

## Оглавление

Аннотация.....	2
Список сокращений.....	3
ГЛАВА 1. МОДЕЛИ ТЕОРИИ АВТОМАТОВ.....	4
1.1. Задачи теории автоматов. Виды автоматов.....	4
1.2. Общая схема и базовые модели конечного автомата.....	6
1.3. Абстрактный синтез конечного автомата .....	9
1.4. Переход от одной модели к другой: обоснование возможности и практика .....	15
Контрольные вопросы.....	20
Задания для самопроверки.....	20
ГЛАВА 2. КЛАССЫ АВТОМАТОВ.....	22
2.1. Мощность множества конечных автоматов.....	22
2.2. Класс явно-минимальных автоматов.....	23
2.3. Класс явно-сократимых автоматов .....	23
2.4. Изоморфные автоматы .....	26
Контрольные вопросы.....	28
Задания для самопроверки.....	28
ГЛАВА 3. МИНИМАЛЬНЫЕ АВТОМАТЫ .....	30
3.1. Эквивалентные состояния автомата и их свойства .....	30
3.2. Минимальная форма автомата .....	33
Контрольные вопросы.....	38
Задания для самопроверки.....	38
ГЛАВА 4. СТРУКТУРНЫЙ СИНТЕЗ КОНЕЧНОГО АВТОМАТА.....	40
4.1. Элементарные автоматы .....	40
4.2. Алгоритм структурного синтеза .....	45
4.3. Тестирование автомата.....	55
4.4. Функциональная полнота системы конечных автоматов .....	59
Контрольные вопросы.....	60
Задания для самопроверки.....	61
Литература.....	62

## Аннотация

Теория автоматов как научная дисциплина возникла в пределах теории управляющих систем (теоретической кибернетики) в середине XX века, в период начавшегося бурного развития средств электронной вычислительной техники и соответствующих областей математического знания. Потребовалась разработка теоретической базы для решения проблем, возникавших при проектировании реальных цифровых ЭВМ, а также в процессе построения и исследования гипотетических систем, таких как нейронные сети. В качестве моделей последних рассматривались конечные автоматы.

Развитие информационных технологий вывело сферу приложения теории автоматов далеко за рамки моделирования аппаратных средств цифровой электроники, расширив ее до фундаментальных основ современной теоретической информатики. Сегодня абстракции и модели, разработанные в теории автоматов, востребованы такими научными дисциплинами как теория формальных грамматик, математическая лингвистика, теория логических моделей, математическая логика и формальные аксиоматические системы, теория кодирования, теория вычислительной сложности и другими. Наиболее тесно теория автоматов связана с теорией алгоритмов и, в частности, с таким ее разделом как теория абстрактных машин.

В настоящее время существует ряд изданий учебного и учебно-методического назначения, в названиях которых содержится словосочетание «теория автоматов». Нередко эти издания носят синтетический характер, объединяя под одной обложкой как элементы самой теории автоматов, так и фрагменты перечисленных выше научных областей, в которых аппарат данной теории находит мощное и вполне оправданное применение.

Цель настоящего учебного пособия – дать представление о ключевых положениях собственно теории автоматов как отдельной научной и учебной дисциплины и помочь в ее освоении. В главе 1 рассмотрены базовые модели теории, их построение и эквивалентность. Глава 2 посвящена основным классам автоматов. В главе 3 излагаются вопросы, относящиеся к минимальным автоматам. В главе 4 большое место отведено практике структурного синтеза автомата – одному из главных и не теряющих актуальности приложений теории. В конце каждой главы даны контрольные вопросы и задания для самопроверки.

Учебное пособие рассчитано на студентов, изучающих дисциплины «Теория автоматов» и «Прикладная теория цифровых автоматов» в соответствии с учебным планом подготовки бакалавров по направлению «Информатика и вычислительная техника».

### [Оглавление](#)

## Список сокращений

АА – абстрактный автомат

КА – конечный автомат

КС – комбинационная схема

ДНФ – дизъюнктивная нормальная форма

СДНФ – совершенная дизъюнктивная нормальная форма

ЭА – элементарный автомат

[Оглавление](#)

# ГЛАВА 1. МОДЕЛИ ТЕОРИИ АВТОМАТОВ

## 1.1. Задачи теории автоматов. Виды автоматов

Предметом теории автоматов является изучение математических моделей преобразователей дискретной информации<sup>\*)</sup>. В данной теории решаются следующие **основные задачи**: анализ и синтез автоматов, определение полноты, минимизация и эквивалентные преобразования автоматов. Дадим краткую формулировку каждой из перечисленных задач.

**Задача анализа.** По заданному автомату описать его поведение. Вариант постановки: по неполному описанию автомата установить некоторые его свойства.

**Задача синтеза.** Построить автомат с наперед заданным поведением (алгоритмом функционирования). Задачу синтеза принято рассматривать двояко: *абстрактный синтез* как построение математической модели автомата и *структурный синтез* как разработку функциональной логической схемы автомата.

**Задача полноты.** Пусть  $M$  – некоторое множество автоматов. Определить, обладает ли совокупность автоматов, составляющих подмножество  $M' \subseteq M$ , свойством полноты. Иными словами, если ко всем автоматам подмножества  $M'$  конечное число раз применить операцию суперпозиции, совпадут ли  $M'$  и  $M$ ?

**Задача минимизации.** Построить автомат, минимальный заданному. Минимальный автомат обладает наименьшим числом компонентов модели (в частности, минимальной мощностью множества так называемых *состояний*) и при этом функционально эквивалентен заданному автомату.

**Задача эквивалентных преобразований.** Определить полную систему правил, позволяющую преобразовывать произвольный автомат в любой эквивалентный ему автомат. Частным случаем данной задачи является переход от одной модели автомата к другой. Два автомата функционально эквивалентны, если их поведение одинаково при воздействии одних и тех же последовательностей входных сигналов. В таком случае говорят о совпадении *моделей поведения* двух автоматов.

В число основных понятий теории автоматов входят:

- абстрактный автомат (АА);

---

<sup>\*)</sup> Автомат – от греч. *ἄτομάτος*, буквально переводится как «самодельствующий».

– композиция автоматов.

Понятие АА позволяет рассматривать дискретные устройства с точки зрения алгоритмов их функционирования, то есть реализуемых последовательностей действий по преобразованию дискретной информации.

**Абстрактным автоматом** называют модель, описываемую пятиместным кортежем:

$$A = (X, Y, S, f_y, f_s),$$

где первые три компонента – непустые множества:

$X$  – множество входных сигналов АА,

$Y$  – множество выходных сигналов АА,

$S$  – множество состояний АА.

Два последних компонента кортежа – характеристические функции:

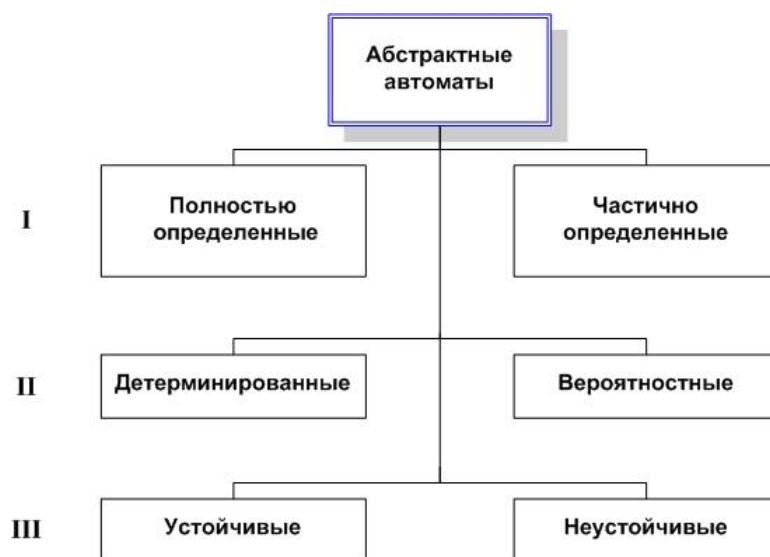
$f_y$  – функция выходов;

$f_s$  – функция переходов АА из одного состояния в другое.

Если множества  $X, Y, S$  – конечные, то такой АА называют **конечным автоматом (КА)**.

Если хотя бы одно из перечисленных множеств не является конечным, то такой АА называют **бесконечным**.

С целью классификации автоматов рассматривают ряд признаков, таких как определенность функции переходов и функции выходов, однозначность заданных функций, устойчивость состояний. Перечислим виды абстрактных автоматов, распределив их по трем классификационным признакам (рис. 1.1).



**Рис. 1.1.** Классификация абстрактных автоматов

[Оглавление](#)

### I. По определенности характеристических функций.

В автоматах *полностью определенных* областью определения функций  $f_s$  и  $f_y$  является множество всех пар  $(s_i, x_k) \in S \times X$ , где  $s_i \in S$ ,  $x_k \in X$ . В автоматах *частично определенных* либо обе характеристические функции, либо одна из них имеют областью определения строгое подмножество декартова произведения  $S \times X$ . Таким образом, характеристические функции подобных автоматов определены не для всех пар  $(s_i, x_k)$ .

### II. По однозначности функции переходов.

В *детерминированных* автоматах выполняется условие однозначности переходов: если АА находится в некотором состоянии  $s_i \in S$ , то под воздействием произвольного входного сигнала  $x_k \in X$  автомат может перейти в одно и только одно состояние  $s_j \in S$ , причем ситуация  $s_i = s_j$  вовсе не исключается. В автоматах *вероятностных* при воздействии одного и того же входного сигнала возможны переходы из состояния  $s_i$  в различные состояния из множества  $S$  с заданной вероятностью.

### III. По устойчивости состояний:

В *устойчивых* автоматах выполняется условие устойчивости: если автомат под воздействием входного сигнала  $x_k \in X$  оказался в состоянии  $s_i \in S$ , то выход из него и переход в иное состояние возможен только при поступлении на вход автомата другого сигнала  $x_z \in X$ ,  $x_z \neq x_k$ . Если условие устойчивости не выполняется хотя бы для одного состояния  $s_j \in S$ , то такой автомат называют *неустойчивым*.

Дальнейшее изложение будем вести, исходя из практических соображений, применительно к **полностью определенным, детерминированным и устойчивым конечным автоматам.**

## 1.2. Общая схема и базовые модели конечного автомата

Введем обозначения мощностей множеств, входящих в указанный выше кортеж:

$$X = \{x_1, x_2, \dots, x_p\}, \quad |X| = p;$$

$$Y = \{y_1, y_2, \dots, y_q\}, \quad |Y| = q;$$

$$S = \{s_1, s_2, \dots, s_n\}, \quad |S| = n.$$

### [Оглавление](#)

Конечный автомат, в описание которого входят таким образом определенные множества, называют  $(n, p, q)$ -автоматом, а самим множествам усваивают наименование *векторов*, например: вектор входных сигналов, вектор состояний.

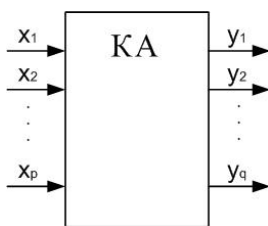
Все автоматы, и в том числе конечные, функционируют в *дискретном исчислении времени*. Моменты времени образуют ряд целых неотрицательных чисел:  $t = 0, 1, 2, 3, \dots$

В каждый дискретный момент времени КА находится в одном и только одном состоянии  $S_i$ , воспринимает одно значение вектора  $X$  и выдает на выходе одно значение вектора  $Y$ . Принято считать, что в момент времени  $t = 0$  автомат находится в начальном состоянии  $S_0$ , которое можно включить в кортеж отдельным, шестым компонентом:

$$A = (X, Y, S, f_y, f_s, S_0).$$

Автомат с выделенным начальным состоянием называют *инициальным*.

**Общую схему автомата** можно представить в виде некоторого «черного ящика», осуществляющего преобразование вектора входных сигналов в вектор выходных сигналов (рис. 1.2):



**Рис. 1.2.** Общая схема конечного автомата

Таким образом, можно записать:  $Y = f_y(X, t)$ .

Фактор времени в приведенном уравнении учитывается введением вектора состояний  $S$ , как своего рода «памяти о прошлом». Действительно, на один и тот же набор входных сигналов (значений компонентов вектора  $X$ ) автомат будет выдавать разные выходные сигналы (значения компонентов вектора  $Y$ ) в зависимости от состояния, в котором он находится в данный момент времени. Текущее состояние, в свою очередь, определяется алгоритмом функционирования автомата.

Рассмотрим **характеристические функции** автомата.

Функция  $f_s$  реализует бинарное отношение вида  $S \times X \rightarrow S$ , то есть каждой паре «состояние – входной сигнал» ставит в однозначное соответствие определенное состояние из множества  $S$ . Аналогично, бинарное отношение для функции  $f_y$  имеет вид  $S \times X \rightarrow Y$ , то есть каждой паре

## [Оглавление](#)

«состояние – входной сигнал» ставится в соответствие конкретный выходной сигнал – элемент множества  $Y$ .

Таким образом, характеристические функции определяют, в какое состояние  $s \in S$  перейдет автомат в следующий,  $(t+1)$ -й момент времени и каково будет значение выходного сигнала  $y \in Y$  в текущий момент времени  $t$ :

$$\begin{aligned} s(t+1) &= f_s(x(t), s(t)) \\ y(t) &= f_y(x(t), s(t)) \end{aligned} \quad \Bigg| \quad (1)$$

Из приведенных уравнений видно, что аргументами характеристических функций являются текущее значение входного сигнала и текущее состояние. Конечный автомат, заданный парой уравнений (1), называется **автоматом I рода** или, по имени автора модели, **автоматом Мили (Mealy)**.

На практике часто встречаются автоматы, выходные сигналы которых в момент времени  $t$  однозначно определяются текущим состоянием автомата и не зависят от компонентов вектора входных сигналов:

$$\begin{aligned} s'(t+1) &= f'_s(x(t+1), s'(t)) \\ y(t) &= f'_y(s'(t)) \end{aligned} \quad \Bigg| \quad (2)$$

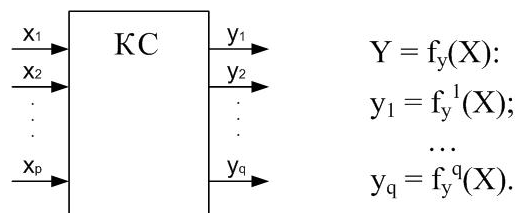
Автомат, заданный парой уравнений (2), называют **автоматом II рода** или **автоматом Мура (Moore)**. Штрих введен в обозначения для отличия записи функций и состояний автомата Мура от автомата Мили. Заметим, что автомат Мили по отношению к автомату Мура «запаздывает» на один дискретный момент времени по входному сигналу.

Автоматы I и II рода являются двумя **базовыми моделями**, изучаемыми теорией автоматов.

Если для выходного сигнала некоторого КА имеет место уравнение:

$$y(t) = f_y(x(t)),$$

то такой автомат называется *тривиальным*. Строго говоря, это уже не автомат, а комбинационная схема (КС), которая реализует совокупность булевых функций  $f_y^1, \dots, f_y^q$  для компонентов вектора выходных сигналов  $Y$  (рис. 1.3).



**Рис. 1.3.** Тривиальный автомат



На практике конечный автомат представляет собой совокупность двух устройств – операционного и управляющего автоматов. Операционный автомат выполняет ряд действий над входными данными и выдает результат, а управляющий автомат задает последовательность этих действий, то есть алгоритм функционирования операционного автомата. Например, в случае кодового замка операционным автоматом является электромагнит, управляющий засовом, а управляющим автоматом – электронная схема, обеспечивающая считывание и анализ сигналов от клавиш, проверку кода, выдачу сигнала операционному автомату на открытие замка, сброс в начальное состояние.

Другой пример – устройство умножения двоичных чисел с фиксированной запятой. Операционный автомат представляет собой ряд взаимосвязанных функциональных элементов – сумматора (например, дополнительного кода), регистров входных данных и результата, сдвигового регистра, цепи переноса двоичной единицы. Управляющий автомат задает порядок, в котором должны действовать составные узлы операционного автомата, чтобы обеспечить последовательность шагов реализуемого алгоритма умножения.

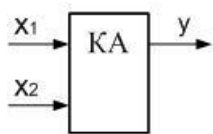
### 1.3. Абстрактный синтез конечного автомата

Задача абстрактного синтеза КА состоит в разработке математической модели на основании заданного алгоритма функционирования автомата. Так как модель представляет собой кортеж  $A = (X, Y, S, f_y, f_s)$ , то в процессе синтеза необходимо конкретизировать все его компоненты.

Множества  $X, Y, S$  могут быть заданы любым из способов, изучаемых в теории множеств. Поскольку множества конечны, вполне допустимо ограничиться простым перечислением их элементов. Это легко сделать, если мощности множеств невелики. В противном случае для задания того или иного множества можно, например, перечислить несколько первых его элементов и указать порождающую рекурсивную процедуру для получения остальных.

Характеристические функции наиболее удобно представимы *табличным* и *графическим* способами. Табличный состоит в построении таблицы переходов и выходов КА, графический способ – в построении взвешенного орграфа переходов автомата.

Для изучения процесса абстрактного синтеза КА рассмотрим **пример**.



Пусть требуется получить математическую модель автомата, алгоритм функционирования которого задан описанием:

«Автомат имеет два входа  $x_1, x_2$  и один выход  $y$ . В начальный момент времени  $y = 0$ . На вход подаются сигналы:  $(x_1, x_2) = (0,0), (0,1), (1,0)$  и  $(1,1)$ . В случае входной

#### [Оглавление](#)

комбинации  $(1,0)$  на выходе формируется значение 1; если  $(x_1, x_2) = (0,1)$ , то выдается  $y = 2$ . В остальных случаях  $y = 0$ .

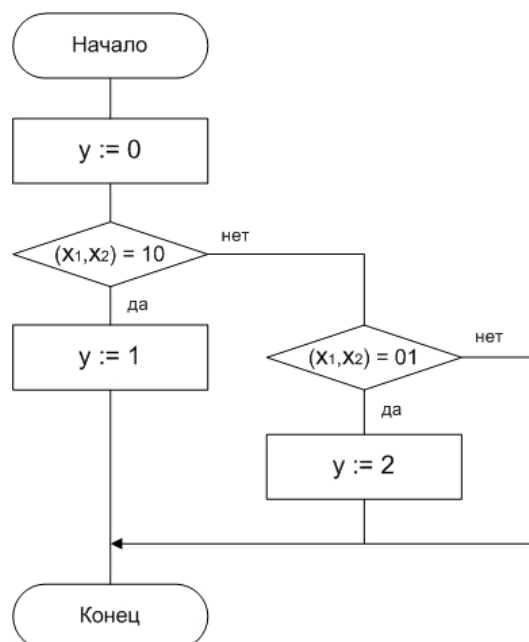
Как видно, описание работы автомата не обязательно связано с какой-либо конкретной системой счисления. Очевидно, в двоичной системе как вход, так и выход заданного автомата будут двухразрядными  $(x_1, x_2, y_1, y_2)$ ; в троичной системе вход останется двухразрядным, а выход будет одноразрядным  $(x_1, x_2, y)$ ; в 4-ричной системе вход и выход станут одноразрядными  $(x, y)$ . Однако для построения математической модели определение системы счисления несущественно.

Зададим множества, входящие в описание модели.

1.  $X = \{(0,0), (0,1), (1,0), (1,1)\}$ , где первый элемент каждой пары соответствует  $x_1$ , второй элемент –  $x_2$ . Для краткости запишем:  $X = \{00, 01, 10, 11\}$ .

2.  $Y = \{0, 1, 2\}$ .

3. Множество состояний  $S$  должно быть сформировано так, чтобы обрабатывалась каждая ветвь алгоритма функционирования автомата. Для этого алгоритм представим в виде *схемы*, иногда называемой *граф-схемой алгоритма*. Если каждый шаг алгоритма принять за *микрокоманду*, то схема алгоритма является наглядным изображением *микропрограммы* автомата как последовательности микрокоманд. Схема алгоритма заданного автомата представлена на рис. 1.4.



