

Лекция 14. Понятие логической сети. Схемы из функциональных элементов для систем булевых функций. Логические сети. Конечные автоматы. Понятие алгоритма. Требования к алгоритмическим процедурам. Рекурсивные функции.

Схемы из функциональных элементов для систем булевых функций

1. С определением и примерами **схем из функциональных элементов** мы уже встречались. Используя терминологию предыдущей главы, можно считать схему многополюсной сетью, входы схемы - входными полюсами, а выход - выходным полюсом. Элементы являются вершинами схемы, а их соединения - дугами, исходящими из входных полюсов или выходов других элементов и входящими во входы элементов или в выходной полюс.

Здесь мы будем рассматривать схемы из элементов, реализующих булевы функции. Для построения (синтеза) такой схемы функция должна быть представлена формулой, выражающей эту функцию через базисные функции, которые реализуются базисными элементами; они могут быть различными и должны обеспечивать возможность выразить их суперпозициями функции требуемого класса. Например, любую функцию можно выразить в виде ДНФ.

Тогда достаточно иметь набор элементов, содержащий:

конъюнктор - элемент с двумя входами, реализующий конъюнкцию;

дизъюнктор - элемент с двумя входами, реализующий дизъюнкцию;

инвертор - элемент с одним входом, реализующий отрицание.

В некоторых случаях удобно использовать и другие элементы, например, двухвходовый **элементарный сумматор**, реализующий сумму по модулю 2.

Расширим понятие схемы из функциональных элементов, допуская несколько различных выходов схемы, которая в этом случае реализует не одну функцию, а несколько (рис. 7.1а).

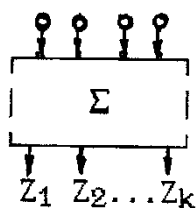


Рис 7.1 а

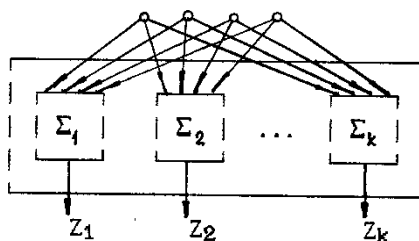


Рис 7.1б

Конечно, можно реализовать систему различных функций $f_1, f_2, \dots, f_k$, построив для них отдельные схемы $\Sigma_1, \Sigma_2, \dots, \Sigma_k$ совмещая (склеивая) соответствующие входные полюсы (рис. 7.1б). Однако некоторые внутренние части различных схем могут совпадать, и это можно использовать, чтобы не дублировать их и тем самым сократить общее число элементов схемы. Пример такого рода уже встречался ранее при реализации среднего значения M_x и выборочной дисперсии S_x для статистической выборки.

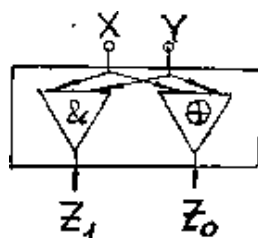
2. Рассматривая арифметические действия над двоичными числами (сложение, умножение и др.), мы отмечали, что знаки результата являются булевыми функциями от булевых переменных - знаков (цифр) слагаемых/сомножителей. Именно: если X и

Y - однозначные числа, то арифметическая сумма $X + Y$ представляет собой, вообще говоря, двузначное число $= Z_1 Z_0$, знаки которого выражаются так:

$$Z_0 = X \oplus Y, \quad Z_1 = X \& Y,$$

где Z_1 - знак переноса в старший разряд - равняется 1, если оба слагаемых равны 1 ($1 + 1 = 10$).

Соответствующая схема из функциональных элементов изображена на рис. 7.2.



При сложении двух двузначных, трехзначных и т.д. чисел младший разряд суммы определяется также: $Z_0 = X_0 \oplus Y_0$. Но возникающий знак переноса складывается с двумя знаками предыдущего (старшего) разряда; знак переноса из второго разряда в третий складывается, в свою очередь, с третьими справа знаками слагаемых и т.д.

