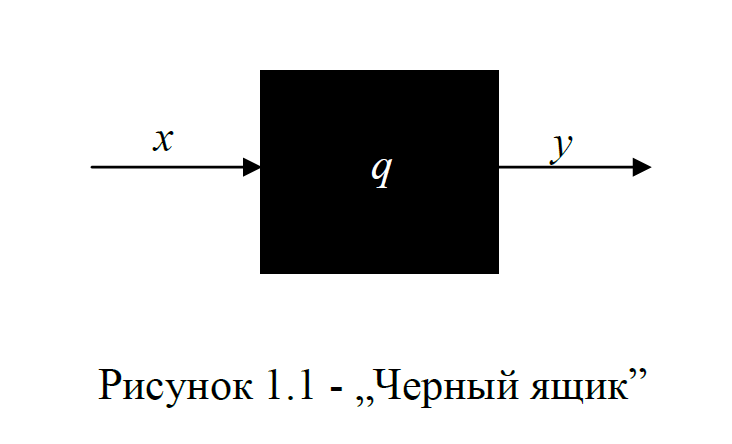
**Конечные автоматы**

1.1 Определение оконченного автомата

Рассмотрим абстрактную систему, традиционно называемую "черный ящик" (рис. 1.1), которая преобразует подаваемые на ее вход символы *x* определенного алфавита *X* в выходные символы *y* алфавита *Y*. Таким образом можно представить поведение большинства дискретных систем, применяемых в различных областях техники, особенно вычислительной.

*q*

Автомат – это такая система, функция которой может быть описана с помощью использования множества внутренних состояний Q {q} таким образом, что для каждого внутреннего состояния указано, в какое следующее состояние попадет система при получении входного символа x, и какой символ y формируется на выходе. Выделяется начальное состояние системы.

В случае конченых множеств X, Y, а, также, Q, автомат называют конечным.

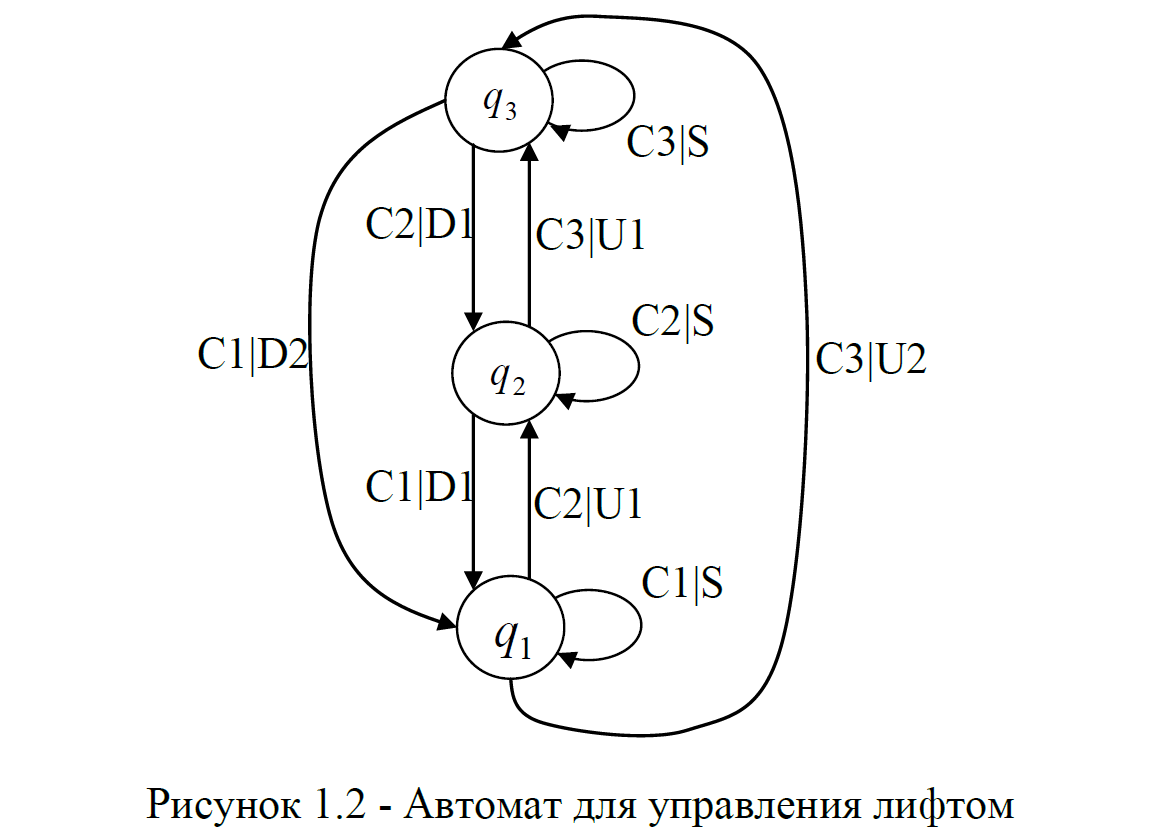
Построим конечный автомат, управляющий лифтом в трехэтажном доме.

