно рассматривать как допускающее, поскольку если переключатель находится в этом состоянии, то устройство, управляемое им, находится в рабочем режиме. Этот признак будет использован далее при рассмотрении КА- распознавателей, определяющих принадлежность заданной входной последовательности (цепочки) заданному языку (см. п.1.5).

Различают две основных модели конечных автоматов – *автоматы Мили* и *Мура.* Автоматы Мура отличаются тем, что выходной сигнал является функцией только текущего состояния, т.е. ω=λ(q). Модель КА (рис.1.3), имеющую один вход и один выход, называют еще *абстрактным автоматом* (АА), поскольку в ней абстрагируются от реальных физических входных и выходных сигналов, рассматривая их просто как буквы некоторого алфавита и в связи с идеализированным дискретным временем.

Преобразователи дискретной информации, математической моделью которых является АА, называют *цифровыми автоматами* (ЦА). Автоматы с числом внутренних состояний более одного | Q | >1 составляют *класс автоматов с памятью*. Далее, если не оговорено противное, будем рассматривать автоматы общего вида – автоматы Мили. Частным случаем таких автоматов является устройство, в котором значение выходного символа ωq не зависит от состояния и определяется текущим входным символом zf. Все состояния в нем эквивалентны, и можно считать, что такой автомат имеет одно состояние и по сути понятие *состояния* оказывается *лишним.* Функция переходов δ в нем вырождена. Поведение его однозначно задается функцией выходов.

Математической моделью устройства является функция ω(t)=λ(z(t)). Такие автоматы называются *автоматами без памят*и или *комбинационными схемами* (КС).

## *1.2. Формальное описание КА*

С математической точки зрения конечный автомат – это шестерка объектов:

А=(Z, W, Q, δ, λ, q 0 ),

где Z= {z1, z2, …,zn} – множество входных символов (входной алфавит);

W= {w1, w2,…,wG} – множество выходных символов (выходной алфавит);

Q= {q1, q2,…,qR} – множество состояний автомата (алфавит состояний);

δ: Q×Z→Q – функция переходов;

λ: Q×Z→Z – функция выходов;

q0∈Q – начальное состояние автомата.

Задание начального состояния q(t=0) = q0 имеет принципиальное значение, ибо, не зная его, невозможно определить реакцию автомата на любое входное слово. Функции переходов и выходов зависят от двух аргументов, поэтому, не зная q0, нельзя найти q(1), q(2), …, а также w(1), w(2),…**.** Автомат с выделенным начальным состоянием (обычно это q0) называется *инициальным*.

Функционирование КА описывается системой:



, (1)

т.е. указываются закон изменения состояния к следующему тактовому моменту (t+1) в зависимости от входного символа z(t) и предыстории q(t), а также значение выходного символа wq (t) в текущий тактовый момент как функции λ состояния q(t) и входного символа z(t) в тот же момент времени. Автомат, для которого функции δ и λ определены на всех парах (qm ,zf), называют *полностью определенным* или *полным*, а тот, для которого функция δ или λ или обе функции определены не на всех парах, – не полностью определенным или *частичным*.

КС на абстрактном уровне описывается тройкой объектов:

A=(Z, W, λ),

где Z и W – входной и выходной алфавиты соответственно, а

λ - функция выходов.

Резюмируя изложенное, можно сказать, что***цифровым автоматом*** *называется преобразователь дискретной информации, способный принимать различные состояния, переходящий из одного состояния в другое под воздействием входных сигналов и вырабатывающий выходные сигналы.*

Реализация КА приведена на рис. 1.5.



Рис. 1.5. Реализация КА

Логический блок ЛБ вырабатывает выходные сигналы W и сигналы U, позволяющие изменить текущее состояние автомата. Блок памяти БП автомата хранит информацию о текущем состоянии Q, которое вместе с входным сигналом Z определяет выходную реакцию автомата W и следующее состояние.

## *1.3. КА – модель цифровых автоматов*

И сложные дискретные процессы, например такие, как протокол передачи информации в компьютерных сетях, и сложные дискретные системы, такие как ЭВМ и их отдельные устройства (процессор, устройства ввода и др.) или пульт управления энергосистемой, и простые, такие как спецпроцессоры, устройство, регулирующее работу светофора, устройство управления лифтом, кодовый замок и другие, – можно считать автоматами.