**Рис. 1.4.** Схема алгоритма функционирования КА

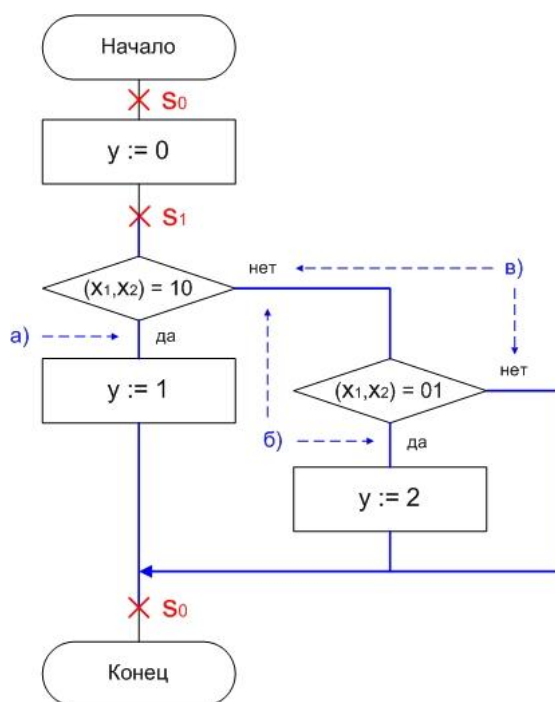
Состояния определим путем *разметки* схемы по следующим правилам, различным для моделей Мили и Мура.

### Модель Мили.

- 1). Вход блока, следующего за начальным, и вход конечного блока отвечают состоянию  $s_0$ .
- 2). Вход блока, следующего за операторным, отвечает состоянию  $s_i$ , где  $i = 1, 2, \dots$  – номер операторного блока<sup>\*)</sup>.

Заметим, что разметка отражает цикличность работы автомата. На уровне модели цикл бесконечный: алгоритм и начинается, и завершается состоянием  $s_0$ . В схемотехнической реализации функционирование продолжается, пока на электрическую схему автомата подается питание или действует разрешающий сигнал от автомата более высокого уровня иерархии.

Разметка схемы алгоритма для модели Мили показана на рис. 1.5. Имеем множество состояний:  $S = \{s_0, s_1\}$ .



**Рис. 1.5.** Разметка схемы алгоритма (модель Мили)

Построим взвешенный орграф переходов автомата Мили. Вершины графа соответствуют состояниям автомата. Дуга, направленная из вершины  $s_i$  в вершину  $s_j$ , задает переход вида:

$$s_i \xrightarrow{f_s} s_j.$$

<sup>\*)</sup> В схемах алгоритмов компьютерных программ операторному блоку усвоено наименование «процесс».

Переход инициируется входным сигналом  $x_k \in X$ . На переходе формируется выходной сигнал  $y_m \in Y$ . Поэтому весом дуги является пара «вход/выход»:  $x_k/y_m$ .

Проанализируем условия переходов. Переход КА из состояния  $s_0$  в состояние  $s_1$  является безусловным: булева функция, описывающая такой переход, тождественно равна единице, на графе соответствующая дуга будет помечена «1». Обратный переход из  $s_1$  в  $s_0$  возможен по трем ветвям алгоритма. На рис. 1.5 они обозначены как а), б), в) и указаны пунктирными стрелками. Условие прохода по каждой из ветвей представим в дизъюнктивной нормальной форме (ДНФ):

а)  $x_1\bar{x}_2$  – истинно первое условие  $(x_1, x_2) = 10$ , выдается выходной сигнал  $y = 1$ ;

б)  $\bar{x}_1\bar{x}_2 \wedge \bar{x}_1x_2 = (\bar{x}_1 \vee x_2) \wedge \bar{x}_1x_2 = \bar{x}_1x_2$  – ложно первое условие, истинно второе условие  $(x_1, x_2) = 01$ ,  $y = 2$ ;

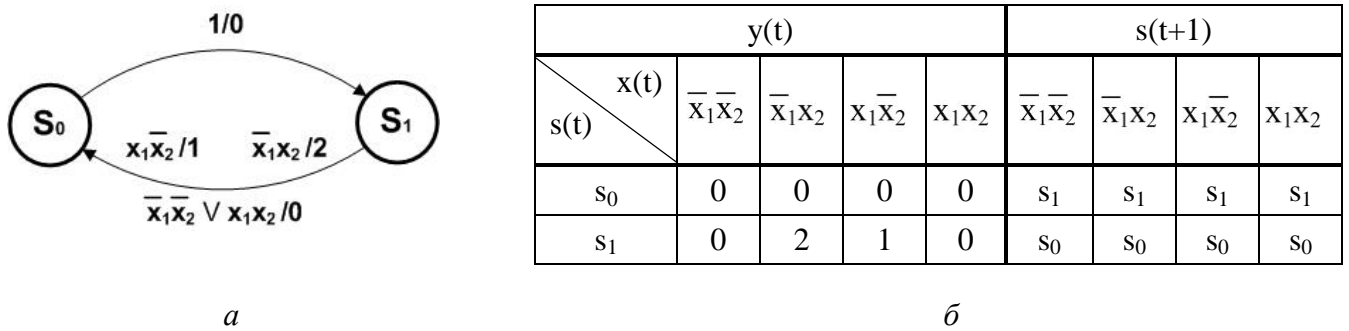
в)  $\bar{x}_1\bar{x}_2 \wedge \bar{x}_1x_2 = (\bar{x}_1 \vee x_2) \wedge (x_1 \vee \bar{x}_2) = \bar{x}_1\bar{x}_2 \vee x_1x_2$  – ложны оба условия, на данной ветви алгоритма операторные блоки отсутствуют, на выходе автомата сохраняется прежний сигнал  $y = 0$ .

Граф переходов показан на рис. 1.6, а. Веса дуг записаны в формате «вход/выход». Подчеркнем, что в автомате Мили выходной сигнал формируется именно на переходе КА из одного состояния в другое. Это становится понятным, если принять во внимание, что функция выходов автомата Мили зависит от двух аргументов: входного сигнала и текущего состояния. В процессе разметки схемы алгоритма каждый операторный блок оказывается между двумя состояниями, причем переход от одного к другому либо безусловный, когда операторные блоки расположены один за другим, либо условный, когда между операторными блоками присутствует не менее одного блока ветвления (см. рис. 1.5).

В данном примере действие, заключенное в операторный блок, состоит именно в выработке выходного сигнала. Если схема алгоритма отражает микропрограмму функционирования некоторого операционного автомата, например, умножения чисел, то в операторных блоках будут присутствовать действия над данными, например, сдвиг множимого в регистре, сложение частичной суммы и множимого на сумматоре и тому подобное. Но каждому действию соответствует инициирующий сигнал, вырабатываемый управляющим автоматом, который и задает требуемую алгоритмом последовательность микрокоманд.

На рис. 1.6, б приведена построенная по графу таблица переходов/выходов КА Мили. Строки таблицы соответствуют состояниям автомата, левая подтаблица отражает значения выходного сигнала в момент времени  $t$  для каждой пары «состояние – набор входных сигналов», правая подтаблица показывает, в какое состояние переходит автомат в следующий момент времени из данного состояния под воздействием каждого набора входных сигналов.

### [Оглавление](#)



**Рис. 1.6.** Результат абстрактного синтеза автомата Мили:

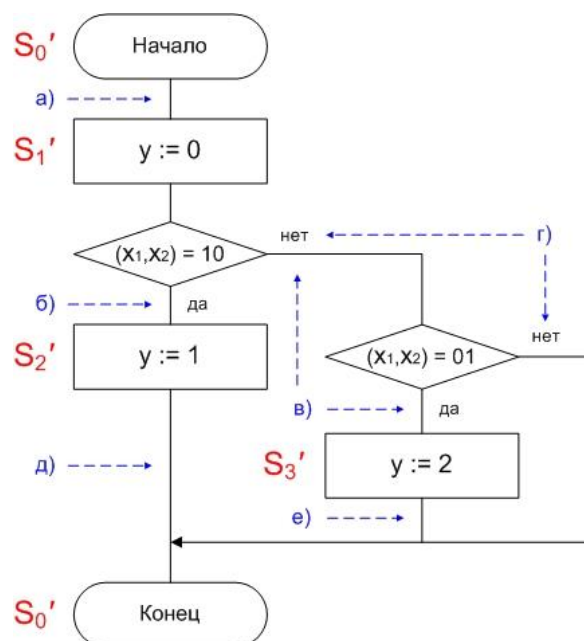
*a* – граф переходов; *б* – таблица переходов/выходов

Модель Мура. Разметка схемы алгоритма состоит в следующем.

- 1). Начальному и конечному блокам сопоставляют состояние  $s_0$ .
- 2). Каждому операторному блоку ставится в соответствие состояние  $s_i$ , где  $i = 1, 2, \dots$  – номер блока.

Таким образом, каждое состояние автомата Мура связано с действием, и обратно – каждое действие, то есть значение выходного сигнала, приписывается определенному состоянию. Напомним, что единственным аргументом функции выходов автомата Мура является его состояние в текущий момент времени.

Разметка схемы алгоритма для случая КА Мура показана на рис. 1.7.



**Рис. 1.7.** Разметка схемы алгоритма (модель Мура)

[Оглавление](#)

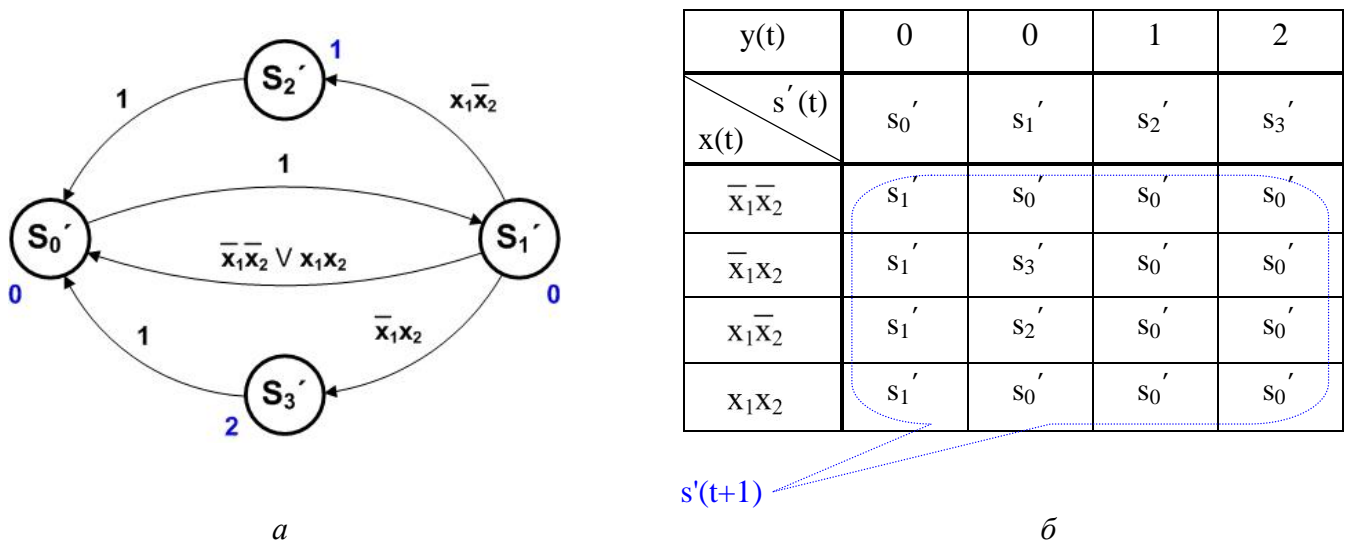
Имеем множество состояний:  $S' = \{s_0', s_1', s_2', s_3'\}$ . Условия переходов (на рис. 1.7 отмечены пунктирными стрелками) таковы:

- а)  $s_0' \rightarrow s_1'$  – безусловно;
- б)  $s_1' \rightarrow s_2' - x_1\bar{x}_2$ ;
- в)  $s_1' \rightarrow s_3' - \bar{x}_1\bar{x}_2 \wedge \bar{x}_1x_2 = \bar{x}_1x_2$ ;
- г)  $s_1' \rightarrow s_0' - \bar{x}_1\bar{x}_2 \wedge \bar{x}_1x_2 = \bar{x}_1\bar{x}_2 \vee x_1x_2$ ;
- д)  $s_2' \rightarrow s_0'$  – безусловно;
- е)  $s_3' \rightarrow s_0'$  – безусловно.

Граф переходов КА Мура показан на рис. 1.8, а. Дуга графа, инцидентная вершинам  $s_i'$  и  $s_j'$ , задает переход вида:

$$s_i' \xrightarrow{f_s'} s_j'.$$

Весом дуги является значение входного сигнала, инициировавшего переход. Так как выходной сигнал в момент времени  $t$  определяется исключительно текущим состоянием автомата, то обозначение выходного сигнала  $y_m$  на графе приписывается вершине  $s_i'$ , соответствующей состоянию, в котором этот выходной сигнал формируется. На рис. 1.8, а обозначения выходных сигналов выделены синим цветом.



**Рис. 1.8.** Результат абстрактного синтеза автомата Мура:

а – граф переходов; б – таблица переходов/выходов

Таблица переходов/выходов (рис. 1.8, б) в отличие от модели Мили не разделяется на две подтаблицы – выходов и переходов, поскольку каждому состоянию соответствует

[Оглавление](#)

определенное значение выходного сигнала независимо от сигналов на входах. Последние влияют только на переходы автомата: под воздействием каждого входного набора КА переходит из текущего состояния  $s'(t)$  в состояние  $s'(t+1)$  в соответствии с алгоритмом функционирования. ■<sup>\*</sup>)

#### 1.4. Переход от одной модели к другой: обоснование возможности и практика

##### А. Возможность перехода от модели Мили к модели Мура

Запишем уравнение функции выходов автомата Мили:

$$y(t) = f_y^{Ми}(x(t), s(t)).$$

Так как автомат один и тот же, *та же* функция выходов в случае модели Мура описывается уравнением:

$$y(t) = f_y^{Му}(s'(t))$$

(индексы «Ми» и «Му», обозначающие соответствующую модель, введены для наглядности).

Следовательно, имеет место соответствие аргументов:

$$s'(t) \simeq [x(t), s(t)].$$

Если данное соответствие рассмотреть для момента времени  $(t+1)$ , получим:

$$s'(t+1) \simeq [x(t+1), s(t+1)] = [x(t+1), f_s^{Ми}(x(t), s(t))].$$

Но  $s'(t+1) = f_s^{Му}(x(t+1), s'(t))$ . Тогда

$$f_s^{Му}(x(t+1), s'(t)) \simeq [x(t+1), f_s^{Ми}(x(t), s(t))].$$

Таким образом, зная функцию переходов автомата Мили  $f_s^{Ми}$ , можно перейти к функции переходов автомата Мура  $f_s^{Му}$ .

##### Б. Возможность перехода от модели Мура к модели Мили

Поставим в соответствие некоторому состоянию автомата Мура в момент времени  $(t-1)$  состояние автомата Мили в момент времени  $t$ :

$$s'(t-1) \simeq s(t) \quad (*)$$

Для  $y(t)$  имеем:

$$y(t) = f_y^{Му}(s'(t)) = f_y^{Му}(f_s^{Му}(x(t), s'(t-1))).$$

Используя соотношение (\*), получаем аргументы функции выходов автомата Мили:  $[x(t), s(t)]$ . Но  $y(t) = f_y^{Ми}(x(t), s(t))$ .

---

<sup>\*</sup>) Знаком «■» будем отмечать завершение рассмотрения примера, а также окончание доказательства теоремы или леммы, означающее высказывание «что и требовалось доказать».

Таким образом, зная  $f_y^{My}$  и  $f_s^{My}$ , можно перейти к  $f_y^{Mi}$ .

Из соотношения (\*) имеем для  $t+1$ :

$$s(t+1) \simeq s'(t), s'(t) = f_s^{My}(x(t), s'(t-1)) \simeq f_s(x(t), s(t)),$$

где  $f_s$  – некоторая функция переходов. Таким образом,  $s(t+1) \simeq f_s(x(t), s(t))$ . Однако  $x(t)$  и  $s(t)$  являются аргументами функции переходов автомата Мили:

$$s(t+1) = f_s^{Mi}(x(t), s(t)).$$

Следовательно, зная  $f_s^{My}$ , можно перейти к  $f_s^{Mi}$ .

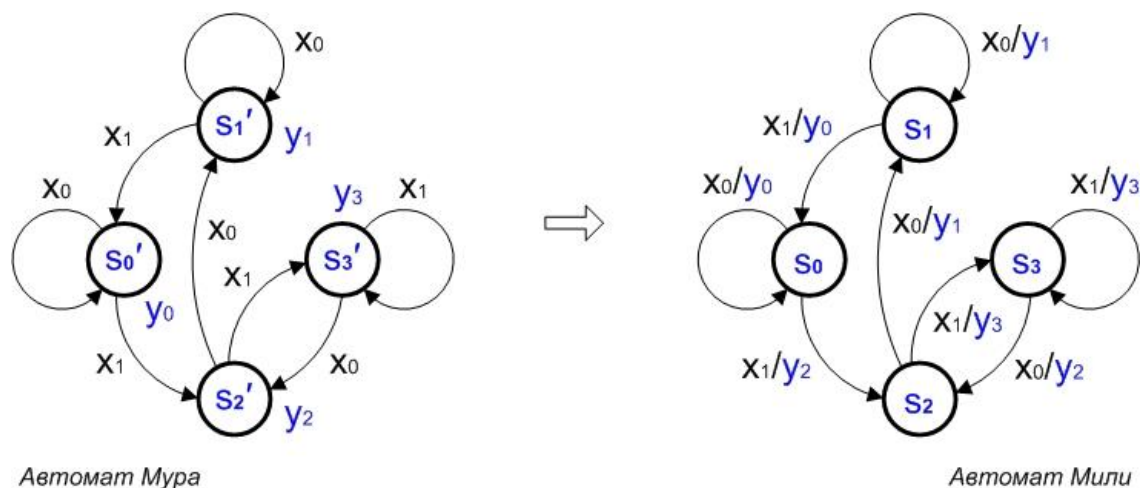
На основании рассуждений, проведенных в пунктах А и Б, можно заключить, что модели Мили и Мура функционально эквивалентны. Автомат, рассматриваемый как «черный ящик», одинаково реагирует на входные сигналы независимо от того, какая модель реализована – Мили или Мура. Иными словами, переход от одной модели КА к другой не влияет на результат преобразования автоматом слов входного алфавита (вектора входных сигналов).

Имеет место следующее утверждение.

**Лемма.** Для каждого конечного автомата Мили может быть построен эквивалентный ему конечный автомат Мура, и наоборот.

**Определение.** Два автомата с одинаковыми входными и выходными алфавитами (множествами сигналов)  $X$  и  $Y$  называются **эквивалентными**, если после установки обоих в начальное состояние их реакции на каждый набор входных сигналов совпадают.

На практике переход от автомата Мура к автомату Мили прост и нагляден в случае представления автомата графом переходов. Пример показан на рис. 1.9.



**Рис. 1.9.** Элементарный переход от автомата Мура к автомату Мили

Каждому *состоянию автомата Мура* (вершине графа КА Мура) ставится в соответствие одно *состояние автомата Мили* (вершина графа КА Мили). Переходы между состояниями

[Оглавление](#)



сохраняются. Таким образом, получаем граф автомата Мили, изоморфный исходному графу автомата Мура. Каждый *выходной сигнал КА Мура*, записанный рядом со своей вершиной, переносим на все дуги графа КА Мили, *входящие* в соответствующую ей вершину.