Упражнение. Разберите подробно, как производятся поразрядные операции при сложении двоичных чисел 1011 и 1101: как определяются двоичные знаки суммы и знаки переноса.

Таким образом, при сложении многозначных чисел $\tilde{X} = X_{n-1} X_{n-2} \dots X_1 X_0$ и $\tilde{Y} = Y_{n-1} Y_{n-2} \dots Y_1 Y_0$ $(k+1)$ -й справа знак Z_k суммы $\tilde{Z} = Z_n Z_{n-1} \dots Z_k \dots Z_1 Z_0$ представляет собой сумму по модулю 2 трех булевых слагаемых X_k , Y_k и σ_k , где σ_k - знак переноса из $(k-1)$ -го разряда в k -й:

$$Z_k = X_k \oplus Y_k \oplus \sigma_k$$

Знак переноса из k -го разряда в $(k+1)$ -й равняется 1, если арифметическая сумма чисел $X_k + Y_k + \sigma_k$ превышает 1, т.е. если хотя бы два из трех слагаемых X_k , Y_k , σ_k равны 1. Это означает, что знак переноса равен значению функции большинства $m_3(X_k, Y_k, \sigma_k)$, которую можно выразить в виде ДНФ:

$$\sigma_{k+1} = m_3(X_k, Y_k, \sigma_k) = X_k \cdot Y_k \vee X_k \cdot \sigma_k \vee Y_k \cdot \sigma_k.$$

Вычисление булевых переменных Z_k и σ_{k+1} можно реализовать схемами, содержащими 2 элементарных сумматора для Z_k и 2 дизъюнктора и 3 конъюнктора для σ_{k+1} (рис. 7.3 а,б).

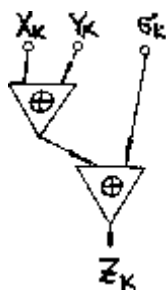


Рис 7.3 а

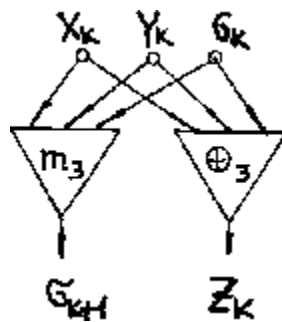


Рис 7.3б

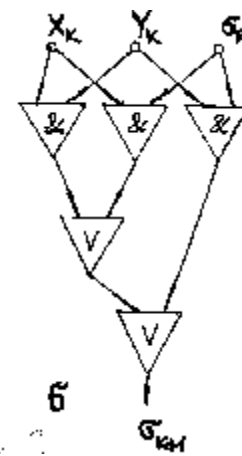


Рис 7.3с

Однако нам будет удобно использовать сокращенную форму (рис. 7.3с), обозначая через $\oplus_3$ каскад из двух элементарных сумматоров для суммирования трех слагаемых (рис. 7.3а) и через m_3 - трехвходовый элемент, реализующий функцию большинства (рис. 7.3б).

Склеивая входы двух этих элементов, мы получаем подсхему S, изображенную на рис. 7.4, которая представляет собой ячейку с тремя входами и двумя выходами, из нескольких экземпляров которой строится схема для реализации арифметического сумматора n-значных двоичных чисел $\tilde{X} = X_{n-1} X_{n-2} \dots X_1 X_0$ и $\tilde{Y} = Y_{n-1} Y_{n-2} \dots Y_1 Y_0$ (если числа $\tilde{X}$ и $\tilde{Y}$ имеют различную разрядность, то нужно дополнить меньшее нулями слева).

Схему рис. 7.2 для расчета Z_0 и можно (хотя и не обязательно) представить для единообразия как ту же ячейку S, если считать, что $\sigma_0 = 0$. Тогда общая схема для суммирования многозначных чисел может быть построена так, как на рис. 7.5. Схема имеет $2n$ входов и $(n+1)$ выходов. Нетрудно подсчитать, что для сложения n-значных чисел требуется $8n$ (если для расчета Z_0 использовать более экономную подсхему рис. 7.2, то $8n - 6$) элементов; например, для 20-значных чисел - 154 элемента.