*Конечный автомат* представляет собой хотя и абстрактную, но с функциональной точки зрения довольно точную *модель* дискретного процесса либо (дискретного) вычислительного или управляющего устройства и является удобным средством описания многих систем, взаимодействующих с окружением и реагирующих на поток внешних событий, так называемых реактивных систем, составляющих многие компоненты аппаратного и программного обеспечения. Такая модель обладает наглядностью и выразительностью, ясностью семантики (трактовки элементов модели) и в то же время достаточно строга и формальна. Ее используют при решении самых разнообразных задач, связанных с информационными процессами, таких как:

* *построение систем ПО*, включая лексические анализаторы компиляторов (компонент компилятора, который отвечает за разбивку исходного текста на такие лексические единицы, как идентификаторы, ключевые слова и знаки пунктуации), и системы проверки корректности схем или протоколов связи и др., которые могут находиться в конечном числе различных состояний;
* *проектирование систем технической диагностики*;
* *проектирование узлов и блоков ЭВМ и других вычислительных систем и устройств*;
* *проектирование устройств промышленной автоматики и др*.

Однако интерпретация автомата как устройства является важной, но не универсальной, поскольку отражает лишь один из аспектов его работы – преобразование множества входных слов Z\* в множество выходных W\*, т.е. реализацию автоматного отображения оператора. Такой подход является источником многих задач, рассматриваемых при проектировании *автоматов-преобразователей.*

Вместе с тем на абстрактном уровне, т.е. не исследующем структуру, более существенной является работа со словами при наличии конечной памяти, что приводит к более общему истолкованию автоматов как алгоритмов с конечной памятью, роль которой выполняют состояния, и рассмотрению их с алгоритмической точки зрения.

## *1.4. КА-распознаватели*

При рассмотрении автоматов с алгоритмической точки зрения возможности автоматов изучают в терминах множеств слов, с которыми они работают, интересуются тем, какие множества конечных последовательностей входных сигналов Z\* можно отличить друг от друга с помощью выходных сигналов, иначе – принадлежит ли поданное на вход слово заданному множеству, т.е. языку.

С точки зрения теории алгоритмов *КА* можно рассматривать как *некоторый алгоритм*, принимающий в качестве *входа* символ за символом произвольную цепочку ω над словарем Z (*словарем* будем называть конечное множество элементов, элементы словаря – *символами*, последовательности символов словаря *– цепочками* или предложениями, а их множество – *языком*. Язык над словарем Z будем обозначать как LZ или просто L). *Выходом* его будет один из двух возможных ответов: «да», т.е. ω∈L, либо «нет», т.е. ω∉L .

Любой конечный механизм задания языка L называется грамматикой. Существуют два типа грамматик – *порождающие* и *распознающие.*

Под первой понимается конечный набор правил, позволяющий строить все правильные предложения языка L. *Распознающая* грамматика задает критерий принадлежности произвольной цепочки ω данному языку.

Таким образом, КА можно рассматривать как устройство, выполняющее алгоритмические операции над языками. Простейшей из них является *распознавание* – выделение автоматом всех входных предложений, принадлежащих языку.

*Метод распознавания* использует *частичный алгоритм*, который для произвольной входной цепочки x остановится и ответит «да» после конечного числа шагов, если эта цепочка принадлежит языку. Такой алгоритм определяет язык L как множество входных цепочек, для которых он выдает «да».

Частичные алгоритмы, определяющие языки, представляются в виде несколько схематизированного устройства, называемого *распознавателем*, общий вид которого приведен на рис. 1.6.

Таким образом, распознаватель – это схематизированный алгоритм, распознающий принадлежность объекта заданному классу. Иначе – алгоритм отвечает на вопрос: «Истинно ли высказывание х∈M?» и выдает ответ «да» или «нет». Такими распознавателями являются конечные автоматы (КА), автоматы с магазинной памятью (АМП), машины Тьюринга (МТ) и др.

Вспомогательная память может быть в виде потенциально бесконечной ленты, как в МТ, либо магазинного типа, как в АМП, либо вообще отсутствовать, как в КА.



Рис. 1.6. Общий вид распознавателя

## *1.5. Формальное описание КА-распознавателя*

С математической точки зрения КА-распознаватель – это пятерка объектов:

А=(Q, Z, q0, δ, F ),

где Q – конечное непустое множество состояний (алфавит состояний);

Z–входной алфавит (конечное множество допустимых входных сигналов);

q0∈Q – начальное состояние;

δ:QxZ→Q – функция переходов;

F⊆Q – множество заключительных (допускающих) состояний.