Входной алфавит автомата состоит из нажатия кнопки вызова соответствующего этажа: X {C1, C2, C3};

выходной алфавит состоит из перемещений на один или два этажа вверх или книзу, а также остановки лифта: Y {U1, U2, D1, D2, S};

состояние соответствует этажу, на котором находится автомат: Q {q1, q2, q3};

для определенности выберем начальное состояние q1 . Функцию переходов автомата удобно представлять диаграммой состояний (рис. 1.2).



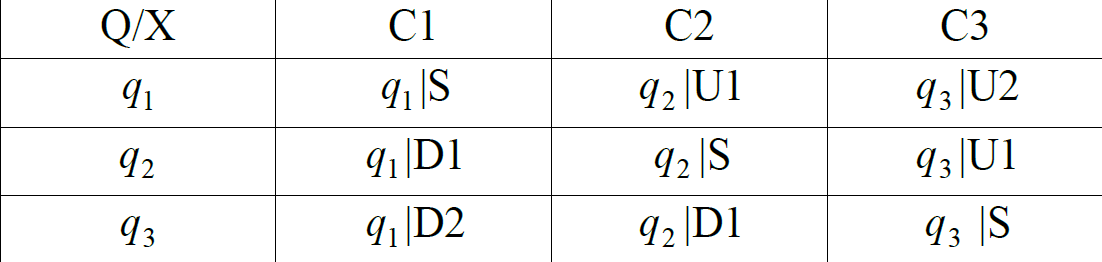
Автомат Мили

Итак, **конечный автомат – это пятерка A (X , Q, Y, q0, F)** ,

где X {x} –конечный входной алфавит, Y {y} – конечный выходной алфавит,

Q {q} –конечное множество внутренних состояний, q0 – начальное состояние, F: (X\* Q)🡪 (Y\* Q) – функция переходов.

Как было рассмотрено выше, функция переходов может быть наглядно представлена диаграммой состояний. Удобным бывает также матричное представление функции переходов. Приведем матрицу переходов для построенного ранее автомата:



Под синтезом, как правило, понимают построение конечного автомата по определенной заданной его спецификации. Спецификация, как в нашем примере с лифтом, может быть словесной. Иногда используют формальные спецификации, например, в виде регулярных выражений.

Синтез автомата заключается в получении его формального описания - диаграммы состояний или матрицы переходов.

Иногда удобным бывает более простое определение автомата, в котором функция переходов задает следующее состояние, а выход автомата зависит лишь от его текущего состояния. Такие автоматы называют автоматами Мура.

Итак, **автомат Мура – это шестерка B (X , S, Y, s0, P, R) ,**

где X ={x} –конечный входной алфавит,

Y= {y} – конечный выходной алфавит,

S= {s} – конечное множество внутренних состояний,

S0 – начальное состояние,

P: (X S)= S– функция переходов, R: (S)= Y – функция выходов.

Приведем пример простого автомата Мура, который распознает

**а(в)\*а**

Отметим, что переходы из состояний определены не для всех символов алфавита. Чтобы построить полностью определенный автомат, распознаватель часто дополняют невозвратным состоянием, в которое переходят при обработке любого недопустимого символа.

Оба способа определения конечного автомата являются эквивалентными в том смысле, что любой автомат Мили может быть преобразован в соответствующий ему автомат Мура и наоборот.

Покажем, что с помощью несложных построений произвольный автомат Мили может быть преобразован в автомат Мура.

Поскольку в автомате Мура выдача *выходного символа происходит при попадании в определенное состояние,* то для каждой пары (q, y) автомата Мили образуем состояние автомата Мура s со значением функции выходов R(s) =y .