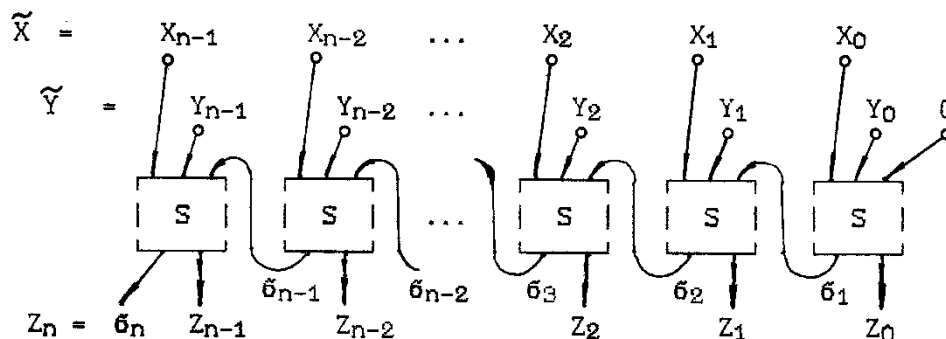


Рис. 7.5

Аналогично может быть построена и схема для умножения “столбиком” n-значных двоичных чисел $X = X_{n-1} \dots X_1 X_0$ и $Y = Y_{n-1} Y_{n-2} \dots Y_1 Y_0$. Рассмотрим в качестве примера случай $n = 3$, т.е. перемножение двух 4-значных чисел. В строках,

обозначенных T_0, T_1, T_2, T_3 , записываются (со сдвигом) произведения числа X на однозначные числа 0 или 1, определяемые знаками числа Y , т.е. либо число X , либо 0. Затем записанные числа складываются; при этом число однозначных слагаемых в каждом разряде - от 1 до n . В общем случае k -й разряд произведения Z_k ($k = 0, 1, \dots, n-1$) зависит от знаков младших разрядов сомножителей, а при $k > n$ - от всех $2n$ знаков, но, в отличие от сложения, в определении Z_k участвуют несколько знаков переноса из младших разрядов. Результат перемножения двух n -значных чисел представляет $2n$ -значное число.

Пример для $n = 4$:

$$\begin{array}{r} \tilde{X} = X_3 \quad X_2 \quad X_1 \quad X_0 \\ \tilde{Y} = Y_3 \quad Y_2 \quad Y_1 \quad Y_0 \end{array}$$

$T_0 = \tilde{X} \cdot Y_0$, сдвиг 0	$X_3 \cdot Y_0$	$X_2 \cdot Y_0$	$X_1 \cdot Y_0$	$X_0 \cdot Y_0$					
$T_1 = \tilde{X} \cdot Y_1$, сдвиг 0	$X_3 \cdot Y_1$	$X_2 \cdot Y_1$	$X_1 \cdot Y_1$	$X_0 \cdot Y_1$					
$T_2 = \tilde{X} \cdot Y_2$, сдвиг 0	$X_3 \cdot Y_2$	$X_2 \cdot Y_2$	$X_1 \cdot Y_2$	$X_0 \cdot Y_2$					
$T_3 = \tilde{X} \cdot Y_3$, сдвиг 0	$X_3 \cdot Y_3$	$X_2 \cdot Y_3$	$X_1 \cdot Y_3$	$X_0 \cdot Y_3$					
	Z_7	Z_6	Z_5	Z_4	Z_3	Z_2	Z_1	Z_0	

$$\begin{array}{r} 1 \ 1 \ 1 \ 0 = 14 \\ 1 \ 0 \ 0 \ 1 = 9 \\ \hline 1 \ 1 \ 1 \ 0 \\ 0 \ 0 \ 0 \ 0 \\ 0 \ 0 \ 0 \ 0 \\ 1 \ 1 \ 1 \ 0 \\ \hline 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 = 12 \\ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 = 6 \end{array}$$

3. Свойства схем из функциональных элементов.