Он может быть задан *таблицей переходов*, в которой на пересечении строки, соответствующей символу zj, и столбца, соответствующего состоянию qi, ставится новое состояние q1=δ(zj,qi), либо графом, вершины которого отмечены состояниями, а дуги соответствуют переходам (qi,qi1), если существует такой символ z∈Z, что

q1=δ (q,z) в случае детерминированного КА и

q1∈δ (q,z) – в случае недетерминированного.

Начальное состояние q0 отмечается направленной в него стрелкой со словом «начало», а заключительное – двойным кружком. КА может быть описан и *аналитически* как процесс порождаемых автоматом А конфигураций для заданной входной цепочки α∈Z.

*«Работа»* КА представляет собой некоторую последовательность шагов (тактов), каждый из которых определяется текущим состоянием qi и входным символом zj, обозреваемым СГ в данный момент времени. Сам *шаг* есть считывание символа, изменение состояния КА и сдвиг СГ на одну ячейку вправо. Чтобы определить будущее поведение КА, надо знать лишь текущее состояние qi и цепочку символов α на входной ленте, состоящую из символа под СГ и всех символов, расположенных вправо от нее. Эти два элемента информации δ(q, α)∈QxZ дают мгновенное описание КА и называются *конфигурацией* автомата А.

Конфигурация (q0, α) называется *начальной*, а пара (q,e), где q∈F, – *заключительной* (или допускающей) конфигурацией. Такт автомата А представляется бинарным отношением ├А, определенным на конфигурациях. Если δ(q, α) содержит q′, то (q, zα) ├А (q′,α) для всех α∈Z. Это говорит о том, что если автомат А находится в состоянии q и СГ обозревает входной символ z, то он может делать такт, за который переходит в состояние q′ и сдвигает СГ.

Различают *детерминированные* (ДКА) и *недетерминированные* конечные автоматы (НКА). Термин «детерминированный» говорит о том, что для всякой последовательности входных символов существует лишь одно состояние, в которое автомат может перейти из текущего. «Язык» ДКА – это множество всех допустимых им цепочек. Вопрос о «допуске» последовательности входных символов ДКА решают следующим образом. Говорят, что КА–распознаватель А=(Q, Z, q0, δ, F) допускает цепочку, если (q0,α)├A\* (q, e) для некоторого q∈F, т.е. если α переводит его из начального в одно из заключительных состояний δ\*(q0, α)∈F. Множество всех цепочек, допускаемых автоматом А, образуют язык, допускаемый A.

Очевидно LA={α∈Z\*|δ\*(q0,α)∈F }.

В противоположность детерминированному «*недетерминированный*» КА обладает свойством находиться сразу в нескольких состояниях одновременно. Эту особенность часто представляют как свойство автомата делать «догадки» относительно его входных данных. Всякий такой автомат допускает язык, допустимый некоторым ДКА, т.е. НКА допускает регулярные языки точно так же, как и ДКА. Различие между ними состоит в типе значений функции δ. В НКА ее значение – некоторое подмножество множества Q, т.е. для НКА – это множество состояний, а для ДКА – одиночное состояние. Будем считать, что НКА допускает цепочку ω, если в процессе чтения ее символов можно выбрать хотя бы одну последовательность переходов в следующие состояния так, чтобы прийти из начального состояния в одно из допускающих. Формально, если А=(Q, Z, q0, δ, F) – некоторый НКА, то LA={ω|δ\*(q0, ω)∩ F=∅}. Таким образом, L(А) есть множество цепочек ω из Z\*, для которых среди состояний δ\*(q0,ω) есть хотя бы одно допускающее.

**Пример.** Построить НКА, допускающий только те цепочки из 0 и 1, которые оканчиваются на 01.

Для решения вопроса о допустимости заданной цепочки рассмотрим ситуации, которые должен помнить автомат относительно прочитанных входных символов Z={0,1}.

Начальным является состояние q0, в котором автомат будет находиться ( а также, возможно, и в других состояниях) до тех пор, пока не «догадается», что на входе началась замыкающая подцепочка 01.

Вероятность того, что следующий символ не является начальным для замыкающей подцепочки 01, даже если это символ 0, всегда существует, поэтому состояние q0 может иметь переходы в себя как по 1, так и по 0. (рис. 1.7).



Рис. 1.7. Граф НКА, допускающего цепочки, оканчивающиеся на 01

Вместе с тем если очередной входной символ – 0, то НКА может предположить, что уже началась замыкающая последовательность 01, поэтому дуга с меткой 0 ведет из состояния q0 в q1 . Заметим, что из q0 выходят две дуги, отмеченные символом 0, т.е. автомат может перейти как в q0, так и в q1 и в действительности переходит в оба эти состояния. В состоянии q1 автомат проверяет, является ли следующий входной символ единицей. Если да, то он переходит в состояние q2 и считает эту цепочку допустимой.