Вершины Si(q1,y1) и Sj(q2,y2) соединяем дугой из S1в S2 ,тогда и только тогда, когда в автомате Мили была дуга из q1 в q2

Проиллюстрируем описанные действия на примере построения автомата Мура по автомату Мили управления лифтом (рис. 1.2);

диаграмма состояний полученного автомата Мура изображена на рис. 1.4

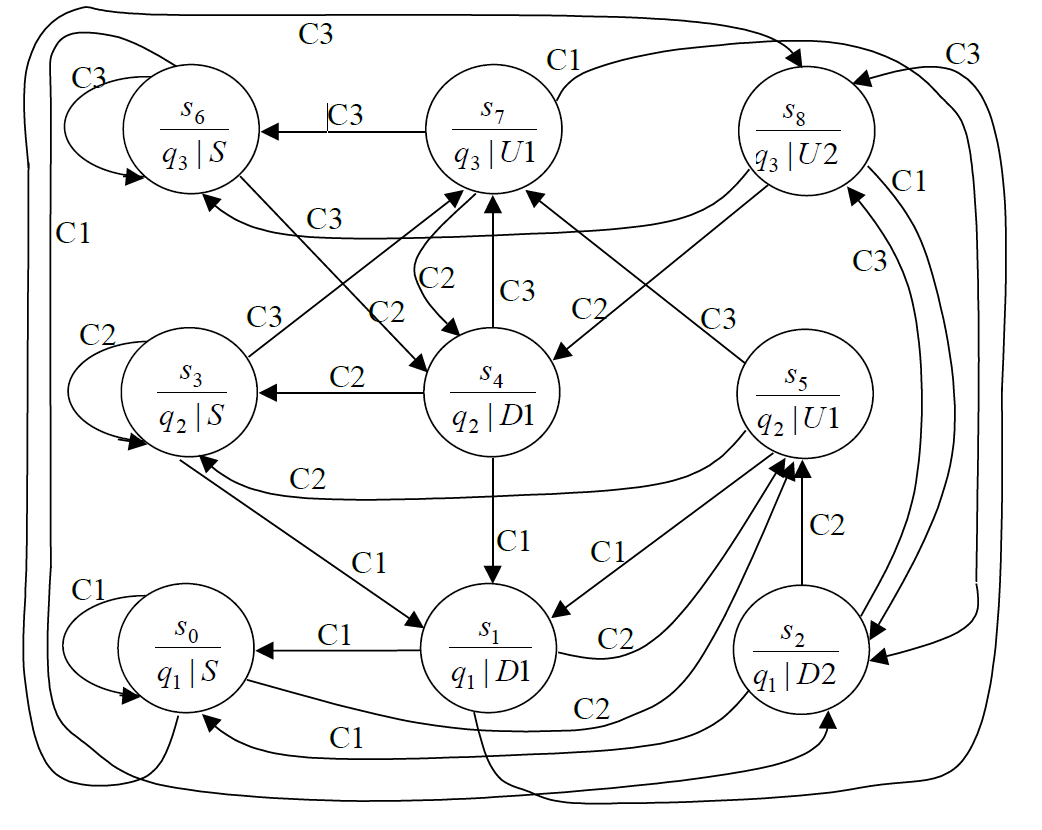


Рисунок 1.4 - Автомат Мура для управления лифтом

Полученный автомат имеет девять состояний; отметим, что новые состояния следует образовывать лишь для тех комбинаций состояний и выходных символов, дуги для которых присутствуют в исходном автомате. Хотя построенный автомат Мура получился на первый взгляд более сложным, чем исходный автомат, использование автоматов Мура, оказывается более удобным.

Формально преобразование можно представить следующим образом.

Пусть задан автомат Мили A= (X , Q, Y, q0, F) . Соответствующий ему автомат Мура B= (X, S, Y, s0, P, R) определим как S =Q\* Y , P(s, x) =s1 тогда и только тогда F(q, x) =(q1 , y) ,

R(s)= y тогда и только тогда F(q, x)= (q1 , y) ;

В качестве начального состояния s0 выберем произвольное из состояний, соответствующих q0 . Действительно, такой выбор является обоснованным, так как выдачу выходного символа в состоянии s0 мы не будем рассматривать, считая это выполненным «до начала» работы автомата.

Обратное преобразование требует намного меньше усилий и заключается в перенесении выходного символа избранного состояния на каждую из его входных дуг.

Итак, мы установили, что автоматы Мили и Мура имеют одинаковую изобразительную мощность, обосновав переход от одного автомата к другому и обратно. Возникает естественный вопрос о целесообразности применения двух типов автоматов.