1) Отсутствие ориентированных циклов (контуров). Этим обеспечивается невозможность попадания выходного значения некоторого элемента на вход того же элемента, даже если оно преобразовано действием других элементов. Иначе могли бы возникнуть противоречия, когда, например, нулевое значение на входе при прохождении по контуру могло превратиться в единицу, и не понятно, какое именно значение должно поступить на вход.

2) Вычисление линейно упорядоченной системы функций можно рассматривать как вычисление - для каждого набора входных значений - многозначного двоичного числа, знаками которого является кортеж выходных значений (слово в алфавите $B = \{0, 1\}$).

3) Следует считать, что вычисление совершается без учета времени, т.е. все вычисляется как бы одновременно и не возникает коллизий, что из нескольких входных аргументов какого-то элемента одни уже вычислены, а другие еще нет.

Пример. Пусть в подъезде 5-этажного дома работают 2 лифта. Требуется построить схему, которая, если оба лифта находятся выше 1-го этажа, определяла бы приоритет вызова одного из лифтов, так чтобы нажатием одной кнопки вызова можно было вызвать ближайший из них.

Занумеруем лифты числами 0 и 1, а этажи с 2 по 5 - числами 0, 1, 2, 3, которые будем представлять их двоичными записями 00, 01, 10, 11. Тогда совместное расположение обоих лифтов задается упорядоченной парой чисел $\alpha\beta \in [0, 3] \subseteq \mathbb{N}$, или четверкой булевых параметров $\alpha_1\alpha_2\beta_1\beta_2$. Например, набор 0110 означает, что первый лифт находится на 3 этаже ($\alpha = 01$), а второй - на 4-м ($\beta = 10$).

Для построения схемы нужна формула, выражающая зависимость номера Z приоритетного лифта от переменных $\alpha_1\alpha_2\beta_1\beta_2$. Составим таблицу 7.1.

Здесь Z принимает значение 0, если 1-й лифт находится ниже 2-го, и 1, если 2-й лифт - ниже 1-го. Значения представлены в графе Z . В случаях, когда оба лифта находятся на одном этаже, можно определить приоритет и, тем самым, функцию Z как угодно. Доопределим Z так, чтобы формула была возможно проще. Это можно сделать, как показано в столбце “доопределение Z ”. В последнем столбце таблицы представлено разложение функции Z по двум первым переменным: $Z = \bar{\alpha}_1\bar{\alpha}_2 \cdot 0 \vee \bar{\alpha}_1\alpha_2 \cdot \bar{\beta}_1 \vee \alpha_1\bar{\alpha}_2 \cdot \bar{\beta}_1 \vee \alpha_1\alpha_2 \cdot 1$. Первое слагаемое в этой дизъюнкции четырех конъюнкций тождественно равно 0; из двух следующих можно вынести за скобки общий множитель $\bar{\beta}_1$: дизъюнкция есть СДНФ функции $(\alpha_1 \oplus \alpha_2)$; четвертое слагаемое равно $\alpha_1\alpha_2$. Окончательно получаем:

$$Z = \bar{\beta}_1 \cdot (\alpha_1 \oplus \alpha_2) \vee \alpha_1\alpha_2.$$

Интересно, что при таком доопределении функция Z не зависит от переменной β_2 . Это значит, что на нахождение 2-го лифта на 2-м или на 3-м этаже система реагирует одинаково, так же как не различается его нахождение на 4-м этаже от 5-го.

Соответствующая схема из 5 элементов изображена на рис. 7.6.