Заметим, что ДКА в каждом состоянии имеет ровно одну выходящую дугу для каждого входного символа, а для НКА такого ограничения нет.

Далее рассмотрим, как такой НКА обрабатывает цепочку ω=00101(рис.1.8).

Таблица 1.1



Рис. 1.8. Процесс обработки цепочки ω

Прочитав последовательность 001, НКА находится в состояниях q0 и q2 и допускает последовательность, поскольку q2 -допускающее состояние. Однако чтение входных символов не закончено. Четвертый входной символ 0 приводит к тому, что ветвь, соответствующая q2, отмирает (нет выхода по 0), в то время как q0 переходит в q0 и q1. По последнему символу происходит переход из q0 в q0, а из q1 - в q2. Поскольку автомат вновь переходит в допускающее состояние q2, цепочка ω=00101 допустима, т.е. ω∈LA. Такой НКА (рис. 1.7) можно задать формально как А=({q0, q1, q2},{0,1},δ,q0,{q2}), где функция переходов задается табл. 1.1.

## *1.6. Примеры КА-распознавателей*

**Пример 1.** Построить детерминированный КА, допускающий в алфавите Z={0,1} все цепочки нулей и единиц, содержащие подцепочку 00, например ω=01001. Язык LA={x00y⏐, x и y – некоторые цепочки, состоящие только из 0 и 1}. Тогда ω=1101101∉LА .

Чтобы построить автомат, необходимо задать все компоненты вектора А=(Z, Q, q0, δ, F). Изначально известно лишь, что алфавитом входных символов является Z={0,1} и что имеется множество состояний этого автомата. Чтобы решить, содержит ли входная последовательность подцепочку 00, автомат А должен помнить следующие факты относительно прочитанных им входных данных.

1. Два нуля («00») еще не появились. На выходе либо ничего не подавалось (начальное состояние), либо на предыдущем шаге символ не был нулем.

2. Два нуля («00») не появились, но предыдущий символ был нулем.

3. Два нуля («00») появились, и тогда всякая считанная далее последовательность допустима, т.е. с этого момента автомат будет находиться лишь в допускающих состояниях.

Каждое из этих условий можно представить как некоторое состояние. Условию 1 соответствует начальное состояние q0 . Конечно, находясь в начале процесса, нужно последовательно прочитать 0 и 1. Но если в состоянии q0 читается 1, то это не приближает нас к ситуации, когда прочитана последовательность «00», поэтому нужно оставаться в состоянии q0, т.е. δ(q0, 1) = q0. Однако если в состоянии q0 читается 0, то мы попадаем в условие 2, т.е. два нуля еще не прочитаны, но уже прочитан «0», и обозначим эту ситуацию через q1, q1=δ(q0, 0).

Если в состоянии q1 читается 0, понятно, что во входной последовательности идет второй 0 подряд, и можно перейти в допускающее состояние (q2 ), которому соответствует условие 3. В случае если в q1 читается 1, то необходимо вернуться в начальное состояние q0 , т.к. ясно, что двух нулей подряд не получится. Следовательно, δ(q1, 0) = q2,δ(q1, 1) = q0 .

Таким образом, Q={q0, q1, q2}, где q0 – начальное состояние, а q2 – единственное допускающее состояние автомата, т.е. F={q2}. Полное описание автомата, допускающего язык L (цепочек, содержащих «00» в качестве подцепочки), имеет вид: А=({q0, q1, q2},{0,1},δ,q0,{q2}), где δ задается табл. 1.2.

Таблица 1.2



Рис. 1.9. ДКА, допускающий

подцепочку 00

Далее проверим, допускает ли построенный ДКА цепочку ω=01001. В п. 1.5 отмечалось, что ДКА допускает цепочку ω∈Ζ, если δ(q0,ω)∈F, т.е. если последний символ цепочки переводит автомат в заключительное состояние.

Процесс порождения конфигураций для заданной цепочки будет следующим:

(q0 , 01001)├ (q1 , 1001) ├ (q0 , 001) ├ (q1 ,01) ├ (q2 , 1) ├ (q2 ,е).

Поскольку ├А\* ( ), то ω=01001∈L(A).



**Пример 2.** Построить НКА, допускающий цепочки в алфавите Z={z1, z2, z3}, у которых последний символ цепочки уже появлялся в ней раньше, например ω=z1z2z3z2z1.