Алгоритм обратного перехода более сложный, поскольку в каждом состоянии  $s_i'$  автомата Мура вырабатывается выходной сигнал одного и только одного значения, а в автомате Мили выходной сигнал формируется на переходе из состояния в состояние и на различных переходах может повторяться. По этой причине приходится вводить состояния автомата Мура, соответствующие каждой паре «состояние – входной сигнал» автомата Мили. Из этого следует, что разные модели одного и того же автомата имеют, вообще говоря, различное число состояний, то есть  $|S^{Ми}| \neq |S^{Му}|$ , что было видно на примере абстрактного синтеза, рассмотренном в разделе 1.3.

Алгоритм перехода от автомата Мили к автомату Мура рассмотрим на следующем **примере**. Пусть автомат задан описанием: «Двоичные цифры 0 и 1 подаются на вход КА, который циклически по модулю 3 считает накопленное число единиц».

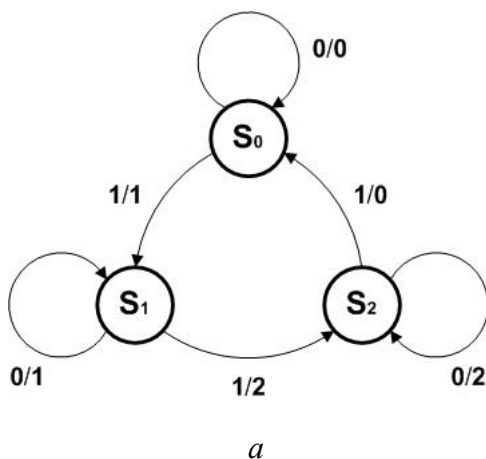
На рис. 1.10 представлены граф переходов и таблица переходов/выходов автомата Мили, полученного в результате абстрактного синтеза.

Входящие в модель множества таковы:

$$X = \{0, 1\};$$

$$Y = \{0, 1, 2\} \quad (\text{значения соответствуют числу принятых единиц});$$

$$S = \{s_0, s_1, s_2\} \quad (\text{автомат хранит: } s_0 \text{ – ноль единиц, } s_1 \text{ – одну единицу, } s_2 \text{ – две единицы}).$$



		y(t)		s(t+1)	
		x(t)	0	1	0
s(t)	0	0	1	$s_0$	$s_1$
	1	1	2	$s_1$	$s_2$
	2	2	0	$s_2$	$s_0$

*b*

**Рис. 1.10.** Исходный КА Мили:

*a* – граф переходов; *b* – таблица переходов/выходов

Выполним переход от данного КА Мили к эквивалентному автомату Мура.

### [Оглавление](#)

Шаг 1. Каждой паре «состояние  $s_i$  – входной сигнал  $x_j$ » автомата Мили ставим в соответствие состояние  $s_{ij}$  автомата Мура. Вводим начальное состояние автомата Мура  $s_0'$ .

В результате формируется подтаблица состояний автомата Мура (рис. 1.11):

		$s'(t)$		$s(t+1)$	
$x(t)$		0	1	0	1
$s(t)$					
$s_0 \simeq s_0'$		$s_{00}$	$s_{01}$	$s_0$	$s_1$
$s_1$		$s_{10}$	$s_{11}$	$s_1$	$s_2$
$s_2$		$s_{20}$	$s_{21}$	$s_2$	$s_0$

Подтаблица состояний автомата Мура      Подтаблица переходов автомата Мили

**Рис. 1.11.** Подтаблица состояний КА Мура, соединенная с подтаблицей переходов КА Мили

Шаг 2. Из полученной таблицы выписываем состояния автомата Мили и соответствующие им состояния автомата Мура. Например, состоянию  $s_0$  соответствуют  $s_{00}$ ,  $s_{21}$  (на рис. 1.11 это показано круглыми пунктирными выносками) и формально  $s_0'$ . Получаем:

$$s_0 \rightarrow \{s_0', s_{00}, s_{21}\}; \quad s_1 \rightarrow \{s_{01}, s_{10}\}; \quad s_2 \rightarrow \{s_{11}, s_{20}\}.$$

Шаг 3. Составляем таблицу переходов автомата Мура. Сначала выписываем в ряд все состояния автомата Мура. Затем заполняем столбцы для каждого значения входного сигнала, руководствуясь правилом: если на шаге 2 состояние  $s_{ij}$  вошло в множество для состояния  $s_p$  автомата Мили, то в столбец для  $s_{ij}$  включаются те состояния из подтаблицы состояний автомата Мура (рис. 1.11), которые входят в строку  $s_p$ .

К примеру, состояние  $s_{11}$  КА Мура входит в множество для состояния  $s_2$  КА Мили. Следовательно, в столбец для  $s_{11}$  войдут состояния:  $s_{20}$  при  $x = 0$  и  $s_{21}$  при  $x = 1$ . На рис. 1.11 эти соответствия указаны прямоугольной пунктирной выноской и двумя пунктирными линиями.

В последнюю очередь для каждого состояния автомата Мура проставляются значения выходного сигнала:  $f_y^{My}(s_{ij}) = f_y^{Mi}(s_i, x_j)$ . Например,  $f_y^{Mi}(s_2, 0) = 2$ , следовательно,  $f_y^{My}(s_{20}) = 2$ .

Состоянию  $s_0'$  можно усвоить такое же значение выходного сигнала, как и для  $s_{00}$ . Полученная таблица переходов/выходов КА Мура показана на рис. 1.12.

## [Оглавление](#)

y(t)	0	0	1	1	2	2	0
$s_{ij}(t)$	$s_0'$	$s_{00}$	$s_{01}$	$s_{10}$	$s_{11}$	$s_{20}$	$s_{21}$
x(t)							
0	$s_{00}$	$s_{00}$	$s_{10}$	$s_{10}$	$s_{20}$	$s_{20}$	$s_{00}$
1	$s_{01}$	$s_{01}$	$s_{11}$	$s_{11}$	$s_{21}$	$s_{21}$	$s_{01}$

$s_{ij}(t+1)$

**Рис. 1.12.** Таблица переходов/выходов эквивалентного КА Мура

**Шаг 4.** Из таблицы переходов видно, что среди состояний автомата есть такие, которым соответствуют: а) одни и те же значения выходного сигнала и б) переходы в одни и те же состояния при любых входных воздействиях. Это так называемые *эквивалентные состояния*. Например, состояния  $s_{00}$  и  $s_{21}$  – эквивалентные, так как находясь в каждом из них, автомат выдает сигнал  $y = 0$  и переходит в состояние  $s_{00}$  при  $x = 0$  и в состояние  $s_{01}$  при  $x = 1$ .

Эквивалентные состояния неразличимы, поэтому имеет смысл оставлять только одно из них. В выявлении неразличимых состояний и исключении лишних заключается так называемая *минимизация автомата*, которая будет подробно рассмотрена в главе 3.

В данном примере имеем три множества эквивалентных состояний:  $\{s_0', s_{00}, s_{21}\}$ ;  $\{s_{01}, s_{10}\}$ ;  $\{s_{11}, s_{20}\}$ . Для состояний каждого множества вводим новое обозначение:

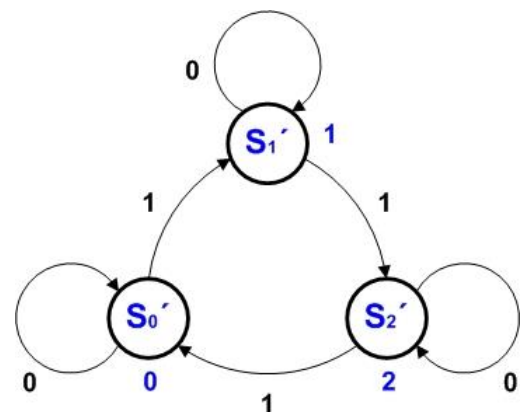
$$s_0' - \{s_0', s_{00}, s_{21}\}; \quad s_1' - \{s_{01}, s_{10}\}; \quad s_2' - \{s_{11}, s_{20}\}.$$

**Шаг 5.** Выполняем построение окончательной таблицы переходов/выходов и графа переходов автомата Мура (см. рис. 1.13).

y(t)	0	1	2
$s'(t)$	$s_0'$	$s_1'$	$s_2'$
x(t)			
0	$s_0'$	$s_1'$	$s_2'$
1	$s_1'$	$s_2'$	$s_0'$

$s'(t+1)$

а



б

**Рис. 1.13.** Окончательная модель КА Мура:

а – таблица переходов/выходов; б – граф переходов

## Оглавление

Обратим внимание, что полученный граф переходов изоморфен графу автомата Мили, что в общем случае необязательно. В данном примере изоморфизм объясняется составом множеств  $S$  и  $Y$ , сформированных на этапе абстрактного синтеза. Обычно после преобразования эквивалентный автомат Мура имеет большее число состояний, чем исходный автомат Мили. ■

### Контрольные вопросы

1. Каков [круг задач](#), решаемых в теории автоматов?
2. Что такое [абстрактный автомат](#)?
3. [Классифицируйте](#) абстрактные автоматы по трем признакам.
4. Что такое [конечный автомат](#)?
5. Каковы [характеристические функции](#) конечного автомата?
6. Какие существуют [способы задания](#) конечного автомата?
7. В чем сходство и различие [базовых моделей](#) конечных автоматов?
8. Какова [цель абстрактного синтеза](#) конечного автомата?
9. В чем заключается [процесс абстрактного синтеза](#) конечного автомата?
10. Чем различаются процессы [абстрактного синтеза автомата для моделей Мили и Мура](#)?
11. Поясните, в чем состоит отличие [графов переходов и таблиц переходов/выходов](#) моделей Мили и Мура.
12. Обоснуйте [возможность перехода](#) от одной модели конечного автомата к другой.
13. Как выполнить элементарный [переход от автомата Мура](#) к эквивалентному автомату Мили?
14. В чем заключается алгоритм [перехода от автомата Мили](#) к эквивалентному автомату Мура?
15. Обоснуйте [эквивалентность моделей Мили и Мура](#) одного и того же конечного автомата.

### Задания для самопроверки

1. Конечный автомат задан описанием алгоритма функционирования. Путем абстрактного синтеза получите модели Мили и Мура заданного автомата.

### [Оглавление](#)

*Вариант А.* Автомат представляет собой циклический счетчик импульсов от 0 до 7. На выходе автомата формируется сигнал  $y = 0$ , если на вход поступили от 0 до 3 импульсов, и  $y = 1$ , если их число от 4 до 7.

*Вариант Б.* Автомат реализует алгоритм бинарного поиска заданного числа в упорядоченном по возрастанию массиве из  $n$  чисел.

*Вариант В.* На вход автомата поступает конечная последовательность данных, состоящая из цифр 1, 2, 3 и сигнала  $\alpha$ . Автомат подсчитывает, сколько во входной последовательности цифр каждого вида. Суммы записываются в три отдельных регистра. После «отработки» каждой цифры входной последовательности автомат формирует выходной сигнал  $y_1$ . Когда на вход устройства поступает сигнал  $\alpha$ , вырабатывается сигнал  $y_2$ , сообщающий о выдаче результатов. Затем автомат выдерживает некоторый временной интервал и по его истечении обнуляет регистры, после чего готов к приему очередной последовательности данных. Считать, что переполнение регистров исключено.

2. Выполните переход от автомата Мили, синтезированного в задании 1, к эквивалентному автомату Мура. Сравните результат с полученной ранее моделью Мура.

## ГЛАВА 2. КЛАССЫ АВТОМАТОВ

### 2.1. Мощность множества конечных автоматов

**Лемма.** Мощность  $N_{КА}$  множества конечных автоматов с  $n$  состояниями,  $p$  входными сигналами и  $q$  выходными сигналами равна  $(qn)^{pn}$ .

Доказательство. Так как модели Мили и Мура функционально эквивалентны, достаточно провести доказательство для одной из них. Рассмотрим структуру таблицы переходов/выходов КА Мили (рис. 2.1):

		y(t)				s(t+1)			
		x <sub>1</sub>	x <sub>2</sub>	...	x <sub>p</sub>	x <sub>1</sub>	x <sub>2</sub>	...	x <sub>p</sub>
s(t)	s <sub>1</sub>								
	s <sub>2</sub>			y <sub>i</sub>				s <sub>j</sub>	
	...								
	s <sub>n</sub>								

$L_y$        $y_i \in Y = \{y_1, y_2, \dots, y_q\}$        $s_j \in S = \{s_1, s_2, \dots, s_n\}$        $L_s$

**Рис. 2.1.** Структура таблицы переходов/выходов автомата Мили

Всего способов заполнения одной строки подтаблицы выходов ( $L_y$ ) –  $q^p$ .

Всего способов заполнения одной строки подтаблицы переходов ( $L_s$ ) –  $n^p$ .

Так как строк в таблице  $n$ , то:

всего возможных подтаблиц выходов –  $q^{pn}$ .

всего возможных подтаблиц переходов –  $n^{pn}$ .

Тогда число всех возможных таблиц переходов/выходов конечных автоматов, равное мощности множества конечных автоматов, есть произведение этих двух величин:

$$N_{КА} = q^{pn} * n^{pn} = (qn)^{pn}. \blacksquare$$

### [Оглавление](#)

## 2.2. Класс явно-минимальных автоматов

Автомат называется **явно-минимальным**, если для каждой пары его состояний  $s_i, s_j$  ( $i \neq j$ ) найдется хотя бы один входной сигнал  $x_k \in X$ , для которого

$$f_y(x_k, s_i) \neq f_y(x_k, s_j).$$

Таким образом, реакция явно-минимального автомата на один и тот же входной сигнал разная для любой пары различных состояний. Ни одно из состояний не может быть удалено из множества  $S$  без потери эквивалентности полученного автомата исходному. Иными словами, сокращению автомат не подлежит.

**Лемма.** Мощность множества явно-минимальных автоматов  $N_{КА}^{ЯМ}$  равна:

$$N_{КА}^{ЯМ} = n^{p^n} \prod_{i=0}^{n-1} (q^p - i),$$

где  $p = |X|$ ,  $n = |S|$ ,  $q = |Y|$ .

**Пример.** Сколько можно построить явно-минимальных автоматов с тремя входными сигналами, двумя выходными сигналами и четырьмя состояниями?

Имеем:  $|X| = 3$ ,  $|Y| = 2$ ,  $|S| = 4$ .

$$N_{КА}^{ЯМ} = 4^{3 \cdot 4} \cdot [2^3(2^3-1)(2^3-2)(2^3-3)] = 2^{27} \cdot [7 \cdot 6 \cdot 5] = 134217728 \cdot 210 = 28'185'722'880. \blacksquare$$

## 2.3. Класс явно-сократимых автоматов

Автомат называется **явно-сократимым**, если при его представлении таблицей переходов/выходов найдется по крайней мере одна пара строк  $(s_i, s_j)$ , которые одинаковы как в подтаблице выходов  $y(t)$ , так и в подтаблице переходов  $s(t+1)$ .

Это означает, что состояния  $s_i$  и  $s_j$  неразличимы (эквивалентны): поведение автомата в состоянии  $s_i$  точно такое же, как в состоянии  $s_j$  при всех входных воздействиях. Поэтому иначе определить понятие явно-сократимого автомата можно следующим образом: автомат является явно-сократимым, если в множестве его состояний найдется хотя бы одна пара неразличимых (эквивалентных) состояний.

Присутствие в модели автомата неразличимых состояний избыточно. Как отмечалось выше, из всех таких состояний достаточно оставить одно, удалив из множества  $S$  остальные без потери эквивалентности полученного автомата исходному.

### [Оглавление](#)

Очевидно, подмножеств неразличимых состояний в явно-сократимом автомате может быть несколько, все они могут иметь различные мощности.

**Пример.** Пусть КА задан таблицей переходов/выходов, показанной на рис. 2.2.

		y(t)			s(t+1)		
		a	b	c	a	b	c
s(t) \ x(t)							
s <sub>0</sub>		0	1	1	s <sub>0</sub>	s <sub>1</sub>	s <sub>3</sub>
s <sub>1</sub>		0	1	0	s <sub>1</sub>	s <sub>2</sub>	s <sub>3</sub>
s <sub>2</sub>		1	1	1	s <sub>3</sub>	s <sub>1</sub>	s <sub>0</sub>
s <sub>3</sub>		0	1	0	s <sub>1</sub>	s <sub>2</sub>	s <sub>3</sub>

**Рис. 2.2.** Таблица переходов/выходов исходного автомата

Здесь  $X = \{a, b, c\}$ ;  $Y = \{0, 1\}$ ;  $S = \{s_0, s_1, s_2, s_3\}$ .

Строки для состояний  $s_1$  и  $s_3$  совпадают. Следовательно, автомат явно-сократимый. Удалим из множества  $S$  состояние  $s_3$ . Таблица сокращается на соответствующую состоянию  $s_3$  строку, в подтаблице переходов  $s_3$  заменяется на  $s_1$  (рис. 2.3).

		y(t)			s(t+1)		
		a	b	c	a	b	c
s(t) \ x(t)							
s <sub>0</sub>		0	1	1	s <sub>0</sub>	s <sub>1</sub>	s <sub>1</sub>
s <sub>1</sub>		0	1	0	s <sub>1</sub>	s <sub>2</sub>	s <sub>1</sub>
s <sub>2</sub>		1	1	1	s <sub>1</sub>	s <sub>1</sub>	s <sub>0</sub>

**Рис. 2.3.** Таблица переходов/выходов КА после сокращения состояния  $s_3$

Одинаковых строк больше нет.

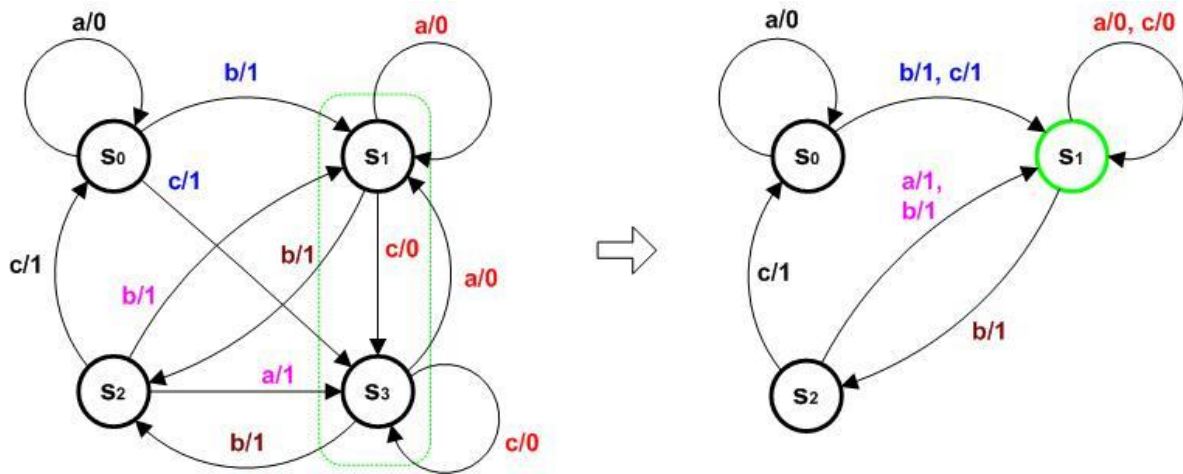
Построим графы переходов исходного и сокращенного автоматов (рис. 2.4).

Дуги, инцидентные  $s_1$ , приобрели кратный вес. Например, в графе исходного КА из вершины  $s_0$  выходит дуга в  $s_1$  с весом  $b/1$  и в  $s_3$  с весом  $c/1$ . Поэтому дуга  $s_0 \rightarrow s_1$  графа сокращенного КА имеет вес  $(b/1, c/1)$ . Аналогична ситуация с дугой  $s_2 \rightarrow s_1$  полученного графа. Кроме того, в исходном графе вершине  $s_2$  инцидентны дуги, исходящие из вершин  $s_1$  и  $s_3$  и

[Оглавление](#)



имеющие одинаковый вес  $b/1$ . Поэтому дуга  $s_1 \rightarrow s_2$  графа сокращенного КА приобретает вес  $b/1$ . Вершины  $s_1$  и  $s_3$  исходного графа смежны относительно дуг, имеющих веса  $c/0$  и  $a/0$ .