С точки зрения удобства построения, более приемлемым бывает автомат Мили.

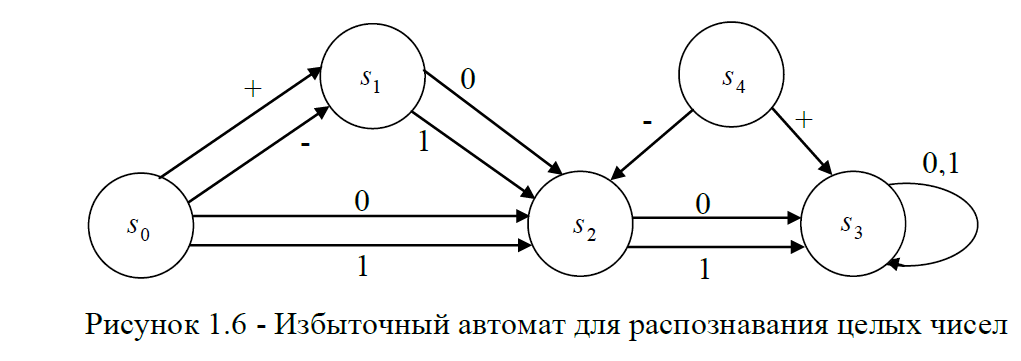
Автомат Мура широко используют в схемотехнике для синтеза реальных вычислительных и логических устройств на основе элементарных автоматов - триггеров. В таких устройствах состояние большей частью сохраняется и модифицируется, и выдается извне по специальной команде.

!!!!!!!!!!!!!

Минимизация конечных автоматов

Рассмотрение простого примера, представленного на рис. 1.6, позволяет сделать вывод, что один и тот же автомат может иметь множество представлений с разным количеством состояний и переходов. Так, например, состояние s4 является недостижимым из начального состояния и потому может быть изъято, а состояния s2 и s3 можно объединить;

Получить минимальное представление автомата весьма важно, если в дальнейшем оно будет реализовано аппаратно или программно. Кроме того, возникает задача для «похожих» автоматов определить формально, являются ли они эквивалентными.



В дальнейшем будем рассматривать автоматы, которые не имеют недостижимых состояний, поскольку такие состояния могут быть изъяты из диаграммы с помощью методов теории графов. Методы минимизации и определение эквивалентности будут представлены для автоматов Мура, что, не ограничивает их общности.

Два состояний автомата s1 и s2 будем называть n-эквивалентными, если для произвольной входной цепочки длиной n символов выходные цепочки символов совпадают. Отношение n-эквивалентности обозначим n~

Два состояний автомата s1 и s2 будем называть эквивалентными, если они являются n-эквивалентные для произвольного неотрицательного целого n.

Отношение эквивалентности обозначим ~.

**Теорема 1.1** В конечном автомате с *m* состояниями два произвольных

состояния являются эквивалентными тогда и только тогда, когда они (*m-* 2)

эквивалентные.

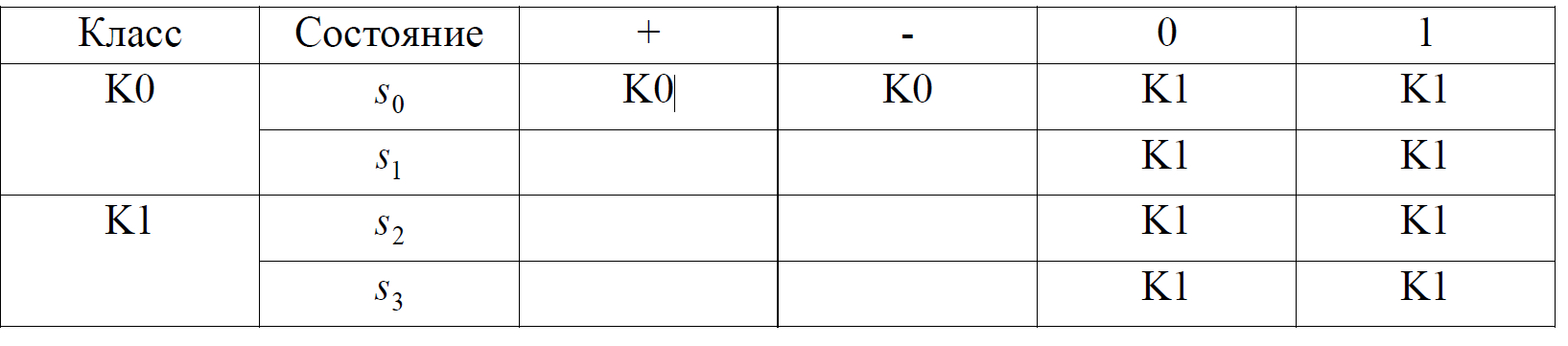
Доказать самрстоятельно!