Определим компоненты вектора А={Z,Q,q0,δ,F} и в частности множество Q={q0, q1, q2, q3, q4}, где:

q0 – начальное состояние автомата, в котором автомат не пытается ничего распознать, он или какой-то его экземпляр находится в так называемом нейтральном состоянии;

q1, q2, q3 – состояния, в которых автомат делает предположение о том, что последний символ цепочки совпадает с индексом состояния;

q4 ∈F - заключительное состояние.

Формально автомат описывается пятеркой А=({q0, q1, q2, q3, q4},{z1, z2, z3}, δ, q0, {q4}), где δ задается табл. 1.3. Граф автомата представлен на рис.1.10

Таблица 1.3



Рис. 1.10. Граф автомата

Далее проверим принадлежность языку L(А) цепочек ω1=z2z1z1z3z2; ω2=z3z1z1z3z2. Процессы порождения конфигураций для ω1 иω2 приведены на рис.1.11 и 1.12 соответственно.



Рис.1.11.Порождение конфигураций для ω1

Так как ├\* (q4,e), то .



Рис. 1.12. Порождение конфигураций для ω2

Поскольку под действием последнего символа цепочки ω2 автомат не переходит в заключительное состояние, ω2 ∉*L(S).*

## *1.7. Задачи*

1. Построить ДКА, допускающий в алфавите Z{0,1} все цепочки нулей и единиц, содержащие в себе:

а) подцепочки 01, например 01, 11010, 1000111;

б) подцепочку 11;

в) подцепочку 100;

г) множество цепочек, которые начинаются или оканчиваются (или и то и другое) последовательностью 01;

д) множество всех цепочек, в которых всякая подцепочка из пяти последовательных символов содержит хотя бы два 0.

2. Определить, допускает ли автомат, заданный графом (рис. 1.13), цепочки ω1=z1 z2 z2 z3 z1 z3; ω2=z2 z1 z3 z3 z1 z2; ω3=z2 z3; ω4=z1 z3, если q0∈F.

3. Построить КА, допускающий все цепочки, в которых за каждой единицей непосредственно следует «0».

4. Построить КА с входным алфавитом V={a, b, c}, распознающий:

а) все цепочки, в которых две последние буквы не совпадают;

б) все цепочки, начинающиеся и заканчивающиеся различными символами;

в) все цепочки, у которых последний символ цепочки уже появлялся раньше.

5. Построить ДКА, допускающий в алфавите Z{0,1} все цепочки, содержащие четное число 0 и четное число 1.

6. Преобразовать НКА (рис.1.7) в ДКА.

7. Построить НКА:



а) с входным алфавитом V={a,b,c,d},распознающий множества цепочек: abc, abd, aacd;

б) с входным алфавитом V={a,b,c},распознающий множества цепочек: ab, bc, ca;

в) с входным алфавитом Z={0,1}, распознающий множества цепочек: 1010, 101, 011.

Рис. 1.13. Граф автомата

## *1.8. КА-преобразователи*

### **1.8.1. Классификация автоматов**

По виду деятельности, весьма разнообразной, автоматы можно сгруппировать в три основных класса:

* информационные,
* управляющие,
* вычислительные.

Примерами *информационных* автоматов являются справочные автоматы на вокзалах, светофоры, устройства аварийной сигнализации; *управляющих* – кодовый замок, устройства управления лифтом, шлагбаумом железнодорожного переезда, различными типами станков*; вычислительных* – процессор ЭВМ, АЛУ, спецпроцессоры либо простое вычислительное устройство, выполняющее одну или несколько арифметических и логических операций, таких как A←A+B+C (сложение с переносом), A←A-C (вычитание заема), A←A&B (конъюнкция А и В). В сложных системах-автоматах, таких как ЭВМ, пульт управления энергосистемой и др., выполняются все три указанных вида деятельности.

### **1.8.2. Абстрактный синтез автоматов**

Абстрактной модели КА соответствует этап абстрактного синтеза, который заключается в переходе от первоначальной записи условий работы проектируемого ЦА к стандартному заданию его автоматного оператора. К стандартным формам относятся в том числе и таблицы переходов-выходов, и графы. В свою очередь этап абстрактного синтеза включает два подэтапа:

- алгоритмический синтез;

- минимизация памяти.