**Рис. 2.4.** Графы переходов исходного (слева) и сокращенного (справа) автоматов

Кроме того, каждой из этих вершин инцидентна петля: вершине  $s_1$  с весом  $a/0$ , вершине  $s_3$  с весом  $c/0$ . Поэтому вершине  $s_1$  графа сокращенного КА инцидентна петля с двойным весом  $(a/0, c/0)$ . Веса дуг исходного и результирующего графов в перечисленных случаях выделены на рис. 2.4 одинаковыми цветами.

Отметим, что полученный автомат принадлежит классу явно-минимальных. ■

**Лемма.** Мощность множества явно-сократимых автоматов  $N_{КА}^{ЯС}$  удовлетворяет неравенству:

$$N_{КА}^{ЯС} \geq (qn)^{pn} - \prod_{i=0}^{n-1} [(qn)^p - i].$$

Дадим пояснение. Ранее было показано, что  $(qn)^{pn} = N_{КА}$ . Оценка числа конечных автоматов с попарно различными строками в таблицах переходов/выходов  $N^{dif}$  может быть выражена неравенством:

$$N^{dif} \leq \prod_{i=0}^{n-1} [(qn)^p - i].$$

Отсюда следует неравенство для  $N_{КА}^{ЯС}$ . ■

## [Оглавление](#)

## 2.4. Изоморфные автоматы

Конечные автоматы, характеристические функции которых одинаковы, называются **изоморфными** друг другу. Это означает, что если некоторая система представлена автоматом  $A$ , то любой автомат, изоморфный  $A$ , также может служить представлением этой системы.

Пусть  $A$  – конечный автомат, заданный таблицей переходов/выходов. Если в подтаблице переходов выполнить перестановку символов  $s_1, s_2, \dots, s_n$ , то полученная таблица задаст автомат, изоморфный  $A$  (в подтаблице выходов ничего не меняется).

**Пример.** Автомат Мили задан таблицей переходов/выходов, показанной на рис. 2.5, *a*.  
Здесь:

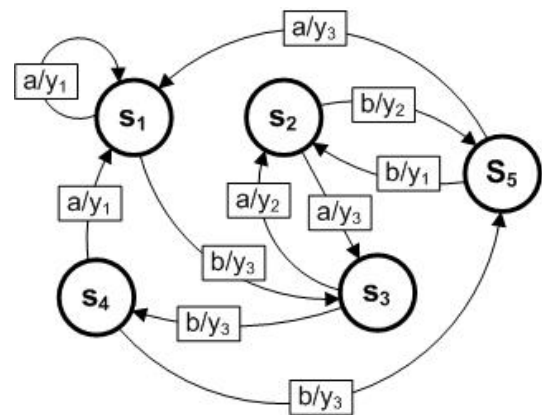
$$X = \{a, b\};$$

$$Y = \{y_1, y_2, y_3\};$$

$$S = \{s_1, s_2, s_3, s_4, s_5\}.$$

Выполним перестановку обозначений состояний по схеме:  $s_1 \rightarrow s_5 \rightarrow s_4 \rightarrow s_2 \rightarrow s_3 \rightarrow s_1$ . Преобразованная таблица переходов/выходов (см. рис. 2.6, *a*) задает автомат, изоморфный исходному; каждое состояние после перестановки выделено своим цветом.

		y(t)		s(t+1)	
		a	b	a	b
x(t) \ s(t)	s <sub>1</sub>	y <sub>1</sub>	y <sub>3</sub>	s <sub>1</sub>	s <sub>3</sub>
	s <sub>2</sub>	y <sub>3</sub>	y <sub>2</sub>	s <sub>3</sub>	s <sub>5</sub>
s <sub>3</sub>	y <sub>2</sub>	y <sub>3</sub>	s <sub>2</sub>	s <sub>4</sub>	
s <sub>4</sub>	y <sub>1</sub>	y <sub>3</sub>	s <sub>1</sub>	s <sub>5</sub>	
s <sub>5</sub>	y <sub>3</sub>	y <sub>1</sub>	s <sub>1</sub>	s <sub>2</sub>	

*a**б*

**Рис. 2.5.** Исходный автомат (модель Мили):

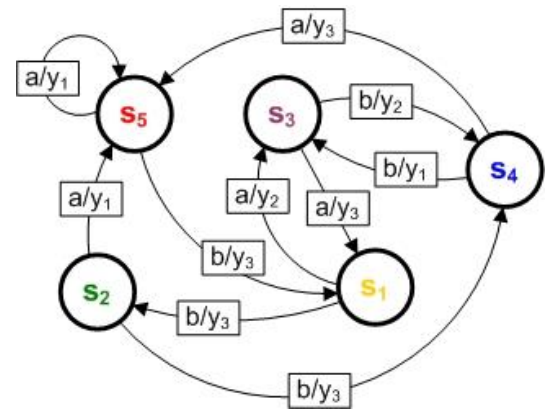
*a* – таблица переходов/выходов; *б* – граф переходов

Изоморфизм автоматов также следует из изображений их графов переходов. Графы, представленные на рисунках 2.5, *б* и 2.6, *б*, в данном случае отличаются только обозначениями вершин. Поэтому понятие изоморфизма автоматов допускает следующую интерпретацию: два автомата изоморфны, если изоморфны их графы переходов.

### [Оглавление](#)

		y(t)		s(t+1)	
		a	b	a	b
s(t)	x(t)				
		a	b	a	b
	$s_5$	$y_1$	$y_3$	$s_5$	$s_1$
	$s_3$	$y_3$	$y_2$	$s_1$	$s_4$
	$s_1$	$y_2$	$y_3$	$s_3$	$s_2$
	$s_2$	$y_1$	$y_3$	$s_5$	$s_4$
	$s_4$	$y_3$	$y_1$	$s_5$	$s_3$

а



б

**Рис. 2.6.** Автомат, изоморфный исходному:

а – таблица переходов/выходов; б – граф переходов

Примечание. Рамки, в которые заключены веса дуг на графах переходов, введены для удобства чтения. ■

Множество различных автоматов, полученных в результате всех возможных перестановок обозначений состояний, называют **семейством перестановок** автомата А.

**Лемма.** Мощность семейства перестановок явно-минимального автомата с  $n$  состояниями равна факториалу  $n$ .

Доказательство. Согласно определению явно-минимального автомата, строки в подтаблице  $y(t)$  различны. Они остаются различными и после перестановки обозначений состояний. Следовательно, таблицы переходов/выходов, получаемые в результате различных перестановок, различны. Так как строк в таблице  $n$ , то число различных перестановок равно  $n!$ . ■

Заметим, что при перестановках обозначений состояний не исключается замена исходных символов на другие.

**Теорема.** Мощность класса минимальных КА, не содержащих изоморфных автоматов, определяется формулой:

$$N_{\text{КА}}^{\text{МИ}} = \frac{n^{pn}}{n!} \prod_{i=0}^{n-1} (q^p - i).$$

Доказательство. Множество явно-минимальных автоматов представляет собой объединение семейств перестановок всех автоматов класса, определенного в формулировке теоремы. Поскольку эти семейства непересекающиеся и, согласно доказанной выше лемме, в каждом семействе ровно  $n!$  различных автоматов, то имеет место равенство:

### [Оглавление](#)

$$n! * N_{КА}^{М\backslash И} = N_{КА}^{ЯМ}.$$

Следовательно,

$$N_{КА}^{М\backslash И} = \frac{1}{n!} N_{КА}^{ЯМ} = \frac{n^{pn}}{n!} \prod_{i=0}^{n-1} (q^p - i). \quad \blacksquare$$

### Контрольные вопросы

1. Поясните вывод формулы для [мощности множества конечных автоматов](#).
2. Какие автоматы относят к [классу явно-минимальных](#)? Приведите пример явно-минимального автомата.
3. Какой автомат называют [явно-сократимым](#)? Приведите пример.
4. Почему и как изменяются [веса дуг графа переходов при удалении вершины](#), соответствующей одному из эквивалентных состояний явно-сократимого автомата? Поясните на примере.
5. Какой формулой оценивается [мощность множества явно-сократимых автоматов](#)? Дайте пояснение.
6. Что такое [изоморфные автоматы](#)? Приведите пример.
7. Каким образом можно наглядно убедиться в [изоморфизме](#) двух автоматов?
8. Сколько существует [минимальных автоматов, множество которых не содержит изоморфных автоматов](#)?

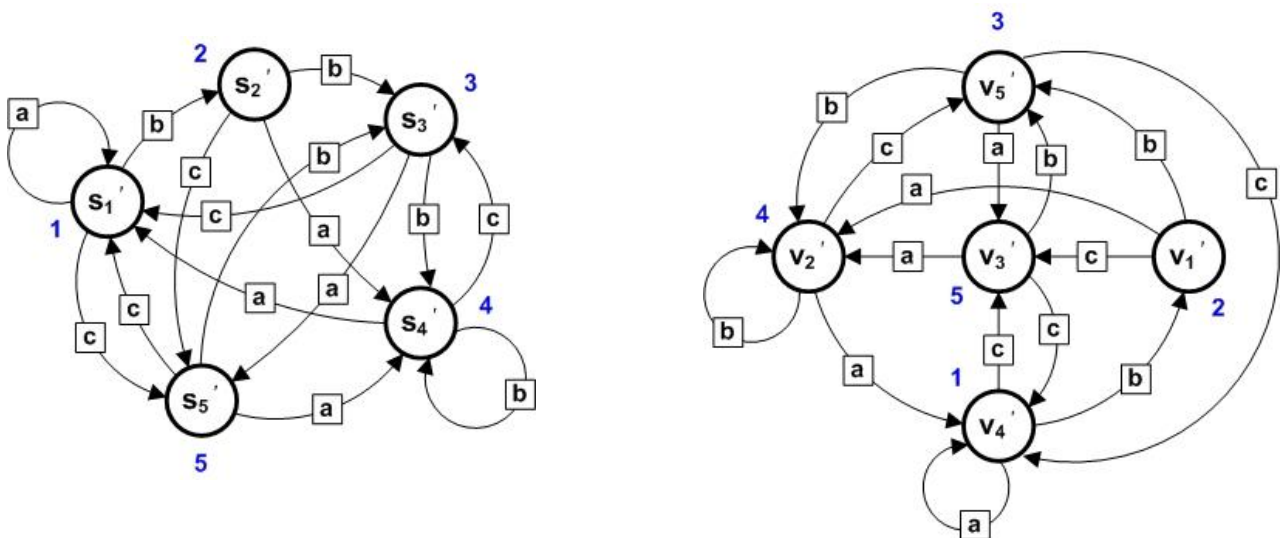
### Задания для самопроверки

1. Постройте график дискретной функции  $g(n) = (qn)^{pn}$  для мощности множества конечных автоматов с  $n$  состояниями,  $p$  входными сигналами и  $q$  выходными сигналами. Сделайте вывод о закономерностях поведения функции при увеличении (уменьшении):
  - а) числа входных сигналов;
  - б) числа выходных сигналов.
2. Предложите автомат с пятью состояниями:
  - а) явно-минимальный;
  - б) явно-сократимый.
 В каждом случае постройте таблицу переходов/выходов и граф переходов автомата.
3. КА задан таблицей переходов/выходов. Докажите принадлежность автомата классу явно-сократимых и выполните удаление избыточных состояний. Постройте и проанализируйте графы переходов исходного и полученного автоматов.

### [Оглавление](#)

		y(t)				s(t+1)			
		$\alpha$	$\beta$	$\gamma$	$\delta$	$\alpha$	$\beta$	$\gamma$	$\delta$
s(t)	x(t)								
s <sub>0</sub>		0	3	0	1	s <sub>0</sub>	s <sub>2</sub>	s <sub>1</sub>	s <sub>5</sub>
s <sub>1</sub>		2	0	1	3	s <sub>1</sub>	s <sub>4</sub>	s <sub>2</sub>	s <sub>3</sub>
s <sub>2</sub>		1	3	0	2	s <sub>3</sub>	s <sub>0</sub>	s <sub>2</sub>	s <sub>4</sub>
s <sub>3</sub>		0	3	0	1	s <sub>0</sub>	s <sub>2</sub>	s <sub>1</sub>	s <sub>5</sub>
s <sub>4</sub>		2	0	1	3	s <sub>1</sub>	s <sub>4</sub>	s <sub>2</sub>	s <sub>3</sub>
s <sub>5</sub>		1	3	2	0	s <sub>3</sub>	s <sub>0</sub>	s <sub>4</sub>	s <sub>2</sub>

4. Два автомата Мура заданы графами переходов:



Постройте таблицу переходов/выходов каждого автомата. Являются ли автоматы изоморфными? Почему?

## ГЛАВА 3. МИНИМАЛЬНЫЕ АВТОМАТЫ

### 3.1. Эквивалентные состояния автомата и их свойства

Рассмотрим следующую модель. Пусть автомат  $A$  последовательно находится в состояниях  $s_i$  и  $s_j$ , причем  $s_i \neq s_j$ . И в том, и в другом случае на вход автомата подается входной сигнал  $x$ , на выходе формируется сигнал  $y$  (см. рис. 3.1). Возможно ли отличить состояния  $s_i$  и  $s_j$  одно от другого? Для ответа на этот вопрос необходимо ввести ряд определений.

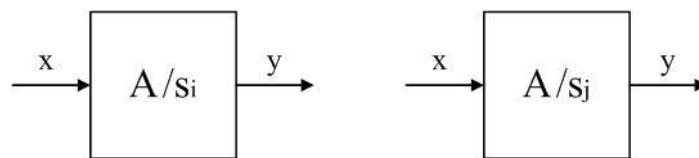


Рис. 3.1. КА в двух состояниях

**Определение 1.** Два состояния  $s_i$  и  $s_j$  автомата  $A$  называются **1-эквивалентными**, если при подаче на вход автомата одного и того же входного сигнала  $x$  реакция автомата (значение выходного сигнала  $y$ ) будет одной и той же независимо от того, находится ли  $A$  в состоянии  $s_i$  или в состоянии  $s_j$ .

**Определение 2.** Два состояния  $s_i$  и  $s_j$  автомата  $A$  называются **1-различимыми**, если при подаче на вход автомата одного и того же входного сигнала  $x$  реакция автомата будет разной в зависимости от того, находится  $A$  в состоянии  $s_i$  или в состоянии  $s_j$ .

**Определение 3.** Два состояния  $s_i$  и  $s_j$  автомата  $A$  называются **k-эквивалентными**, если при воздействии на автомат одной и той же последовательностью входных сигналов длиной  $k$  автомат выдает одну и ту же выходную последовательность независимо от того, находится ли  $A$  в состоянии  $s_i$  или в состоянии  $s_j$ .

**Определение 4.** Два состояния  $s_i$  и  $s_j$  автомата  $A$  называются **k-различимыми**, если при воздействии на автомат одной и той же последовательностью входных сигналов длиной  $k$  реакция автомата будет представлять собой различные выходные последовательности в зависимости от того, находится  $A$  в состоянии  $s_i$  или в состоянии  $s_j$ .

**Определение 5.** Два состояния  $s_i$  и  $s_j$  автомата  $A$  называются (просто) **эквивалентными**, если при воздействии на автомат одной и той же произвольной последовательностью входных

[Оглавление](#)

сигналов из множества  $X$  реакция автомата будет одинаковой вне зависимости от того, в каком состоянии находится автомат –  $s_i$  или  $s_j$ .

**Определение 6.** Два состояния  $s_i$  и  $s_j$  автомата  $A$  называются (просто) **различимыми**, если при воздействии на автомат одной и той же произвольной последовательностью входных сигналов из множества  $X$  реакция автомата будет различной в зависимости от того, в каком состоянии находится автомат –  $s_i$  или  $s_j$ .

Примечание. В контексте определений 1 – 6 можно говорить об эквивалентности или различимости состояний  $s_i$  и  $s_j$  разных автоматов, то есть когда  $s_i \in S_1$ ,  $s_j \in S_2$ , где  $S_1$  и  $S_2$  – множества состояний автоматов  $A_1$  и  $A_2$  соответственно.

Рассмотрим **свойства эквивалентных состояний** и продолжим ряд определений.

**Свойство 1 (лемма).** Если два состояния КА являются  $k$ -эквивалентными, то они будут и  $m$ -эквивалентными для каждого  $m \leq k$ .

Доказательство. От противного. Пусть состояния КА  $s_i$  и  $s_j$   $k$ -эквивалентны, но  $m$ -различимы ( $m \leq k$ ). Тогда существует входная последовательность длиной  $m$ , на которой  $s_i$  отличается от  $s_j$ . Но в таком случае  $s_i$  и  $s_j$  будут различимы и при входной последовательности длиной  $k$ , то есть  $s_i$  и  $s_j$  являются  $k$ -различимыми, что противоречит условию леммы. Следовательно, предположение неверно,  $s_i$  и  $s_j$   $m$ -эквивалентны. ■

**Свойство 2 (лемма).** Если два состояния КА являются  $k$ -различимыми, то они будут и  $m$ -различимыми для каждого  $m \geq k$ .

Доказательство. Аналогично предыдущему или по свойству 1: пусть  $s_i$  и  $s_j$   $k$ -различимы, но  $m$ -эквивалентны ( $m \geq k$ ). Однако на основании свойства 1, если два состояния  $m$ -эквивалентны, то они должны быть и  $k$ -эквивалентны, так как  $k \leq m$ . Полученное противоречие доказывает справедливость леммы. ■

**Определение 7.** Состояние, в которое переходит КА из состояния  $s_i$  при подаче входной последовательности длиной  $k$ , называют  **$k$ -тым преемником**  $s_i$  по отношению к данной входной последовательности.

В частном случае, нулевым преемником состояния  $s_i$  является само  $s_i$ .

На практике представляет интерес так называемое *разбиение* множества состояний КА на *классы* по следующим правилам:

- 1) все состояния, принадлежащие одному классу, должны быть  $k$ -эквивалентными;
- 2) все состояния, принадлежащие разным классам, должны быть  $k$ -различимыми.

## [Оглавление](#)

**Определение 8.** Разбиение множества  $S$  состояний КА на классы в соответствии с правилами 1) и 2) называется **к-эквивалентным разбиением** автомата и обозначается  $P_k$ .

**Определение 9.** Классы к-эквивалентного разбиения автомата называются **классами к-эквивалентности** и обозначаются  $\Sigma_{k1}, \Sigma_{k2}, \dots$

**Определение 10.** Состояния КА, принадлежащие одному и тому же классу к-эквивалентности  $\Sigma_{ki}$ , называются **смежными**; состояния, принадлежащие разным классам к-эквивалентности, называют **разобщенными**.

Замечание. Ни одно из состояний КА не может принадлежать одновременно двум разным классам к-эквивалентности, так как это означало бы к-различимость состояния самому себе. Следовательно, суммарное число состояний в к-эквивалентном разбиении  $P_k$  равно мощности множества состояний КА:

$$\begin{aligned} P_k &= \{\Sigma_{k1}, \Sigma_{k2}, \dots, \Sigma_{km}\}; \\ \Sigma_{ki} \cap \Sigma_{kj} &= \emptyset, \quad i = \overline{1, m}, \quad j = \overline{1, m}, \quad i \neq j; \\ |P_k| &= |\Sigma_{k1}| + |\Sigma_{k2}| + \dots + |\Sigma_{km}| = |S|. \end{aligned}$$

**Свойство 3 (лемма).** к-эквивалентное разбиение автомата единственно.

Доказательство проведем от противного. Пусть к-эквивалентное разбиение  $P_k$  состоит из классов к-эквивалентности:

$$P_k = \{\Sigma_{k1}, \Sigma_{k2}, \dots, \Sigma_{km}\}.$$

Пусть  $P_k$  не единственно. Тогда существует другое, альтернативное к-эквивалентное разбиение  $P'_k$  того же автомата:

$$P'_k = \{\Sigma'_{k1}, \Sigma'_{k2}, \dots, \Sigma'_{kv}\}.$$

Пусть  $\Sigma_{kj} \in P_k, \Sigma_{kj} = \{s_{j1}, s_{j2}, \dots, s_{jd}\}$ .