Доказанная теорема позволяет предложить простой алгоритм минимизации

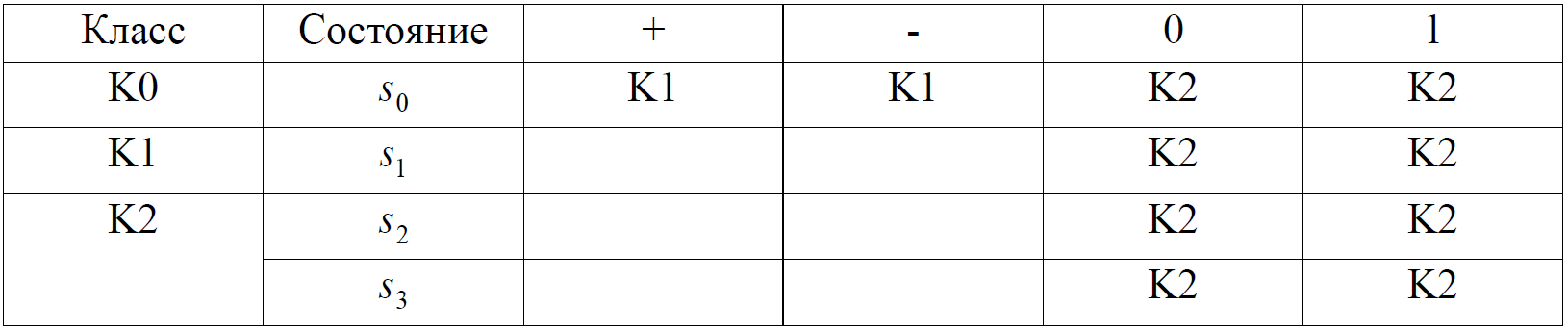
конечных автоматов.

Алгоритм состоит в последовательном построении разбиения множества состояний на одно-, два-, три- и так далее эквивалентные. Если текущее разбиение совпадает с предыдущим, то полученные классы эквивалентности и определяют минимальный автомат. В таблице разбиения указан номер класса следующего состояния. Первоначальное разбиение определяется функцией выхода:

нульэквивалентные состояния имеют одинаковые выходные символы\_\_



Класс K0 расщепляется на два новых эквивалентных класса, так как его строки не совпадают:



Дальнейшего расщепления классов эквивалентности не происходит, поэтому минимальный автомат содержит три состояния; диаграмма состояний соответствует ранее построенной в подразделе 1.2 и изображенной на рис. 1.3. Состояния минимального автомата соответствуют классам эквивалентности.

Алгоритм минимизации автоматов позволяет также определить их эквивалентность. Действительно, если необходимо установить, являются ли эквивалентными автоматы и , достаточно выполнить минимизацию автомата и определить, принадлежат ли их начальные состояния к тому же самому классу эквивалентности.

Итак, конечные автоматы являются простым и эффективным средством представления дискретных систем. Они широко используются в различных областях техники. Следует помнить, что любой, даже весьма мощный суперкомпьютер представляет собой конечный автомат. Пусть количество его состояний исчисляется миллиардами, но оно всегда является конечным

Вопросы

1. Из чего состоит конечный автомат?
2. Назовите способы представления конечного автомата.
3. Что изображают на диаграмме состояний конечного автомата?
4. Какова структура матрицы переходов конечного автомата?
5. Чем отличается автомат Мили от автомата Мура?
6. Сформулируйте основные действия по преобразованию автомата Мили в автомат Мура.
7. Сформулируйте основные действия по преобразованию автомата Мура в автомат Мили.
8. Укажите основные области применения конечных автоматов.
9. Укажите конечные автоматы в окружающих нас искусственных объектах.
10. В чем заключается целесообразность использования двух различных типов конечных автоматов?
11. Какие автоматы называют эквивалентными?
12. Охарактеризуйте основные шаги алгоритма минимизации конечного автомата?
13. Как определить, являются ли конечные автоматы эквивалентными?