На первом получают таблицы переходов-выходов автомата, который может содержать лишние внутренние состояния, что отражается на объеме устройства. Функция состояний заключается в том, чтобы участвовать в определении зависимости между входами и выходами автомата, и физической природой состояний при этом не интересуются. Следовательно, любое множество состояний, выполняющих эту функцию, является приемлемым, независимо от того, выражают они какой-либо интуитивный смысл или нет. Эта свобода выбора множества состояний весьма выгодна, поскольку позволяет заменить одно множество другим и, в частности, найти оптимальное или минимальное множество состояний. В основе минимизации множества состояний лежит понятие эквивалентности и гомоморфизма автоматов.

При использовании конечно-автоматной модели *в реальных задачах элементам формальной системы придаются конкретные свойства и признаки*.

### **1.8.3. Примеры КА–преобразователей и их синтез**

**Пример 1.** Построить КА, управляющий шлагбаумом (ш/б) железнодорожного (ж/д) переезда.

*1. Первоначальное описание условий работы ЦА*

Схема движения поездов и автотранспорта через железнодорожный переезд, регулируемый шлагбаумом, приведена на рис.1.14.



Рис. 1.14. Регулируемый железнодорожный переезд

Для разработки алгоритма работы ЦА рассмотрим ситуации.

1. Если первая колесная пара ведущего вагона поезда миновала пункт А на ж/д полотне, то ш/б будет закрыт.
2. Если последняя колесная пара последнего вагона прошла пункт В при движении поезда слева, то ш/б будет открыт.
3. Если первая колесная пара ведущего вагона поезда при движении справа миновала пункт С на ж/д полотне, то ш/б будет закрыт.
4. Если последняя колесная пара последнего вагона прошла пункт D при движении поезда справа, ш/б будет закрыт.
5. Условие 2 должно быть выполнено, если нет поезда справа или он еще не достиг пункта С.
6. Условие 4 должно быть выполнено, если нет поезда слева или он еще не достиг пункта А.

*2. Стандартное описание автоматного оператора ЦА*

Для формального описания модели проектируемого ЦА необходимо задать все компоненты вектора S={Z, W, Q, δ, λ, q0,}. Будем считать, что символы А, В, С, Д – идентификаторы сигналов о пересечении поездами соответствующих пунктов ж/д полотна и являются символами входного алфавита Z, т.е. Z={A, B, C, D}. На основе анализа ситуаций работы автомата введем понятие «опасной зоны» (заштрихована), которая свидетельствует о том, что переезд закрыт, если хотя бы один из поездов пересек пункт А или С. Такая условность позволяет сократить число возможных ситуаций, каждая из которых будет отмечена состоянием: q0 – нет ни одного поезда в ОЗ; q1 – наличие одного из поездов в ОЗ; q2 – два поезда в ОЗ. Таким образом, Q={q0, q1, q2}. Автомат должен формировать два выходных символа ω0 – открыть ш/б; ω1 – закрыть ш/б, т.е. W={ω0, ω1}. Формально автомат описывается шестеркой S=({q0, q1, q2}, {A, B, C, D}, {ω0, ω1}, δ, λ, q0), где функции δ и λ задаются совмещенной таблицей переходов и выходов (табл.1.4). Граф, описывающий работу автомата, приведен на рис. 1.15.

Таблица 1.4



Рис.1.15. Модель устройства, управляющего ш/б ж/д переезда

**Пример 2.** Построить конечно-автоматную модель 2-разрядного двоичного реверсивного счетчика (рис. 1.16) , способного работать как в режиме сложения, так и в режиме вычитания. Для формального описания работы КА необходимо задать все компоненты вектора А={Z, Q, W, δ, λ, q0,}, придав им конкретные свойства и признаки.

Таблица 1.5



Рис. 1.16. Общий вид РС

**Анализ условий работы РС**

Число состояний автомата равно четырем и определяется самим заданием: РС содержит два элемента памяти – триггеры Т1,Т2, состояния которых Q1, Q2 и будут определять состояния автомата в целом Q={q0, q1, q2, q3}(см.табл. 1.5).

Для определения алфавитов Z и W проанализируем работу РС при различных комбинациях единичных входных импульсов.

При поступлении единичных импульсов на суммирующий вход X1 РС работает как суммирующий двоичный счетчик с модулем М=4, изменяя при этом свои состояния последовательно от q0 до q3. Если в состоянии q3 на автомат поступает очередной импульс по каналу X1, то счетчик обнуляется (состояние q0) и начинает считать заново, а на выходе по каналу Y1 выдается сигнал переполнения ω1, что соответствует комбинации на выходах: у1=1, у2=0. При этом комбинация на входных каналах (х1=1, х2=0) может повторяться сколь угодно долго. Однако при смене состояний внутри каждого цикла, т.е. q0→ q1, q1→ q2, q2→ q3, на выходе автомата будет формироваться другой абстрактный сигнал - ω2, соответствующий комбинации (y1=0, y2=0).