Поскольку каждое состояние, входящее в  $\Sigma_{kj}$ , к-эквивалентно любому другому состоянию из этого же класса и не существует ни одного состояния вне  $\Sigma_{kj}$ , которое было бы к-эквивалентно хотя бы одному состоянию в  $\Sigma_{kj}$ , то в альтернативном разбиении  $P'_k$  должен быть класс эквивалентности  $\Sigma'_{ki}$ , включающий те же состояния, что и  $\Sigma_{kj}$ , и не содержащий никаких других состояний:

$$\Sigma'_{ki} = \{s_{j1}, s_{j2}, \dots, s_{jd}\}.$$

Предположив  $j = \overline{1, m}$ , получим, что каждому классу в  $P_k$  соответствует идентичный класс в  $P'_k$ . Так как суммарное число состояний в  $P'_k$  такое же, как в  $P_k$ , то

$$P'_k = P_k.$$

Таким образом, к-эквивалентное разбиение автомата является единственным. ■

## [Оглавление](#)



**Свойство 4 (лемма).** Состояния КА, являющиеся разобщенными в  $P_k$ , должны быть разобщенными и в  $(k+1)$ -эквивалентном разбиении  $P_{k+1}$ .

Доказательство следует из свойства 2.

**Свойство 5 (лемма).** Если КА имеет два различных, но  $k$ -эквивалентных состояния, то он также имеет два состояния  $k$ -эквивалентных, но  $(k+1)$ -различимых.

Доказательство также следует из свойства 2.

**Определение 11.**  $k$ -эквивалентное разбиение автомата  $A$  называется (просто) **эквивалентным разбиением**, если во всех классах этого разбиения смежные состояния эквивалентны.

Обозначение:  $P_{\text{экв}}$ .

**Определение 12.** Каждый класс, входящий в эквивалентное разбиение, называется **классом эквивалентности**.

Обозначение:  $\Sigma_1, \Sigma_2, \dots$

Таким образом,  $P_{\text{экв}} = \{\Sigma_1, \Sigma_2, \dots, \Sigma_m\}$ . Эквивалентное разбиение является наиболее детальным разбиением автомата. Оно может быть получено итерационно путем построения  $k$ -эквивалентных разбиений при  $k = 1, 2, 3$  и т.д. При совпадении очередного разбиения  $P_i$  с предыдущим  $P_{i-1}$  имеем  $P_{\text{экв}}$ . Очевидно, для эквивалентного разбиения автомата с  $n$  состояниями потребуется не более  $(n-1)$  итерации.

Понятия эквивалентных состояний и эквивалентного разбиения позволяют ввести понятие эквивалентных автоматов. Два автомата  $A_1$  и  $A_2$  называют *эквивалентными*, если каждому состоянию  $s_i$  автомата  $A_1$  соответствует, по крайней мере, одно эквивалентное ему состояние автомата  $A_2$  и, наоборот, каждому состоянию  $s_j$  автомата  $A_2$  сопоставимо хотя бы одно эквивалентное ему состояние автомата  $A_1$ . В противном случае автоматы  $A_1$  и  $A_2$  называют *различимыми*.

### 3.2. Минимальная форма автомата

Пусть  $A$  – автомат с  $m$  классами эквивалентности:  $P_{\text{экв}} = \{\Sigma_1, \Sigma_2, \dots, \Sigma_m\}$ ,  $m \leq n$ ,  $n = |S|$ . Введем обозначение  $s^{(i)}$  для некоторого состояния, входящего в класс  $\Sigma_i$ :

$$s^{(i)} \in \Sigma_i, i = \overline{1, m}.$$

**Минимальной формой автомата**  $A$  называют автомат  $\tilde{A}$ , имеющий  $m$  состояний, которые образуют множество:

$$\{s^{(1)}, s^{(2)}, \dots, s^{(m)}\}.$$

#### [Оглавление](#)

Процесс отыскания минимальной формы называют **минимизацией автомата**. Полученный автомат называют **минимальным**. Очевидно, минимизация состоит в определении  $P_{\text{экв}}$  и последующем построении характеристических функций  $\tilde{f}_s, \tilde{f}_y$  автомата  $\tilde{A}$  с учетом того, что каждый класс в  $P_{\text{экв}}$  – это одно состояние  $\tilde{A}$ .

Таким образом, индивидуальное распознавание каждого состояния исходного автомата становится ненужным: важно только определить, к какому классу эквивалентности оно относится.

Автомат можно привести к минимальной форме различными путями. Одним из наиболее распространенных способов минимизации является **итерационный алгоритм Мили**. Алгоритм состоит из двух этапов, на которых осуществляется поиск пар эквивалентных состояний. Рассмотрим эти этапы.

Этап 1, шаг 1. Получение первичного разбиения.

Два состояния  $s_i$  и  $s_j$  относим к одному классу эквивалентности  $\Sigma_{1h}$ , если для каждого  $x_k \in X$  выполняется равенство:

$$f_y(s_i, x_k) = f_y(s_j, x_k),$$

то есть при любом входном воздействии реакция автомата одинакова в обоих состояниях.

Этап 2, шаг  $(i + 1)$ ,  $i \geq 1$ . Итерационное разбиение ранее полученных классов.

Два состояния  $s_u$  и  $s_v$ , принадлежащие одному классу эквивалентности  $\Sigma_{ij}$  (индекс  $i$  – номер шага, индекс  $j$  – номер класса), относим к классу  $\Sigma_{i+1j}$ , если для каждого  $x_k \in X$  верно:

$$\begin{cases} f_s(s_u, x_k) \in \Sigma_{iv} \\ f_s(s_v, x_k) \in \Sigma_{iv}, \end{cases}$$

то есть под воздействием каждого входного сигнала автомат, находясь как в  $s_u$ , так и в  $s_v$ , переходит в состояния, принадлежащие одному и тому же классу эквивалентности  $\Sigma_{iv}$ , полученному на предыдущем шаге разбиения.

Условием завершения алгоритма является совпадение разбиений на очередном и предыдущем шагах.

**Пример.** Автомат Мили  $A$  задан таблицей переходов/выходов (см. рис. 3.2). Состояния для краткости обозначены цифрами 1, 2, ..., 9. Требуется получить минимальную форму автомата.

Этап 1, шаг 1. Согласно алгоритму Мили, первичное разбиение должно отвечать следующему условию. Если несколько (два и более) состояний при конкретном значении входного сигнала обеспечивают выдачу выходного сигнала одного и того же значения, и это

## [Оглавление](#)

верно для каждого из всех возможных значений входного сигнала, то эти состояния составляют класс эквивалентности.

		y(t)			s(t+1)		
		a <sub>1</sub>	a <sub>2</sub>	a <sub>3</sub>	a <sub>1</sub>	a <sub>2</sub>	a <sub>3</sub>
s(t) \ x(t)	1	0	1	1	2	4	4
	2	1	0	0	1	1	5
3	1	0	0	1	6	5	
4	0	1	1	8	1	1	
5	1	1	0	6	4	3	
6	0	1	1	8	9	6	
7	1	1	0	6	1	3	
8	1	0	0	4	4	7	
9	0	1	1	7	9	7	

**Рис. 3.2.** Таблица переходов/выходов исходного автомата А

Процесс первичного разбиения отражен в таблице на рис. 3.3.

Вх. / Вых.	s(t)	Вх. / Вых.	s(t)
x = a <sub>1</sub> / y = 0	<u>1, 4, 6, 9</u>	x = a <sub>1</sub> / y = 1	<u>2, 3, 5, 7, 8</u>
x = a <sub>2</sub> / y = 0	<u>2, 3, 8</u>	x = a <sub>2</sub> / y = 1	<u>1, 4, 5, 6, 7, 9</u>
x = a <sub>3</sub> / y = 0	<u>2, 3, 5, 7, 8</u>	x = a <sub>3</sub> / y = 1	<u>1, 4, 6, 9</u>

**Рис. 3.3.** Этап 1. Первичное разбиение множества состояний автомата А

В результате шага 1 получаем три класса эквивалентности:

$$\Sigma_{11} = \{1, 4, 6, 9\}, \Sigma_{12} = \{2, 3, 8\}, \Sigma_{13} = \{5, 7\}.$$

Этап 2. На каждом шаге данного этапа последовательно рассматриваем классы, полученные на предыдущем шаге. В текущем классе для каждого состояния и каждого входного сигнала определяем, какому классу предыдущего шага принадлежит состояние, в которое переходит автомат в момент времени t+1.

### [Оглавление](#)

Шаг 2. Разбиение классов, полученных на шаге 1 (см. рис. 3.4).

s(t) \ x(t)		s(t+1)			s(t) \ x(t)		s(t+1)			s(t) \ x(t)		s(t+1)		
		a <sub>1</sub>	a <sub>2</sub>	a <sub>3</sub>			a <sub>1</sub>	a <sub>2</sub>	a <sub>3</sub>			a <sub>1</sub>	a <sub>2</sub>	a <sub>3</sub>
$\Sigma_{11}$	1	$\Sigma_{12}$	$\Sigma_{11}$	$\Sigma_{11}$	$\Sigma_{12}$	2	$\Sigma_{11}$	$\Sigma_{11}$	$\Sigma_{13}$	$\Sigma_{13}$	5	$\Sigma_{11}$	$\Sigma_{11}$	$\Sigma_{12}$
	4	$\Sigma_{12}$	$\Sigma_{11}$	$\Sigma_{11}$		3	$\Sigma_{11}$	$\Sigma_{11}$	$\Sigma_{13}$		7	$\Sigma_{11}$	$\Sigma_{11}$	$\Sigma_{12}$
	6	$\Sigma_{12}$	$\Sigma_{11}$	$\Sigma_{11}$		8	$\Sigma_{11}$	$\Sigma_{11}$	$\Sigma_{13}$					
	9	$\Sigma_{13}$	$\Sigma_{11}$	$\Sigma_{13}$										

**Рис. 3.4.** Шаг 2: разбиение классов шага 1

На шаге 2 получаем следующие классы эквивалентности:

$\Sigma_{21} = \{1, 4, 6\}$ ,  $\Sigma_{22} = \{2, 3, 8\}$ ,  $\Sigma_{23} = \{5, 7\}$ ; состояние 9 выделяется в отдельный класс:  $\{9\}$ .

Шаг 3 (см. рис. 3.5).

s(t) \ x(t)		s(t+1)			s(t) \ x(t)		s(t+1)			s(t) \ x(t)		s(t+1)		
		a <sub>1</sub>	a <sub>2</sub>	a <sub>3</sub>			a <sub>1</sub>	a <sub>2</sub>	a <sub>3</sub>			a <sub>1</sub>	a <sub>2</sub>	a <sub>3</sub>
$\Sigma_{21}$	1	$\Sigma_{22}$	$\Sigma_{21}$	$\Sigma_{21}$	$\Sigma_{22}$	2	$\Sigma_{21}$	$\Sigma_{21}$	$\Sigma_{23}$	$\Sigma_{23}$	5	$\Sigma_{21}$	$\Sigma_{21}$	$\Sigma_{22}$
	4	$\Sigma_{22}$	$\Sigma_{21}$	$\Sigma_{21}$		3	$\Sigma_{21}$	$\Sigma_{21}$	$\Sigma_{23}$		7	$\Sigma_{21}$	$\Sigma_{21}$	$\Sigma_{22}$
	6	$\Sigma_{22}$	$\{9\}$	$\Sigma_{21}$		8	$\Sigma_{21}$	$\Sigma_{21}$	$\Sigma_{23}$					

**Рис. 3.5.** Шаг 3: разбиение классов шага 2

Результат шага 3:  $\Sigma_{31} = \{1, 4\}$ ,  $\Sigma_{32} = \{2, 3, 8\}$ ,  $\Sigma_{33} = \{5, 7\}$ ;  $\{6\}$ ,  $\{9\}$ .

Шаг 4 (см. рис. 3.6).

s(t) \ x(t)		s(t+1)			s(t) \ x(t)		s(t+1)			s(t) \ x(t)		s(t+1)		
		a <sub>1</sub>	a <sub>2</sub>	a <sub>3</sub>			a <sub>1</sub>	a <sub>2</sub>	a <sub>3</sub>			a <sub>1</sub>	a <sub>2</sub>	a <sub>3</sub>
$\Sigma_{31}$	1	$\Sigma_{32}$	$\Sigma_{31}$	$\Sigma_{31}$	$\Sigma_{32}$	2	$\Sigma_{31}$	$\Sigma_{31}$	$\Sigma_{33}$	$\Sigma_{33}$	5	$\{6\}$	$\Sigma_{31}$	$\Sigma_{32}$
	4	$\Sigma_{32}$	$\Sigma_{31}$	$\Sigma_{31}$		3	$\Sigma_{31}$	$\{6\}$	$\Sigma_{33}$		7	$\{6\}$	$\Sigma_{31}$	$\Sigma_{32}$
						8	$\Sigma_{31}$	$\Sigma_{31}$	$\Sigma_{33}$					

**Рис. 3.6.** Шаг 4: разбиение классов шага 3

[Оглавление](#)

Результат шага 4:  $\Sigma_{41} = \{1, 4\}$ ,  $\Sigma_{42} = \{2, 8\}$ ,  $\Sigma_{43} = \{5, 7\}$ ;  $\{3\}$ ,  $\{6\}$ ,  $\{9\}$ .

Шаг 5 (см. рис. 3.7).

		x(t)			s(t+1)					x(t)			s(t+1)		
		a <sub>1</sub>	a <sub>2</sub>	a <sub>3</sub>	a <sub>1</sub>	a <sub>2</sub>	a <sub>3</sub>			a <sub>1</sub>	a <sub>2</sub>	a <sub>3</sub>			
$\Sigma_{41}$	1	$\Sigma_{42}$	$\Sigma_{41}$	$\Sigma_{41}$	$\Sigma_{42}$	2	$\Sigma_{41}$	$\Sigma_{41}$	$\Sigma_{43}$	$\Sigma_{43}$	5	{6}	$\Sigma_{41}$	{3}	
	4	$\Sigma_{42}$	$\Sigma_{41}$	$\Sigma_{41}$		8	$\Sigma_{41}$	$\Sigma_{41}$	$\Sigma_{43}$		7	{6}	$\Sigma_{41}$	{3}	

**Рис. 3.7.** Шаг 5: попытка разбиения классов шага 4

Результат шага 5:  $\Sigma_{51} = \{1, 4\}$ ,  $\Sigma_{52} = \{2, 8\}$ ,  $\Sigma_{53} = \{5, 7\}$ ;  $\{3\}$ ,  $\{6\}$ ,  $\{9\}$ . Имеем совпадение с разбиением, полученным на шаге 4. Конец итераций.

В итоге применения алгоритма Мили получены шесть классов эквивалентности. Следовательно, число состояний автомата в минимальной форме равно шести. Введем обозначения:

$$s^{(1)} = \{1, 4\}, s^{(2)} = \{2, 8\}, s^{(3)} = \{3\}, s^{(4)} = \{5, 7\}, s^{(5)} = \{6\}, s^{(6)} = \{9\}.$$

Таблица переходов/выходов минимального автомата  $\tilde{A}$  имеет вид, показанный на рис. 3.8.

		y(t)			s(t+1)		
		a <sub>1</sub>	a <sub>2</sub>	a <sub>3</sub>	a <sub>1</sub>	a <sub>2</sub>	a <sub>3</sub>
$s^{(1)}$	$s^{(1)}$	0	1	1	$s^{(2)}$	$s^{(1)}$	$s^{(1)}$
$s^{(2)}$	$s^{(1)}$	1	0	0	$s^{(1)}$	$s^{(1)}$	$s^{(4)}$
$s^{(3)}$	$s^{(1)}$	1	0	0	$s^{(1)}$	$s^{(5)}$	$s^{(4)}$
$s^{(4)}$	$s^{(5)}$	1	1	0	$s^{(5)}$	$s^{(1)}$	$s^{(3)}$
$s^{(5)}$	$s^{(2)}$	0	1	1	$s^{(2)}$	$s^{(6)}$	$s^{(5)}$
$s^{(6)}$	$s^{(4)}$	0	1	1	$s^{(4)}$	$s^{(6)}$	$s^{(4)}$

**Рис. 3.8.** Таблица переходов/выходов минимального автомата  $\tilde{A}$

Фрагменты графов автоматов  $A$  и  $\tilde{A}$  представлены на рис. 3.9. Соответствия вершин показаны пунктирными линиями.

## Оглавление

Читателю рекомендуется в качестве упражнения самостоятельно проанализировать оба графа, сопоставить картины инцидентности дуг вершинам, проставить веса дуг и убедиться в функциональной эквивалентности минимального автомата исходному. ■

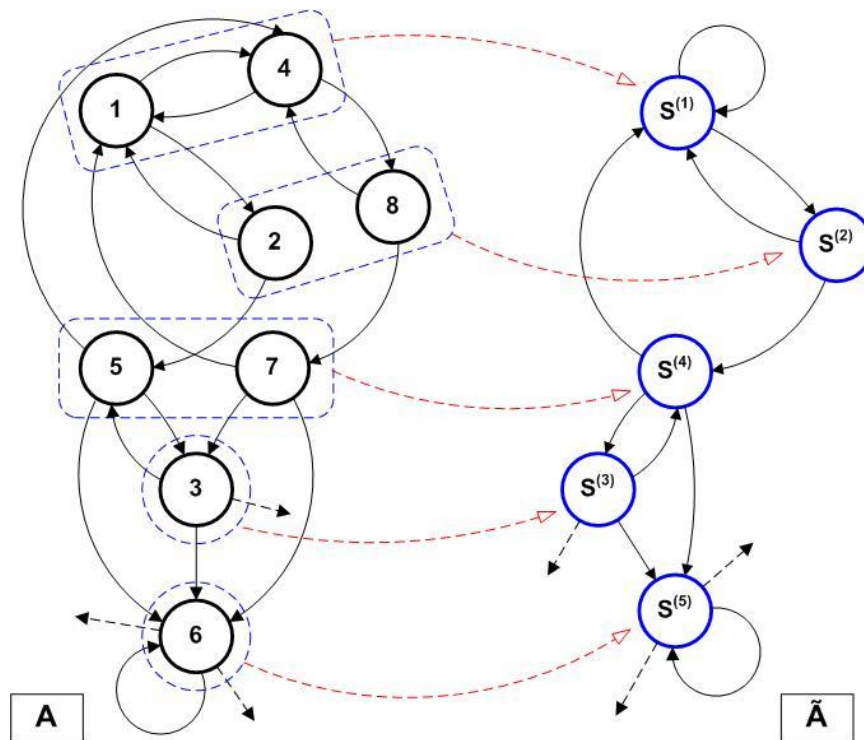


Рис. 3.9. Фрагменты графов исходного и минимального автоматов

### Контрольные вопросы

1. Что такое [эквивалентные состояния](#) конечного автомата? Каковы их свойства?
2. Что такое [k-эквивалентное разбиение](#) конечного автомата?
3. В чем отличие [эквивалентного разбиения](#) автомата от его k-эквивалентного разбиения?
4. Что называют [минимальной формой автомата](#)?
5. В чем состоит итерационный [алгоритм Мили отыскания минимальной формы автомата](#)?

### Задания для самопроверки

1. Задан автомат, имеющий:  $|S| = 6$ ,  $|X| = 3$ ,  $|Y| = 2$ . Постройте таблицу переходов/выходов автомата при условии, что в множестве его состояний имеется пара состояний 2-эквивалентных

### [Оглавление](#)

и пара состояний 2-различимых. Сколько всего можно построить автоматов, удовлетворяющих данному условию?

2. Получите минимальную форму автомата, заданного таблицей переходов/выходов. Сопоставьте графы исходного и минимального автоматов.