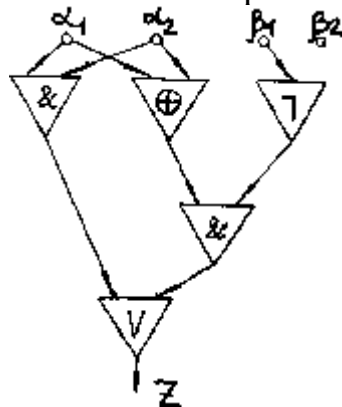


Рис 7.6

Таблица 7.1

$\alpha\beta$	$\alpha_1 \alpha_2 \beta_1 \beta_2$	Z	Доопределение Z	Z после доопределения	$f(\alpha_1, \alpha_2, \beta_1, \beta_2)$
00	0000		0	0	
01	0001	0		0	
02	0010	0		0	
03	0011	0		0	
10	0100	1		1	$\overline{\beta_1}$
11	0101		1	1	
12	0110	0		0	
13	1111	0		0	
20	1000	1		1	$\overline{\beta_1}$
21	1001	1		1	
22	1010		0	0	
23	1011	0		0	
30	1100	1		1	
31	1101	1		1	
32	1110	1		1	
33	1111		1	1	

Логические сети

1. Расширим понятие схемы из функциональных элементов. Определим ее функционирование в дискретном времени $t = 0, 1, 2, \dots$. Считается, что в каждый момент времени на входы схемы подаются значения аргументов: для n -входовой схемы

$$X(t) = (x_1(t), x_2(t), \dots, x_n(t)).$$

Если схема состоит, как мы до сих пор рассматривали, только из функциональных элементов, то в тот же момент времени t на выходах схемы реализуются функции алгебры логики на входном наборе, определяемые структурой схемы и составом элементов, т.е. возникает выходной набор

$$Z(t) = (z_1(t), z_2(t), \dots, z_m(t)),$$

причем значения $z_i(t)$ зависят только от совокупности значений аргументов $x_1(t), x_2(t), \dots, x_n(t)$ в тот же момент времени.

Введем теперь новый тип элемента с одним входом, который называется **задержкой** и обозначается символом “з”; его функционирование определяется так: если $x(t)$ - входной аргумент элемента задержки, то $z(t+1) = x(t)$, т.е. значение на выходе задержки в каждый момент времени равно значению на входе элемента в предыдущий момент времени. Заметим, что значение $z(0)$ не может быть определено указанным равенством: полагают по определению, что $z(0) = 0$.

Включение элементов задержки в схему позволяет связывать значения, подаваемые на входы и вычисляемые внутри схемы (как выходные, так и промежуточные) в разные моменты времени.

Схема из функциональных элементов и задержек, действующая в тактовом (дискретном) режиме, называется **логической сетью**. При этом допустимы

ориентированные циклы (контуры), если в них присутствует хотя бы один элемент задержки.

Существенное свойство логической сети: если задать входную последовательность $X(0), X(1), \dots$, то однозначно определяется выходная последовательность $Z(0), Z(1), \dots$. Действительно, на вход каждого функционального элемента или задержки подается либо входное значение, либо выход другого элемента, в частности, задержки. Поэтому для всех t определены выходные значения каждого элемента, а для $t = 0$ выходы всех задержек равны 0 по определению.

Пример 1 (рис. 7.7). Выходное значение $Z(t)$ определяется равенством $Z(t) = (X(t) \vee Y(t)) \& z(t) = [поскольку\ z(t) = Y(t-1)] = (X(t)Y(t)) \& Y(t-1)$. $z(0) = 0$. Поэтому входная последовательность $(X(t), Y(t))$ перерабатывается схемой рис. 7.7 в выходную последовательность $Z(t)$ (см. таблицу). В этой логической сети нет контуров.