При поступлении единичных импульсов на вычитающий вход x2 РС работает как вычитающий двоичный счетчик с модулем М=4. Из состояния q0 он переходит в состояние q3, а затем последовательно из q3  в q0 (рис. 1.17). При смене состояния с q0 на q3 на выходе формируется сигнал заема (ω3), что соответствует комбинации y1=0, y2=1. При смене состояний внутри цикла вычитания q3→ q2, q2→ q1, q1→ q0 на выходе автомата формируется абстрактный сигнал ω2 .Таким образом, входной алфавит Z={z1, z2, z3}, W={ω1, ω2, ω3,}, где каждому абстрактному символу соответствуют комбинации сигналов на входе и выходе РС, приведенные в табл. 1.6, 1.7.

Таблица 1.6 Таблица 1.7



Очевидно, что комбинации (х1=1, х2=1) и (Y1=1, Y2=1) не имеют смысла, т.к. противоречат логике работы автомата.

**Стандартное описание автоматного оператора РС**

Формально оператор описывается шестеркой А=({q0, q1, q2, q3},{z1,z2,z3}, {ω1, ω2, ω3,}, δ, λ,q0,), где функции δ и λ задаются совмещенной таблицей переходов и выходов (табл. 1.8). Граф работы КА приведен на рис. 1.17.



Таблица 1.8



Рис. 1.17. Граф работы РС

**Пример 3.** Построить КА, вставляющий «0» в двоичную последовательность после четырех единиц подряд.

**Анализ условий работы автомата**

Входной алфавит Z={0,1} определен здесь заданием. W={ω0, ω1}, где ω0 –вставить «0», ω1 – ничего не менять, повторить входной символ.

Содержательную сторону состояний определим следующим образом:

q0 – ожидание двоичного символа;

q1 – получена первая единица;

q2 – получена вторая *подряд* единица;

q3 – получена третья *подряд* единица;

q4 – получена четвертая *подряд* единица.

При определении терминальных значений состояний q1…q4 существенным и определяющим является слово «*подряд*». Если каждому из этих состояний присвоить значение «получена вторая, третья, четвертая единицы», то это будет уже другой КА, вставляющий «0» не после 4-х *подряд* идущих «1», а после каждых 4-х последовательно поступивших «1» в двоичной последовательности, например:

ω=010110101101101…

**Стандартное описание автоматного оператора ЦА**

Формально заданный автомат описывается шестеркой А=({q0, q1, q2, q3, q4},{0,1},{ω0, ω1}, δ, λ, q0), где δ и λ задаются табл. 1.9. Граф работы КА приведен на рис. 1.18.

Таблица 1.9



Рис.1.18. Граф автомата Мили, вставляющего «0» после 4-х подряд «1»

### **1.8.4. Задачи**

1. Построить конечный автомат, выбрасывающий лишние пробелы в тексте.
2. Построить КА, добавляющий бит нечетности к цепочке из «0» и «1».
3. Построить модель кодового замка с пятью кнопками (А, Б, В, Г, Д), открывающегося при наборе кода В\*Д и остающегося открытым, пока нажата кнопка Д. Символ \*∈Z означает, что ни одна кнопка не нажата, символы А, Б, В, Г, Д∈Z соответствуют нажатой кнопке. Множество W={ω0, ω1}, где ω0 – замок открыт, ω1 – замок закрыт.
4. Построить КА, убирающий подчеркнутые нули в потоке битов:



и распознающий начальный и заключительный флажки.