*Варианты:*

а)

		y(t)		s(t+1)	
		$\gamma$	$\delta$	$\gamma$	$\delta$
s(t)	x(t)				
0		u	w	0	3
1		u	u	3	5
2		w	u	1	4
3		u	w	2	5
4		w	u	1	4
5		w	w	0	2

б)

		y(t)		s(t+1)	
		$\omega$	$\varepsilon$	$\omega$	$\varepsilon$
s(t)	x(t)				
1		0	1	3	5
2		1	1	1	3
3		1	0	2	4
4		1	0	6	1
5		0	1	3	5
6		1	0	2	4

## ГЛАВА 4. СТРУКТУРНЫЙ СИНТЕЗ КОНЕЧНОГО АВТОМАТА

### 4.1. Элементарные автоматы

**Элементарный автомат (ЭА)** – это автомат Мура, имеющий два устойчивых состояния при полной таблице переходов/выходов. Другое название ЭА – **триггер** (от англ. *trigger* – защелка, спусковое устройство; в технической литературе *flip-flop* – хлопанье, «открыто – закрыто»). Триггер предназначен для хранения («защелкивания») одного двоичного разряда, соответствующего логическому нулю или логической единице. Он может сколь угодно долго находиться в одном из двух устойчивых состояний и под воздействием входного сигнала скачкообразно переключаться из одного состояния в другое. Таким образом, триггер – это элемент памяти на один двоичный разряд. Реже встречаются триггеры с более чем двумя устойчивыми состояниями, например, тристабильные или триггеры с многими состояниями.

Подробно триггеры изучаются в дисциплине «Схемотехника», поэтому ограничимся кратким рассмотрением основных типов одноступенчатых триггеров с позиции теории автоматов. Предварительно отметим, что триггеры бывают асинхронные и синхронные, в случае последних переключение триггера из одного состояния в другое возможно только при разрешающем действии синхросигнала. Синхросигнал – это тактовые импульсы обычно меандрового типа, причем срабатывание триггера возможно как по фронту (срезу) импульса, так и по уровню сигнала (высокий уровень – низкий уровень). В первом случае имеем триггер с динамическим управлением, во втором – триггер со статическим управлением.

Выходной сигнал триггера как элементарного автомата определяется исключительно состоянием, в котором он находится, то есть функция выходов ЭА есть функция от состояния. Входной сигнал влияет на переход (переключение) триггера в момент времени  $(t+1)$  из одного состояния в другое в зависимости от состояния в момент времени  $t$ . Из сказанного понятно, почему триггер реализует модель автомата Мура.

Для каждого триггера приведем стандартное обозначение, принятое на функциональных логических схемах, таблицу переходов/выходов и граф переходов.

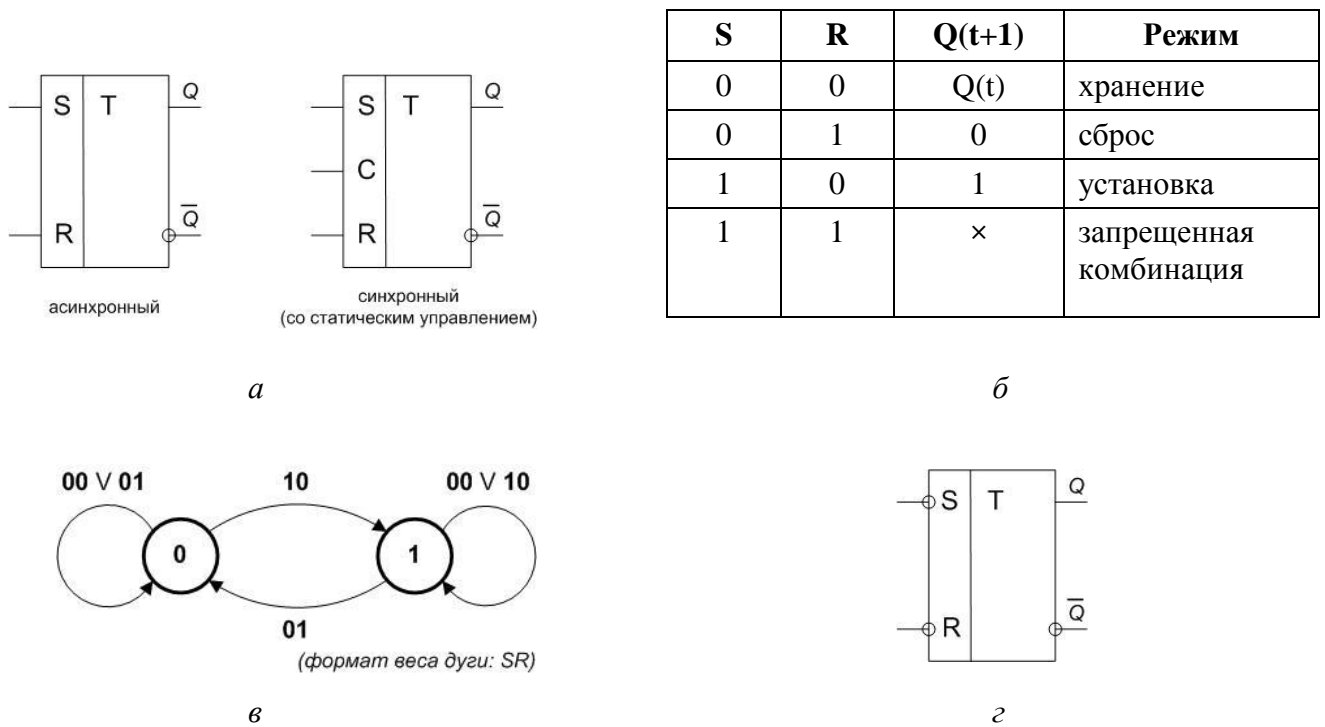
#### **RS-триггер.**

Наименование: от англ. *reset* – сброс и *set* – установка.

Обозначения асинхронного и синхронного RS-триггера показаны на рис. 4.1, а. Таблица переходов/выходов и граф переходов асинхронного RS-триггера приведены на рис. 4.1 б, в.

#### [Оглавление](#)





**Рис. 4.1.** RS-триггер:

*а* – обозначение на схемах; *б* – таблица переходов/выходов;

*в* – граф переходов; *г* – асинхронный с инверсными входами

О состоянии триггера (хранении логического нуля или логической единицы) свидетельствует непосредственно выходной сигнал  $Q$ :

$Q = 0$ , если триггер находится в состоянии «0» (хранение логического нуля);

$Q = 1$ , если триггер находится в состоянии «1» (хранение логической единицы).

Помимо прямого выхода  $Q$  триггер имеет инверсный выход  $\bar{Q}$ .

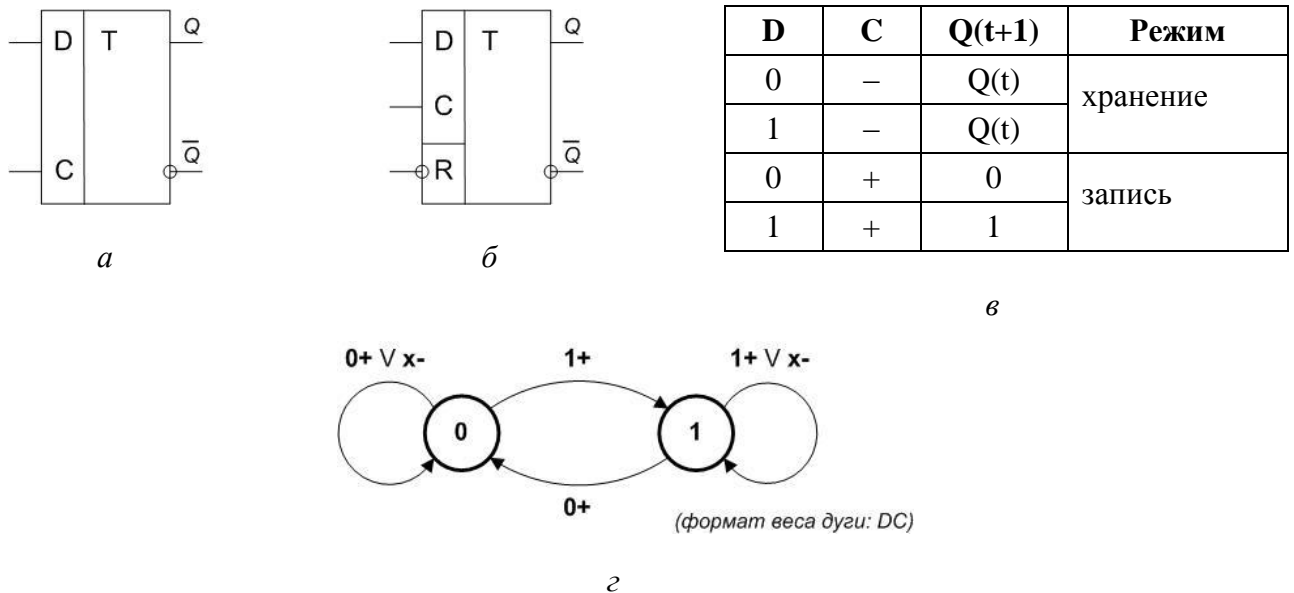
На графе переходов (рис. 4.1, *в*) видно, что триггер сохраняет состояние «0» при  $SR = 00$  (режим хранения) или  $SR = 01$  (режим сброса), а состояние «1» – в режиме хранения или в случае  $SR = 10$  (режим установки). Переключение триггера из состояния «0» в состояние «1» происходит под воздействием входного сигнала  $SR = 10$  и, соответственно, переключение из «1» в «0» выполняется при  $SR = 01$ . Входная комбинация  $SR = 11$  запрещена, так как при одновременном поступлении на входы  $S$  и  $R$  логической единицы состояние триггера оказывается неопределенным, что объясняется его схемотехническим построением.

Возможен вариант RS-триггера с инверсными входами  $S$  и  $R$  (см. рис. 4.1, *г*): активным, то есть приводящим к срабатыванию триггера входным сигналом является не логическая единица, а логический ноль. Асинхронный RS-триггер является основой схем триггеров всех остальных типов.

### [Оглавление](#)

### Д-триггер.

Имеет смысл рассматривать в синхронном варианте. При разрешающем значении синхросигнала  $C$  данные с  $D$ -входа записываются в триггер, в противном случае сохраняется предыдущее состояние триггера. Обозначение, таблица переходов/выходов и граф переходов представлены на рис. 4.2 *а, в, г*. Разрешающее значение синхросигнала в таблице и на графе показано символом «+», его отсутствие – символом «-». Произвольное значение сигнала на входе  $D$  обозначено «х».



**Рис. 4.2.** Синхронный D-триггер:

*а* – обозначение; *б* – с асинхронным входом сброса; *в* – таблица переходов/выходов;  
*г* – граф переходов

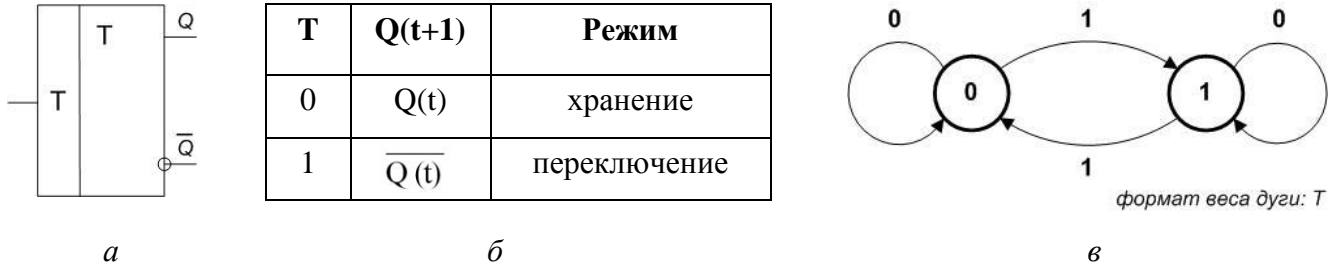
Существуют синхронные триггеры (не только D-типа) с отдельным асинхронным  $R$ -входом сброса и (или) асинхронным  $S$ -входом установки, часто инверсными. Эти дополнительные входы используют для перевода триггера в начальный момент времени в состояние хранения логического нуля либо логической единицы. Пример обозначения подобного D-триггера см. на рис. 4.2, *б*.

### Т-триггер.

Особенностью триггера данного типа является работа в так называемом «счетном» режиме: каждый раз при поступлении активного сигнала на вход  $T$  автомат переключается в состояние, противоположное текущему:  $s'(t + 1) = \overline{s'(t)}$ .

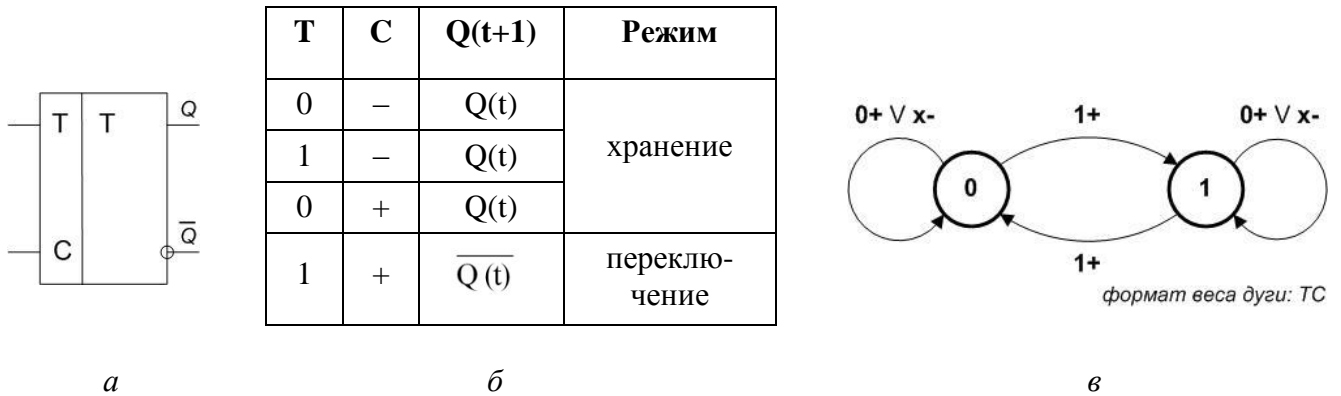
### [Оглавление](#)

Триггер строится по асинхронной и синхронной схемам. В последнем случае срабатывание возможно только при разрешающем значении синхросигнала. На рис. 4.3 представлены обозначение, таблица переходов/выходов и граф переходов асинхронного Т-триггера; на рис. 4.4 – то же для синхронного Т-триггера.



**Рис. 4.3.** Асинхронный Т-триггер:

*a* – обозначение; *b* – таблица переходов/выходов; *в* – граф переходов



**Рис. 4.4.** Синхронный Т-триггер:

*a* – обозначение; *b* – таблица переходов/выходов; *в* – граф переходов

### JK-триггер.

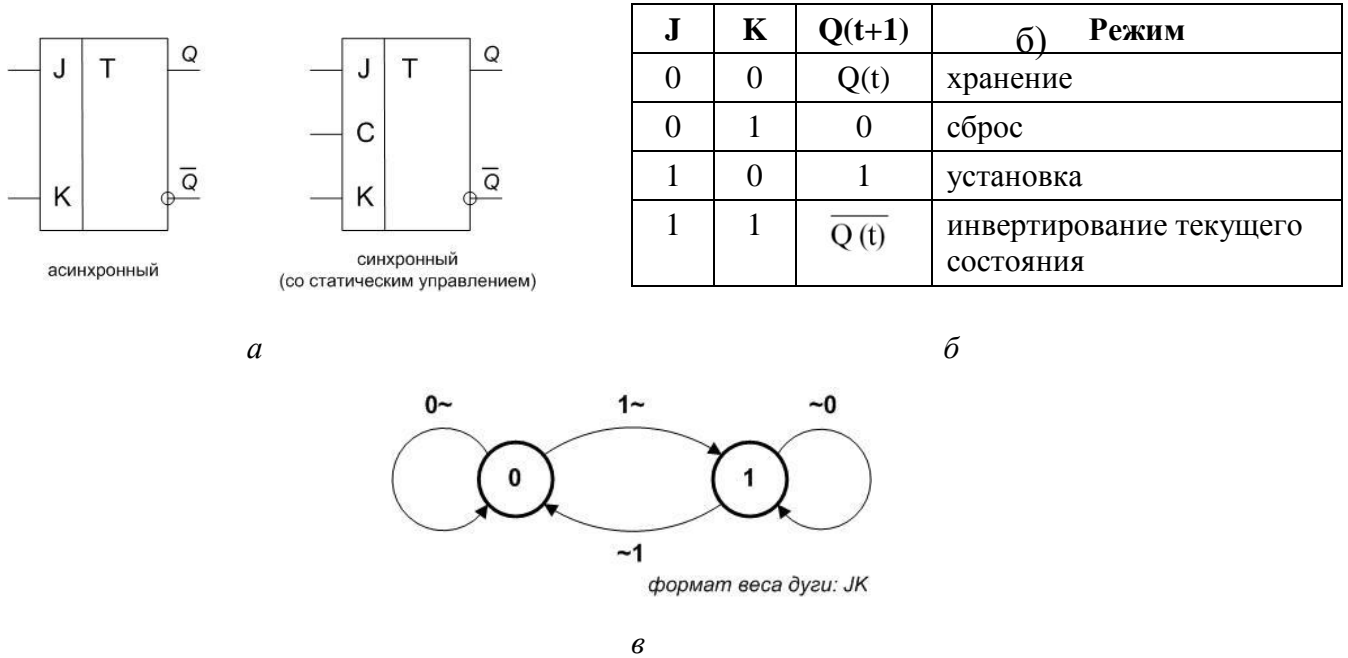
Наименование: от англ. *jump* – скачок, перепад и *kill* – подавлять, уничтожать. Обозначение, таблица переходов/выходов и граф переходов показаны на рис. 4.5.

Знак «~» на графе переходов означает, что соответствующий входной сигнал может иметь произвольное значение: или логический ноль, или логическая единица.

Алгоритм работы JK-триггера такой же, как у RS-триггера, за исключением входной комбинации JK = 11. Если для RS-триггера аналогичный набор входных сигналов является запрещенным, то в случае JK-триггера одновременное поступление двух единиц на информационные входы приводит к переключению автомата в состояние, противоположное

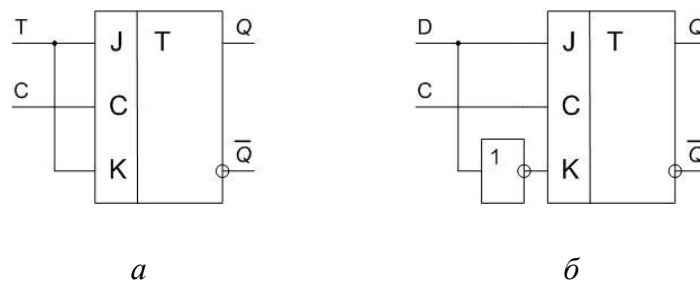
[Оглавление](#)

текущему. Отсюда следует, что JK-триггер может работать в режиме Т-триггера (см. рис. 4.6, а), и становится понятным, почему возможно произвольное значение одного из входных сигналов на каждой дуге графа переходов. Действительно, переход из состояния «0» в состояние «1» возможен как в случае JK = 10 (установка), так и в случае JK = 11 (переключение в противоположное состояние).



**Рис. 4.5.** JK-триггер:  
 а – обозначение; б – таблица переходов/выходов; в – граф переходов

Помимо режима Т-триггера возможно использование данного ЭА в режиме D-триггера (см. рис. 4.6, б). Действительно, при D = 0 в момент времени t имеем JK = 01 (режим сброса) и получаем Q(t+1) = 0, при D = 1 имеем JK = 10 (режим установки) и Q(t+1) = 1.



**Рис. 4.6.** JK-триггер в режиме работы:  
 а – Т-триггера; б – D-триггера

[Оглавление](#)

Информацию о переключениях рассмотренных триггеров под воздействием входных сигналов можно свести в общую таблицу (рис. 4.7).