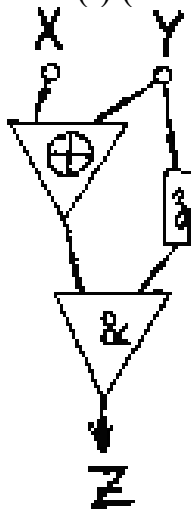


Рис 7.7

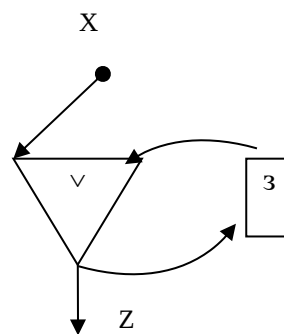


Рис 7.8

t	0	1	2	3	4	5	...
X	0	1	1	1	0	0	...
Y	1	0	1	0	1	1	...
Z	0	1	0	1	0	1	...

Пример 2 (рис. 7.8). $Z(t) = X(t) \vee z(t) = X(t) \vee Z(t-1)$.

Аналогично, $Z(t-1) = X(t-1) \vee Z(t-2)$. Продолжая эту рекурсию, получаем:

$$Z(t) = X(t) \vee X(t-1) \vee X(t-2) \vee \dots \vee X(0) = \bigvee_{\tau=0}^t X(\tau).$$

$$\text{Исходя из определения дизъюнкции } Z(t) = \begin{cases} 0, & \text{если } \forall_{\tau \leq t} : X(\tau) = 0, \\ 1, & \text{в противном случае,} \\ & \text{т.е. если } \exists_{\tau \leq t} : X(\tau) = 1. \end{cases}$$

Таким образом, $Z(t)$ тождественно равно 0 до появления на входе первой единицы, после чего тождественно равно 1.

Пример 3 (рис. 7.9). Обозначим через A и B выходы конъюнкторов. Выход задержки равен выходу сети в предыдущий момент: $z(t) = Z(t-1)$. Выразим теперь $Z(t)$ через введенные обозначения:

$$A(t) = X(t) \& \bar{Z}(t-1),$$

$$B(t) = X(t) \& Z(t-1),$$

$Z(t) = A(t) \vee B(t) = [X(t) \& \bar{Z}(t-1)] \vee [\bar{X}(t) \& Z(t-1)] = [\text{известное тождество}] = X(t)$
 $\oplus Z(t-1) = X(t) \oplus X(t-1) \oplus Z(t-2) = \dots = X(t) \oplus X(t-1) \oplus \dots \oplus X(0) = \bigoplus_{\tau=0}^t X(\tau)$. Итак, выходное значение меняется после каждой следующей входной единицы, или, по-другому, $Z(t)$ определяется четностью числа единиц в начальном отрезке входной последовательности.

Используя последнее выражение, можно построить эквивалентную схему (рис. 7.10).

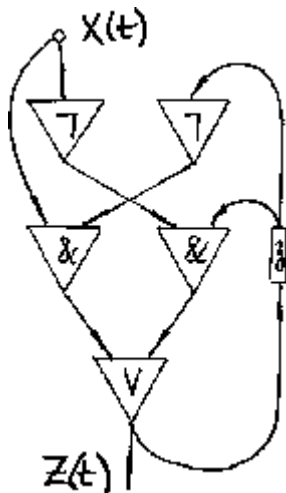


Рис 7.9



Рис 7.10

Данный пример показывает, что выходное значение может зависеть от многих (в частности, от всех) предыдущих значений аргумента: так, если изменить значение $X(t)$ при одном значении t , то изменятся все последующие значения $Z(t)$.

Как показывают рассмотренные примеры, логическая сеть перерабатывает (т.е. отображает) входную последовательность $X(t)$ (или последовательность векторов $(X(t), Y(t))$ и т.п.) в выходную $Z(t)$. Это отображение называется **оператором** $X(t) \rightarrow Z(t)$.

Конечные автоматы

1. Конечным автоматом называется система $(A, B, Q,)$, где $A = \{a_1, a_2, \dots, a_p\}$ - входной алфавит, $B = \{b_1, b_2, \dots, b_r\}$ - выходной алфавит, $Q = \{q_1, q_2, \dots, q_w\}$ - множество внутренних состояний (q_1 называется начальным состоянием); функция выходов $\varphi: Q \times A \rightarrow B$ и функция переходов $\psi: Q \times A \rightarrow Q$ определены на множестве $Q \times A$.