1. Построить КА, продающий пиво и выдающий сдачу. Автомат может принимать монеты достоинством 5 и 10 рублей, а кружка пива стоит 15 рублей. Кроме отверстий для приема монет и выдачи сдачи у автомата есть кнопки «Наливай» и «Сброс».
2. Построить КА для продажи билетов на пригородный поезд. Автомат может принимать монеты достоинством 5 копеек, 10 копеек, 50 копеек, 1 рубль. Цена билета 2 рубля 15 копеек. Таким образом, Z={5 к., 10 к., 50 к., 1 р., \*}, где \* - отсутствуют монеты. W={0, 1, С}, где 1 – дать билет, 0 – не давать билет, С – сброс. Сигнал «1» может быть выдан только при точном наборе стоимости билета.
3. Построить конечно-автоматную модель вычитающего двоичного счетчика последовательного счета с модулем М=5, обладающего свойством самовосстановления после сбоя (в качестве элементов памяти использовать ТС-триггеры).
4. Построить конечно-автоматную модель суммирующего двоичного счетчика последовательного счета с модулем М=5, обладающего свойством самовосстановления после сбоя (в качестве элементов памяти использовать ТС-триггеры).
5. Построить конечный автомат для подсчета числа слов, начинающихся с ОС и заканчивающихся на А, таких как «остановка», «осциллограмма», «острога» и др., в русском тексте, составленном из 33 букв алфавита и пропусков.
6. Построить КА, управляющий движением лифта 6-этажного жилого дома. Кнопка номера этажа соответствует сигналу вызова лифта на соответствующий этаж. Движение лифта должно быть заблокировано при перегрузке (больше 6 человек). Автомат должен вырабатывать четыре управляющих сигнала: вверх, вниз, останов, блокировка. Входной сигнал принимается только стоящим лифтом.
7. Построить КА, управляющий движением лифта 6-этажного жилого дома, отличающийся от КА (задача 10) возможностью «подбирать» пассажиров, попутно следующих а) вниз; б) вверх, т.е. входной сигнал принимается не только стоящим, но и движущимся лифтом.
8. Построить КА, вставляющий дополнительный нуль в двоичную последовательность после каждых пяти подряд идущих единиц. Например, двоичную последовательность 001111110010011111110 автомат должен преобразовать в последовательность 00111110100100111110110 (подчеркнуты автоматически вставленные биты).
9. Построить КА, управляющий светофором автоматического регулирования транспорта на перекрестке (рис.1.19). Каждый из светофоров может показывать один из сигналов: К, Ж, З, стрелка, мигающий зеленый, мигающий желтый (ночной). Управление светофорами должно быть согласовано так, чтобы не создавалась аварийная ситуация.



Рис. 1.19. Регулируемый перекресток

1. Построить конечный автомат, управляющий движением транспорта на перекрестке главной и второстепенной улиц. Для каждой из них показывается один из сигналов: К, Ж, З и мигающий зеленый. Естественно, что автомат должен управлять светофором так, чтобы не создавалась аварийная ситуация (например, не должно быть зеленого сигнала на главной и второстепенной улицах одновременно). С 23 часов до 6 часов утра светофор переводится в режим мигания.
2. Построить КА, обеспечивающий пешеходам возможность пересечь магистраль, движение по которой регулируется автоматическим 3-секционным светофором (К, Ж, З). Движение транспорта по магистрали должно прекращаться по требованию пешехода нажатием кнопки. Разрешение пешеходу должно быть и визуальным, и звуковым. Возврат автомата в режим авторегулирования должен быть автоматическим.

# Содержание отчета

Отчет должен содержать результаты предварительной подготовки к работе, результаты вычислений в виде таблиц, графиков, с соответствующими заголовками, исходными данными и пояснениями, краткие выводы и оценку результатов.

# Список источников

## Основная литература

1. Теория автоматов/ Ю.Г. Карпов СПб.: Питер, 2002. 224 с.
2. Джон Хопкрофт, Раджив Мотвани, Джеффри Ульман. Введение в теорию автоматов, языков, вычислений, 2-е изд.: Пер. с англ. М.: Издательский дом «Вильямс», 2002. 528 с.
3. Кузнецов О.П., Адельсон-Вельский Г.М. Дискретная математика для инженера. М.: Энергоатомиздат,1988. 480 c.
4. Айзерман М.А., Гусев Л.А. и др. Логика, автоматы, алгоритмы. М.: Физматгиз, 1963. 556с.
5. Трахтенброт Б.А. Алгоритмы и вычислительные автоматы. М.: Сов.радио, 1974.

6. Конспект лекций

## Интернет-ресурсы

1. <http://ofap.ulstu.ru/vt/Theory_of_automats/content.htm>
2. [http://theory-a.ru](http://theory-a.ru/)

# ПРИЛОЖЕНИЕ А Пример оформления титульного листа для отчета по практической работе

ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ КЕМЕРОВСКОЙ ОБЛАСТИ

**ГБОУ СПО ЮРГИНСКИЙ ТЕХНОЛОГИЧЕСКИЙ КОЛЛЕДЖ**

Отделение ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ И РЕМОНТ РЭТ

**Решение задач по теории конечных автоматов**

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 2(3)

|  |  |
| --- | --- |
| МДК 01.02 | Математический аппарат для построения компьютерных сетей |
| Специальность | 230111 Компьютерные сети  Студент гр.    Преподаватель  Оценка « » М.В.Поликарпочкин |
| 2013 | |