Q(t)	Q(t+1)	RS-триггер		D-триггер	T-триггер	JK-триггер	
		S(t)	R(t)	D(t)	T(t)	J(t)	K(t)
0	0	0	~	0	0	0	~
0	1	1	0	1	1	1	~
1	0	0	1	0	1	~	1
1	1	~	0	1	0	~	0

**Рис. 4.7.** Сводная таблица переходов/выходов триггеров

Как было отмечено выше, присутствие знака «~» говорит о допуске значения как логического нуля, так и логической единицы. Читателю рекомендуется сопоставить приведенную таблицу с собственными таблицами переходов/выходов триггеров и охарактеризовать режимы их работы.

## 4.2. Алгоритм структурного синтеза

Исходные данные:

1) математическая модель конечного автомата Мили или Мура в виде взвешенного орграфа переходов либо таблицы переходов/выходов, полученная в результате абстрактного синтеза КА;

2) набор элементарных автоматов;

3) набор логических элементов, образующих базис и реализующих функционально полную систему булевых функций.

Требуется построить функциональную логическую схему автомата в результате поэтапного выполнения алгоритма структурного синтеза КА.

Алгоритм структурного синтеза рассмотрим на конкретном **примере**.

Пусть автомат Мили задан графом переходов (см. рис. 4.8). Система счисления: двоичная. Элементарные автоматы: синхронные RS-триггеры, синхронные D-триггеры.

### [Оглавление](#)

Логический базис: конъюнкция, дизъюнкция, отрицание<sup>\*)</sup>.

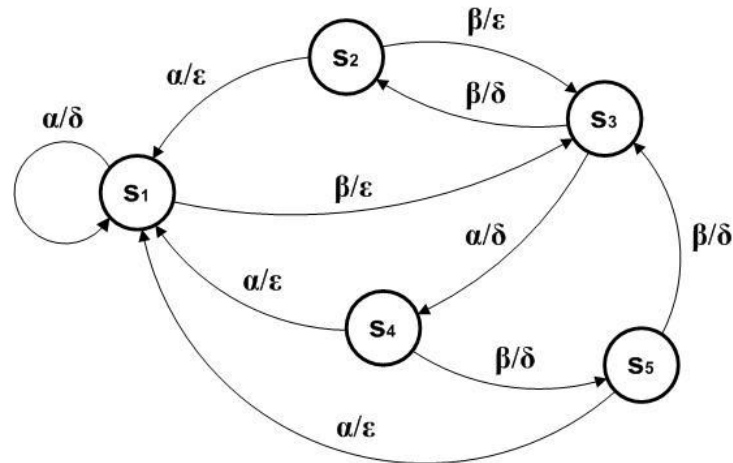


Рис. 4.8. Взвешенный орграф переходов заданного КА

Имеем:  $X = \{\alpha, \beta\}$ ;  $Y = \{\delta, \epsilon\}$ ;  $S = \{s_1, s_2, s_3, s_4, s_5\}$

**Этап 1. Кодирование входного алфавита.** Присваиваем логические значения элементам множества  $X$ . Пусть  $\alpha = 0$ ,  $\beta = 1$ .

**Этап 2. Кодирование выходного алфавита.** Аналогично поступаем с элементами множества  $Y$ . Пусть  $\delta = 0$ ,  $\epsilon = 1$ .

Примечание. Так как условием задачи оговорена двоичная система счисления, и множество как входных, так и выходных сигналов равно  $\{0, 1\}$ , будем считать, что вход и выход автомата одноразрядные и представлены булевыми переменными –  $x$  и  $y$  соответственно.

**Этап 3. Кодирование состояний автомата.**

1). Определяем *разрядность кода*  $L$  – минимально необходимое количество элементарных автоматов или, иначе говоря, разрядность памяти автомата. Так как система счисления двоичная, то

$$L = \lceil \log_2 |S| \rceil,$$

где  $|S|$  – мощность множества состояний,  $\lceil \dots \rceil$  – ближайшее целое число сверху.

<sup>\*)</sup> В схемотехнической практике проектирования цифровых автоматов требования к использованию элементов памяти и комбинационной логики оговариваются значительно строже, исходя из технических характеристик применяемой интегральной элементной базы. Например, логический базис может быть задан как 2И-НЕ, либо 3И-2ИЛИ-НЕ, либо 3ИЛИ-НЕ и тому подобное. В теории автоматов принципиальное значение имеет функциональная полнота системы булевых функций, составляющих базис (см. раздел 4.4).

[Оглавление](#)

Поскольку в данном примере  $|S| = 5$ , то  $L = 3$ . Линейка памяти автомата трехразрядная, то есть память строится на трех триггерах.

2). Определяем *двоичный код каждого состояния* автомата. Существует множество алгоритмов кодирования состояний, например, противоночное, экономичное, соседнее кодирование и ряд других. В случае противоночного кодирования стремятся к тому, чтобы в каждый дискретный момент времени, по возможности, переключался только один разряд. Это позволяет избежать ложных срабатываний элементов памяти из-за гонок (критических состязаний) в комбинационной схеме. Такой способ актуален для асинхронных схем. Обоснование эффективности противоночного кодирования относится к давним разработкам – периода элементной базы малой степени интеграции. В современных СБИС время срабатывания триггера мало отличается от времени задержки распространения сигнала в логическом элементе и большее значение имеют задержки в линиях связи (слоях металлизации на кристалле), значительно выгоднее проектировать синхронный автомат, не заботясь о противоночном кодировании, требующем дополнительных аппаратных затрат.

Воспользуемся одним из возможных алгоритмов кодирования, сочетающим достоинства экономичного, соседнего и противоночного методов.

*Шаг 1.* Нулевой код сопоставляем состоянию, вершина которого имеет наибольшую полустепень захода, то есть среди всех вершин графа ей инцидентно наибольшее количество заходящих дуг.

*Шаг 2* – итерационный. Находим состояние, наиболее связанное с ранее закодированными. Назовем это состояние текущим. Из всех свободных кодовых комбинаций выбираем ту, для которой минимальна сумма кодовых расстояний<sup>\*)</sup> между ней и существующими кодами состояний, смежных текущему. Кратность связей учитываем введением поправочного коэффициента  $k_{св}$ . Например, если две вершины графа связаны парой дуг, то  $k_{св} = 2$ . Очевидно, число итераций шага 2 равно  $|S| - 1$ .

Рассмотрим применение сформулированного алгоритма к кодированию состояний данного конкретного автомата.

*Шаг 1.* Наибольшее количество входящих дуг на графе переходов инцидентно вершине  $s_1$ . (полустепени захода:  $p^+(s_1) = 4$ ,  $p^+(s_3) = 3$ ,  $p^+(s_2) = p^+(s_4) = p^+(s_5) = 1$ ). Состоянию  $s_1$  усваиваем нулевой код:  $s_1 = \mathbf{000}$ .

---

<sup>\*)</sup> Кодовым расстоянием  $d$  между кодовыми комбинациями  $i$  и  $j$  одинаковой разрядности называется число различных разрядов в  $i$  и  $j$ . Например,  $i = 1010$ ,  $j = 1100$ , следовательно,  $d_{ij} = 2$ .

*Шаг 2. Итерация 2.1.* Вершины  $s_2, s_3, s_4, s_5$  связаны с вершиной  $s_1$  каждая одной дугой. Не имеет значения, какое из состояний предпочесть. Выбираем  $s_2$ . Очевидно, код  $s_2$  должен содержать единицу в любом разряде, обеспечивая кодовое расстояние  $d_{21} = 1$ . Пусть  $s_2 = \mathbf{001}$ .

*Итерация 2.2.* С подмножеством вершин  $\{s_1, s_2\}$  наиболее связана вершина  $s_3$  (тремя дугами:  $s_1 \rightarrow s_3, s_2 \rightarrow s_3, s_3 \rightarrow s_2$ ); вершины  $s_4$  и  $s_5$  связаны с  $\{s_1, s_2\}$  одной дугой каждая. Следовательно, очередным кодируемым состоянием должно быть  $s_3$ . Рассчитываем кодовые расстояния:

$s_3$	Свободные комбинации	$d_{31} + 2d_{32}$
	010	$1 + 2 \cdot 2 = 5$
	<b>011</b>	$2 + 2 \cdot 1 = 4$
	100	$1 + 2 \cdot 2 = 5$
	<b>101</b>	$2 + 2 \cdot 1 = 4$
	110	$2 + 2 \cdot 3 = 8$
	111	$3 + 2 \cdot 2 = 7$

Поскольку вершины  $s_2$  и  $s_3$  связывают две дуги,  $d_{32}$  входит в сумму с поправочным коэффициентом  $k_{св} = 2$ . Наименьшим значением суммы кодовых расстояний обладают комбинации 011 и 101. Предпочтение можно отдать любой из них. Пусть  $s_3 = \mathbf{011}$ .

*Итерация 2.3.* Подмножество закодированных вершин:  $\{s_1, s_2, s_3\}$ . Число связей для  $s_4$  – две, для  $s_5$  – две; выбор  $s_4$  или  $s_5$  равновероятный. Кодируем состояние  $s_4$ :

$s_4$	Свободные комбинации	$d_{41} + d_{43}$
	<b>010</b>	$1 + 1 = 2$
	100	$1 + 3 = 4$
	101	$2 + 2 = 4$
	110	$2 + 2 = 4$
	111	$3 + 1 = 4$

По наименьшему значению суммы кодовых расстояний определяем:  $s_4 = \mathbf{010}$ .

*Итерация 2.4.* Незакодированным осталось состояние  $s_5$ :

$s_5$	Свободные комбинации	$d_{51} + d_{53} + d_{54}$
	100	$1 + 3 + 2 = 6$
	101	$2 + 2 + 3 = 7$
	<b>110</b>	$2 + 2 + 1 = 5$
	111	$3 + 1 + 2 = 6$

### [Оглавление](#)





$$Q_2: 0 \rightarrow 0, S_2R_2 = 0\sim; \quad Q_1: 1 \rightarrow 0, S_1R_1 = 01; \quad Q_0: 0 \rightarrow 0, D_0 = 0.$$

Фрагмент таблицы, отвечающий рассмотренному случаю, показан на рис. 4.10.

Время: t		t+1				t (входы триггеров)								
x	s <sub>i</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	y	s <sub>j</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	S <sub>2</sub>	R <sub>2</sub>	S <sub>1</sub>	R <sub>1</sub>	D <sub>0</sub>
0	•													
	•													
	•													
	4	0	1	0	1	1	0	0	0	0	~	0	1	0
	•													
	•													
	•													

**Рис. 4.10.** Фрагмент кодированной таблицы переходов/выходов автомата для перехода  $s_4 \rightarrow s_1$  при  $x = 0$

Таким образом, строка таблицы соответствует конкретному переходу в автомате при выработке значения выходного сигнала, заданного алгоритмом функционирования КА. Количество строк полной таблицы такого формата, очевидно, равно числу всех переходов в автомате, то есть мощности множества дуг графа переходов. Однако в таблицу целесообразно включить и неиспользуемые кодовые комбинации, что может оказаться полезным при получении минимальных форм функций  $f_y$  и  $f_s$  на следующем этапе синтеза.

Полная двоичная кодированная таблица переходов/выходов автомата приведена на рис. 4.11.

Строки таблицы упорядочены по возрастанию кодов состояний для каждого значения входного сигнала. Для удобства чтения столбцы таблицы, относящиеся к одному и тому же разряду памяти, выделены одинаковым цветом. Для справки в таблицу добавлены столбцы, соответствующие триггерам других типов: JK-триггеру для разряда  $Q_1$  и T-триггеру для разряда  $Q_0$ . Знаком «~» обозначены как произвольные значения логических переменных, так и позиции, несуществующие в данном КА.

### Этап 5. Построение и минимизация функции выхода.

Областью определения функции выхода автомата Мили являются множество входных сигналов и множество состояний автомата. Поэтому для данного автомата выходной сигнал  $y$  выражается булевой функцией четырех аргументов:

#### [Оглавление](#)

$$y = f_y(x, Q_2, Q_1, Q_0).$$

Пользуясь аппаратом алгебры логики, можно представить функцию выхода в одной из совершенных нормальных форм – конъюнктивной или дизъюнктивной (СКНФ или СДНФ).

Время: t		t+1				t (входы триггеров)					Входы триггеров других типов <sup>*)</sup>						
x	S <sub>i</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	y	S <sub>j</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	S <sub>2</sub>	R <sub>2</sub>	S <sub>1</sub>	R <sub>1</sub>	D <sub>0</sub>	J <sub>1</sub>	K <sub>1</sub>	T <sub>0</sub>
0	1	0	0	0	0	1	0	0	0	0	~	0	~	0	0	~	0
	2	0	0	1	1	1	0	0	0	0	~	0	~	0	0	~	1
	4	0	1	0	1	1	0	0	0	0	~	0	1	0	~	1	0
	3	0	1	1	0	4	0	1	0	0	~	~	0	0	~	0	1
	~	1	0	0	~	~	~	~	~	~	~	~	~	~	~	~	~
	~	1	0	1	~	~	~	~	~	~	~	~	~	~	~	~	~
	5	1	1	0	1	1	0	0	0	0	1	0	1	0	~	1	0
	~	1	1	1	~	~	~	~	~	~	~	~	~	~	~	~	~
1	1	0	0	0	1	3	0	1	1	0	~	1	0	1	1	~	1
	2	0	0	1	1	3	0	1	1	0	~	1	0	1	1	~	0
	4	0	1	0	0	5	1	1	0	1	0	~	0	0	~	0	0
	3	0	1	1	0	2	0	0	1	0	~	0	1	1	~	1	0
	~	1	0	0	~	~	~	~	~	~	~	~	~	~	~	~	~
	~	1	0	1	~	~	~	~	~	~	~	~	~	~	~	~	~
	5	1	1	0	0	3	0	1	1	0	1	~	0	1	~	0	1
	~	1	1	1	~	~	~	~	~	~	~	~	~	~	~	~	~

<sup>\*)</sup> Для справки.

**Рис. 4.11.** Полная кодированная таблица переходов/выходов автомата

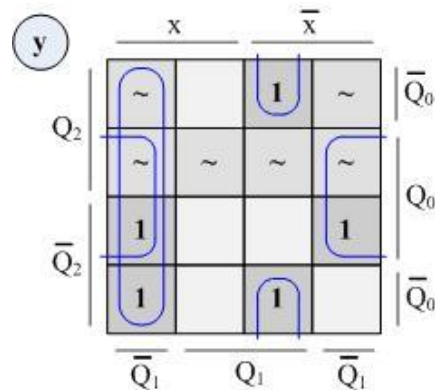
Таблицей истинности фактически является построенная таблица переходов/выходов (рис. 4.11). Количество нулевых и единичных значений функции  $y$  в данном случае одинаково, поэтому предпочтение можно отдать СДНФ, учитывая доступность традиционных методов минимизации булевой функции, заданной в СДНФ. Получим:

$$y = \overline{x} \overline{Q_2} \overline{Q_1} Q_0 \vee \overline{x} \overline{Q_2} Q_1 \overline{Q_0} \vee \overline{x} Q_2 Q_1 \overline{Q_0} \vee x \overline{Q_2} \overline{Q_1} \overline{Q_0} \vee x \overline{Q_2} \overline{Q_1} Q_0.$$

Таким образом,  $f_y$  представлена в виде дизъюнкции пяти импликант, каждая ранга 4. Минимизацию функции целесообразно выполнить методом диаграмм Вейча (карт Карно): см. рис. 4.12.

## [Оглавление](#)

Значения  $y$ , отмеченные знаком « $\sim$ » в таблице переходов/выходов, при необходимости могут быть доопределены до логической единицы и задействованы в процессе минимизации. Такое доопределение никак не влияет на картину истинностных значений функции.



**Рис. 4.12.** Диаграмма Вейча для функции выхода

Причина в том, что двоичные наборы аргументов, соответствующие  $y = \langle \sim \rangle$ , в процессе функционирования автомата не формируются. Иными словами, в данном автомате таких двоичных комбинаций не существует. Участие же соответствующих им импликант в минимизации дает более сокращенный вид ДНФ, что, в свою очередь, приводит к экономии аппаратных средств при покрытии функции логическими элементами. На диаграмме Вейча ячейки для этих импликант содержат тот же знак « $\sim$ ».

В итоге минимальная ДНФ функции выхода имеет вид:

$$y_{\min} = x \bar{Q}_1 \vee \bar{Q}_1 Q_0 \vee \bar{x} Q_1 \bar{Q}_0.$$

### Этап 6. Получение и минимизация функций возбуждения ЭА.

Совокупность функций возбуждения триггеров реализует функцию переходов автомата  $f_s$ . Поскольку  $s(t+1) = f_s(x, s(t))$ , то в общем случае каждая функция возбуждения  $f_{tr}^i$  есть функция двух множественных аргументов:

$$f_{tr}^i = f^i(\{x_1, x_2, \dots, x_p\}, \{Q_{L-1}, Q_{L-2}, \dots, Q_0\}), \quad i = \overline{1, k},$$

где  $k$  – суммарное число информационных входов триггеров, составляющих линейку памяти автомата,  $L$  – ее разрядность,  $p$  – число входных переменных.

В данном случае  $k = 5$ ,  $L = 3$ ,  $p = 1$ . Для удобства обозначим каждую из функций возбуждения так же, как обозначен соответствующий вход триггера:  $S_2, R_2, S_1, R_1, D_0$ . Запишем СДНФ для каждой функции согласно таблице переходов/выходов (см. рис. 4.11):

### [Оглавление](#)

$$S_2 = x\bar{Q}_2Q_1\bar{Q}_0;$$

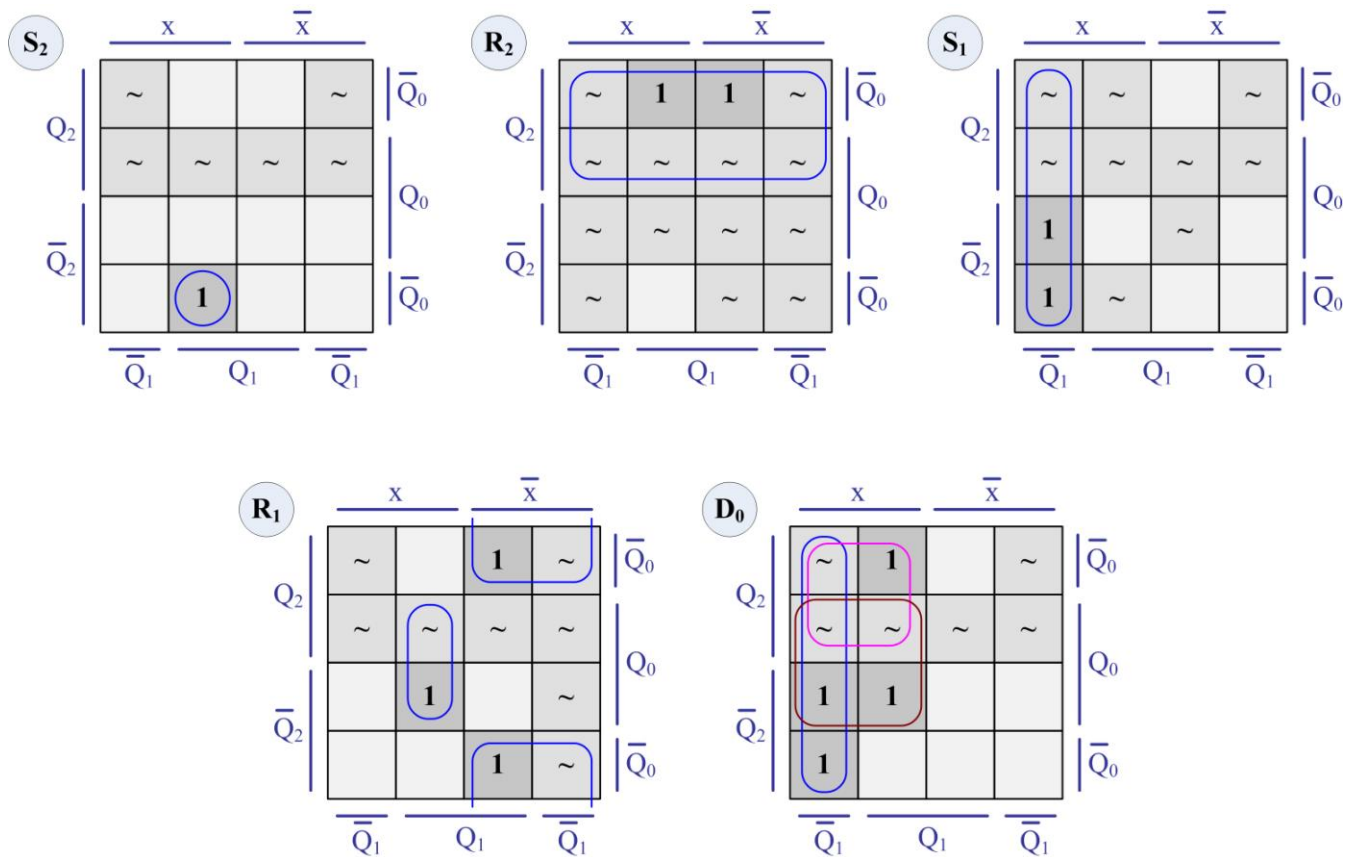
$$R_2 = \bar{x}Q_2Q_1\bar{Q}_0 \vee xQ_2Q_1\bar{Q}_0;$$

$$S_1 = x\bar{Q}_2\bar{Q}_1\bar{Q}_0 \vee x\bar{Q}_2\bar{Q}_1Q_0;$$

$$R_1 = \bar{x}\bar{Q}_2Q_1\bar{Q}_0 \vee \bar{x}Q_2Q_1\bar{Q}_0 \vee x\bar{Q}_2Q_1Q_0;$$

$$D_0 = x\bar{Q}_2\bar{Q}_1\bar{Q}_0 \vee x\bar{Q}_2\bar{Q}_1Q_0 \vee x\bar{Q}_2Q_1\bar{Q}_0 \vee xQ_2Q_1\bar{Q}_0.$$

Приведем указанные функции к минимальным формам с учетом произвольных и несуществующих значений, отмеченных в таблице знаком «~». Соответствующие диаграммы Вейча показаны на рис. 4.13.