Функционирование автомата происходит в тактовом дискретном режиме для $t = 0, 1, 2, \dots$. На вход автомата поступает входная последовательность $a(0), a(1), a(2), \dots, a(i) \in A$. Пара функций φ и ψ образует **систему канонических уравнений**:

$$\begin{aligned}
 b(t) &= \varphi(q(t), a(t)), \\
 q(t+1) &= \psi(q(t), a(t)),
 \end{aligned}$$

при этом считается, что $q(0) = q_1$.

Первое уравнение задает выходные значения в момент времени t как функцию входного значения в тот же момент времени и текущего внутреннего состояния автомата. Второе уравнение определяет переход в очередное внутреннее состояние, в котором окажется автомат в следующий момент времени. Совокупность $(q(t), a(t))$,

полностью определяющая текущее значение выхода и следующее внутреннее состояние автомата, называют полным состоянием автомата.

Основное свойство автомата состоит в следующем: если задать входную последовательность $\{a(t)\}$, то канонические уравнения однозначно определяют выходную последовательность $\{b(t)\}$ и последовательность состояний $\{q(t)\}$. Действительно, $(q(0), a(0))$ определяют $(q(1), b(0))$, $(q(1), a(1))$ определяют $(q(2), b(1))$ и т.д. Иными словами, автомат осуществляет однозначное отображение входной последовательности в выходную, т.е. реализует оператор $A(t) \rightarrow B(t)$.

Поскольку функции и определены на конечных множествах, их можно задать таблицами, которые обычно сводятся в одну таблицу для их декартова произведения $\varphi \times \psi : Q \times A \rightarrow Q \times V$, называемую **таблицей (или матрицей) переходов автомата**; она имеет w строк и p столбцов.

Пример 1. Таблица 7.2 задает функции переходов и выходов для автомата с алфавитами $A = \{a, b, c\}$, $V = \{1, 2\}$, $Q = \{q_1, q_2, q_3, q_4\}$; символы q_i и b_j разделены знаком /.

Таблица 7.2

	a	b	c
q_1	$q_3/1$	$q_3/2$	$q_2/1$
q_2	$q_4/1$	$q_1/1$	$q_1/1$
q_3	$q_2/1$	$q_3/1$	$q_3/2$
q_4	$q_4/1$	$q_2/1$	$q_1/2$

Автомат можно задавать также **графом (или диаграммой) переходов** – ориентированным графом (Q, E) , где Q – множество вершин, а дуги соответствуют клеткам таблицы переходов: если на пересечении строки q_i и столбца a_j в таблице содержится пара (q_k / b) , то этой клетке соответствует дуга (q_i, q_k) , которой приписана упорядоченная пара $(a_j; b)$. Граф переходов для таблицы 7.2 изображен на рис. 7.11.

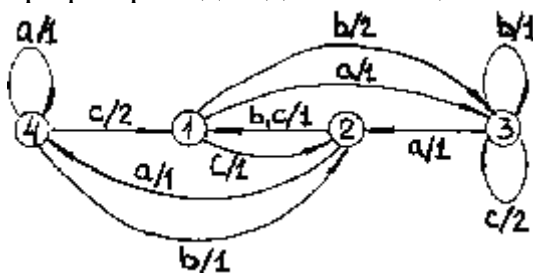


Рис 7.11

Из каждой вершины графа переходов исходят p дуг, соответствующих столбцам таблицы переходов. Однако для большей наглядности несколько кратных дуг, которым приписаны пары $(a_i; b), \dots, (a_i; b)$ с одинаковым выходным значением, могут быть склеены в одну дугу, которой приписывается кортеж $(a_i, \dots, a_i; b)$. Так, автомат на рис. 7.11 в ситуациях q_2b и q_2c переходит в одно и то же состояние q_1 при одинаковом выходном значении 1.