**Рис. 4.13.** Минимизация функций возбуждения триггеров автомата

В итоге минимизации имеем:  $S_{2\min} = x\bar{Q}_2Q_1\bar{Q}_0;$

$$R_{2\min} = Q_2;$$

$$S_{1\min} = x\bar{Q}_1;$$

$$R_{1\min} = \bar{x}\bar{Q}_0 \vee xQ_1Q_0;$$

$$D_{0\min} = xQ_2 \vee x\bar{Q}_1 \vee xQ_0.$$

### [Оглавление](#)



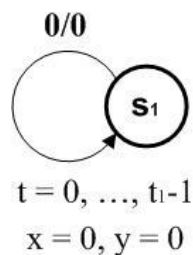
Примечание. Правила построения функциональных схем цифровой электроники изучаются в курсе «Схемотехника». Полученные знания и навыки закрепляются на практике при выполнении курсовой работы по названной учебной дисциплине.

### 4.3. Тестирование автомата

Цель тестирования состоит в выявлении логических ошибок, возможно допущенных в процессе структурного синтеза, и установлении соответствия полученной функциональной логической схемы исходной модели автомата. Тестирование логики схемы требует прохождения по всем ветвям алгоритма работы автомата. Тестовые последовательности входных сигналов должны подбираться таким образом, чтобы задействовать все возможные переходы в автомате на множестве состояний. Одновременно контролируется корректность выдачи выходных сигналов.

Формирование тестовых последовательностей – особая и, вообще говоря, нетривиальная задача. В данном случае ограничимся изучением принципа тестирования на конкретном тестовом наборе применительно к автомату, синтезированному в разделе 4.2.

Пусть в начальном состоянии  $s = s_1$  (000),  $x = 0$ , при этом  $y = 0$ . Если входной сигнал не меняется, то автомат сколь угодно долго хранит начальное состояние и на выходе остается нулевой сигнал. Ситуация отражена в первой строке таблицы тестирования на рис. 4.16 и соответствует петле, инцидентной вершине  $s_1$  графа переходов, фрагмент которого показан на рис. 4.15. При этом в каждый момент времени, пока  $x = 0$ , в автомате формально совершается переход  $s_1 \rightarrow s_1$  (триггер разряда  $Q_2$  – в режиме хранения, триггер разряда  $Q_1$  – в режиме сброса, триггер разряда  $Q_0$  – в режиме записи нуля).



**Рис. 4.15.** КА в начальном состоянии

Таблица тестирования (рис. 4.16) для удобства разбита на две части (обозначены I ч. и II ч.), связанные по столбцу «№ п/п» в отношении 1:1. В первой, основной части содержатся сведения

[Оглавление](#)

о входном сигнале, состоянии автомата и выходном сигнале в каждый дискретный момент времени  $t$ .

I ч.

№ п/п	Время	Вход x	Состояние				Выход y
			$s_i$	$Q_2$	$Q_1$	$Q_0$	
1	$0 \dots t_1-1$	<b>0</b>	$s_1$	0	0	0	<b>0</b>
2	$t_1$	<b>1</b>	$s_1$	0	0	0	<b>1</b>
3	$t_2$	<b>1</b>	$s_3$	0	1	1	<b>0</b>
4	$t_3$	<b>0</b>	$s_2$	0	0	1	<b>1</b>
5	$t_4$	<b>1</b>	$s_1$	0	0	0	<b>1</b>

II ч.

№ п/п	Линии шины								Контрольные точки							Входы триггеров				
	1	2	3	4	5	6	7	8	a	b	c	d	e	f	g	$S_2$	$R_2$	$S_1$	$R_1$	$D_0$
1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0
2	1	0	0	1	0	1	0	1	1	0	0	0	0	1	0	0	0	1	0	1
3	1	0	0	1	1	0	1	0	1	0	1	1	0	0	0	0	0	0	1	1
4	0	1	0	1	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0
5	1	0	0	1	0	1	0	1	1	0	0	0	0	1	0	0	0	1	0	1

**Рис. 4.16.** Таблица тестирования КА входной последовательностью (0...01101)

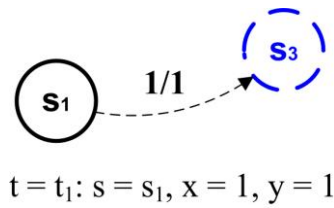
Вторая, вспомогательная часть отражает, что происходит в схеме в каждый момент времени: каково состояние шины, выходов логических элементов комбинационной части схемы (контрольные точки, обозначенные на схеме буквами от а до g), от которых зависят значения функций возбуждения триггеров и функции выхода, а также каковы значения самих функций возбуждения, то есть какие логические значения подготавливаются комбинационной схемой и подаются на входы триггеров для переключения автомата в следующее состояние в  $(t+1)$ -й момент времени.

Пусть на вход  $x$  автомата в последовательном коде подается тестовый набор (**1101**).

1).  $t = t_1$ . Автомат в состоянии  $s_1$ . Поступает входной сигнал  $x = 1$ . Состояния линий шины и сигналы в контрольных точках отражены в строке № 2 таблицы тестирования. На выходе формируется сигнал  $y = 1$ , что вызвано переходом сигнала в точке  $f$  от 0 к 1. Входы триггеров подготовлены к переключению автомата  $s_1$  (**000**)  $\rightarrow$   $s_3$  (**011**). Ситуация показана на рис. 4.17.

### [Оглавление](#)



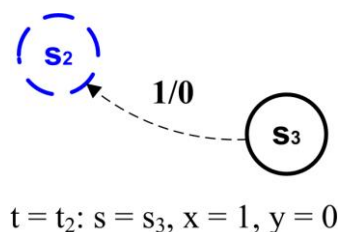


**Рис. 4.17.** Поведение КА в момент времени  $t_1$

Переключение произойдет при поступлении очередного синхроимпульса, то есть в момент времени  $t_2$ . Режимы срабатывания триггеров будут следующими: разряд  $Q_2$  – хранение, разряд  $Q_1$  – установка, разряд  $Q_0$  – запись единицы. Заметим, что заданное алгоритмом значение сигнала  $y$  остается на линии выхода до тех пор, пока автомат сохраняет текущее состояние и действует соответствующий входной сигнал. Таким образом, входной сигнал в текущий момент времени определяет: а) значение выходного сигнала для текущего состояния и б) в какое состояние из текущего перейдет автомат в следующий момент времени. Это соответствует уравнениям модели автомата Мили для  $y(t)$  и  $s(t+1)$ . Из сказанного также понятно, почему выходной сигнал «привязан» к переходу именно в модели Мили, хотя переход как в модели Мили, так и в модели Мура инициируется входным сигналом.

Итак, если на входе автомата Мили, сохраняющего одно и то же состояние, произойдет изменение входного сигнала, то изменится и выходной сигнал в соответствии с алгоритмом работы автомата. В автомате Мура этого не произойдет: изменение выходного сигнала возможно только после смены состояния, то есть по факту переключения триггеров линейки памяти автомата.

2).  $t = t_2$ . Автомат переходит в состояние  $s_3$ . На вход подается сигнал  $x = 1$ . Данные тестирования занесены в строку № 3 таблицы тестирования. Формируется  $y = 0$ . На входы триггеров поступают сигналы, которые обеспечат переход автомата  $s_3 (011) \rightarrow s_2 (001)$  в момент времени  $t_3$ . Ситуация отражена на рис. 4.18.



**Рис. 4.18.** Поведение КА в момент времени  $t_2$

[Оглавление](#)

3).  $t = t_3$ . Автомат переключается в состояние  $s_2$ . Режимы срабатывания триггеров: разряд  $Q_2$  – хранение, разряд  $Q_1$  – сброс, разряд  $Q_0$  – запись единицы. Поступает сигнал  $x = 0$ . В таблице тестирования этим условиям соответствует строка № 4. Значение выходного сигнала  $y$  становится равным  $1$ . На входах элементов памяти формируются сигналы для перехода автомата  $s_2 (001) \rightarrow s_1 (000)$  в следующий,  $t_4$ -й момент времени. Ситуация показана на рис. 4.19.

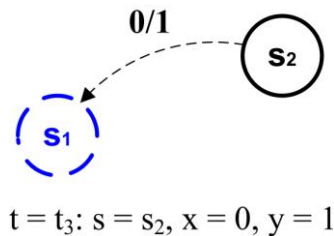


Рис. 4.19. Поведение КА в момент времени  $t_3$

4).  $t = t_4$ . Автомат переходит в состояние  $s_1$ . Срабатывание триггеров: разряды  $Q_2$  и  $Q_1$  – в режиме хранения, разряд  $Q_0$  – в режиме записи нуля. Подается сигнал  $x = 1$ . Ситуация точно такая же, как в момент времени  $t_1$ : см. рис. 4.17. Сигнал на выходе:  $y = 1$ , в таблице тестирования пятая и вторая строки полностью совпадают, в следующий момент времени ( $t_5$ ) произойдет переключение автомата  $s_1 (000) \rightarrow s_3 (011)$ .

Таким образом, в результате тестирования входной последовательностью  $0\dots01101$  в автомате выполняются следующие переходы и выдаются соответствующие значения выходного сигнала:

$$s_1 \xleftarrow[y=0]{x=0} s_1 \xrightarrow[y=1]{x=1} s_3 \xrightarrow[y=0]{x=1} s_2 \xrightarrow[y=1]{x=0} s_1 \xrightarrow[y=1]{x=1} s_3$$

$(t_1) \quad (t_2) \quad (t_3) \quad (t_4) \quad (t_5)$

Видно, что поведение автомата на данной тестовой последовательности отвечает его исходной модели. Фрагмент графа переходов, соответствующий рассмотренному тесту, показан на рис. 4.20.

Примечание. Рассмотренный пример тестирования показывает, как можно верифицировать логику, то есть алгоритм функционирования автомата. При его схемотехнической реализации и переходе к принципиальной электрической схеме необходимо принять во внимание следующее.

### [Оглавление](#)

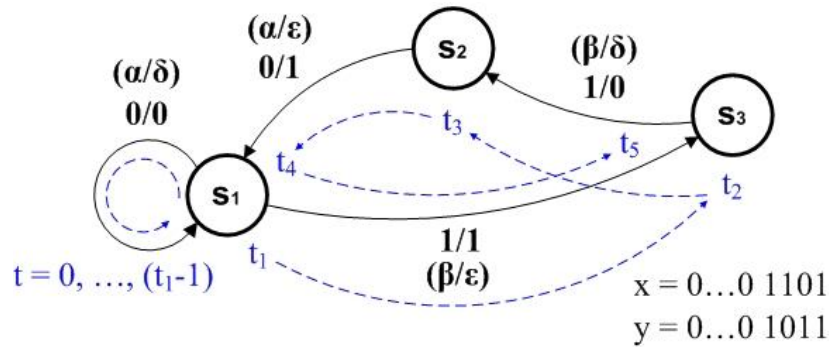


Рис. 4.20. Фрагмент графа переходов КА на заданной тестовой последовательности

Все события в автомате, то есть изменение значений входного сигнала, переключение автомата из одного состояния в другое, выдача выходного сигнала происходят в дискретные моменты времени, которые задаются синхросигналом  $C$ . Отсюда следует, что длительность разрешающего синхроимпульса (в случае статического управления) должна быть достаточной для срабатывания триггеров, но не должна превышать времени, за которое значения на входах триггеров изменятся из-за действия нового значения входного сигнала, во избежание ложных срабатываний триггеров. В случае использования элементов памяти с динамическим управлением эта проблема отпадает. Соответственно, снятие сигнала  $y$  с линии выхода должно происходить после завершения переходных процессов в схеме, то есть с некоторой задержкой относительно синхросигнала. ■

#### 4.4. Функциональная полнота системы конечных автоматов

Теория автоматов утверждает следующее принципиально важное положение – **теорему о функциональной полноте системы конечных автоматов**:

Для построения функциональной логической схемы произвольного КА необходимо и достаточно, чтобы в множестве заданных элементарных автоматов существовал хотя бы один с полной таблицей переходов/выходов, а в множестве логических элементов, реализующих комбинационную схему автомата, нашлось бы такое подмножество, которое отвечало бы функционально полной системе булевых функций.

Действительно, функциональная логическая схема автомата представляет собой совокупность комбинационной схемы и памяти. Для построения памяти автомата независимо

#### [Оглавление](#)

от ее разрядности необходимо и достаточно иметь элементарные автоматы хотя бы одного типа, если они обеспечивают хранение любых значений логических переменных в заданной системе счисления. Комбинационная схема КА представляет собой реализованную на логических элементах совокупность произвольных булевых функций. Любая булева функция может быть выражена суперпозицией элементарных функций, если они образуют функционально полную систему (так называемая *теорема о функциональной полноте системы булевых функций*, доказываемая в дискретной математике). Сказанное дает объяснение сформулированной теореме о функциональной полноте системы конечных автоматов.

Однако обосновать данное утверждение также возможно исходя из конструктивных соображений, следующих из рассмотренного алгоритма структурного синтеза конечного автомата.

### Контрольные вопросы

1. Какие автоматы называют элементарными? Каково их назначение?
2. Укажите обозначения триггеров основных типов.
3. Каковы алгоритмы функционирования RS-, D-, JK- и T-триггеров? В чем сходство и различие таблиц переходов/выходов, графов переходов указанных элементарных автоматов?
4. В алгоритмических режимах каких триггеров может функционировать JK-триггер? Сопоставьте таблицы переходов/выходов различных его режимов с таблицами переходов/выходов соответствующих триггеров.
5. Какова задача структурного синтеза автомата?
6. Из каких этапов состоит алгоритм структурного синтеза?
7. Каким образом осуществляется кодирование состояний автомата?
8. Почему возможно использование несуществующих в автомате кодовых комбинаций в процессе минимизации функции выхода и функций возбуждения элементарных автоматов?
9. На что оказывает влияние доопределение произвольных значений переменных при минимизации функции переходов и функции выходов автомата?
10. Каким образом можно осуществить тестирование автомата по его функциональной логической схеме?
11. Изменяется ли выходной сигнал автомата Мили и автомата Мура в общем случае, если
  - а) состояние автомата сохраняется, а входной сигнал меняет значение;
  - б) входной сигнал сохраняется, а автомат переключается в другое состояние?

### Оглавление

Как можно пояснить ответы на эти вопросы при помощи функциональной логической схемы автомата?

12. Сформулируйте теорему о функциональной полноте системы конечных автоматов. Обоснуйте справедливость данной теоремы.

### Задания для самопроверки

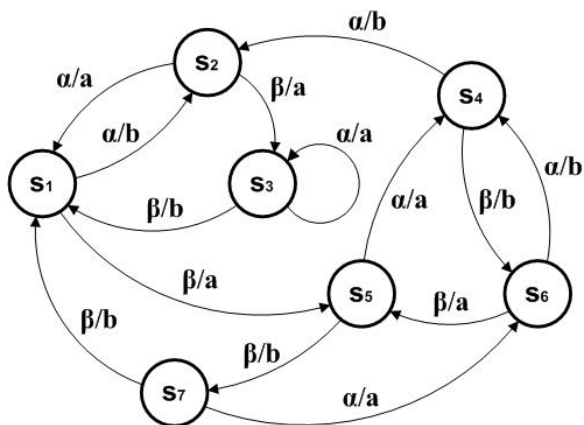
1. Осуществите структурный синтез конечного автомата Мили, заданного взвешенным орграфом переходов/выходов.

2. Выполните тестирование полученной функциональной логической схемы автомата из состояния  $s_1$  заданной тестовой последовательностью.

3. Укажите, изменения какого характера последуют на этапах структурного синтеза автомата Мура, эквивалентного заданному автомату Мили.

Варианты для заданий 1 – 3:

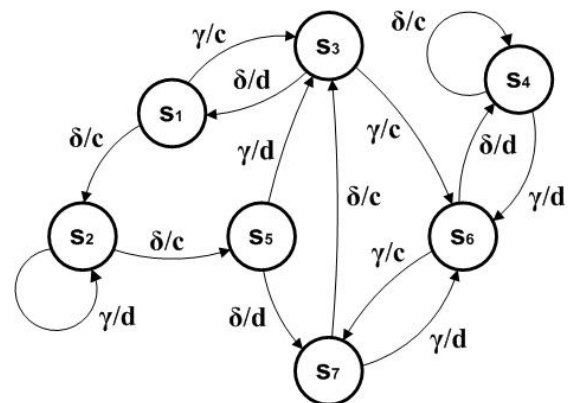
а)



Тестовая последовательность:

а)  $\alpha\beta\beta\beta\alpha$

б)



б)  $\gamma\gamma\delta\delta$ .

### Оглавление

## Литература

1. Брауэр В. Введение в теорию конечных автоматов; пер. с нем. – М.: Радио и связь, 1987. – 392 с.
2. Гилл А. Введение в теорию конечных автоматов; пер. с англ. – М.: Наука, 1966. – 272 с.
3. Карпов Ю.Г. Теория автоматов: Учебник для вузов. – СПб.: Питер, 2002. – 224 с.
4. Короткова М.А. Математическая теория автоматов: учебное пособие для вузов. – М.: Изд-во МИФИ, 2008. – 116 с.
5. Мелихов А.Н. Ориентированные графы и конечные автоматы. – М.: Наука, 1971. – 416 с.
6. Савельев А.Я. Основы информатики: Учебник для вузов. – М.: Изд-во МГТУ им. Баумана, 2001. – 328 с.
7. Савельев А.Я. Прикладная теория цифровых автоматов: Учебник для вузов. – М.: Высшая школа, 1987. – 272 с.
8. Хопкрофт Д., Мотвани Р., Ульман Дж. Введение в теорию автоматов, языков и вычислений; пер. с англ. – М.: «Вильямс», 2002. – 527 с.
9. Цилькер Б.Я. Организация ЭВМ и систем: Учебник для вузов. – СПб.: Питер, 2006. – 668 с.

### [Оглавление](#)