

**С.Л. Блюмин, А.М. Корнеев**

ДИСКРЕТНОЕ  
МОДЕЛИРОВАНИЕ  
СИСТЕМ  
АВТОМАТИЗАЦИИ И УПРАВЛЕНИЯ

**Блюмин, С. Л.** Дискретное моделирование систем автоматизации и управления [Текст]: монография / С.Л. Блюмин, А.М. Корнеев ; Липецкий эколого-гуманит. ин-т. – Липецк: ЛЭГИ, 2005. – 124 с. ; библиогр. с. 121-123 ; 21 см.

Представлены подходы к построению дискретных моделей, базирующихся на клеточной интерпретации. Рассмотрены дискретные пространственные, дискретные пространственно-временные, конечно-аргументные модели. Дискретно-аргументный подход можно использовать для описания сложных технологических процессов, характеризующихся многостадийностью обработки. Технологический процесс представлен в виде клеток, каждая из которых соответствует отдельному технологическому агрегату или переделу. Описав в виде входов, состояний и выходов характеристики сырья, технологических параметров и свойств готовой продукции, можно моделировать технологии обработки любых видов продукции. Использование конечных и вероятностных автоматов, итеративных цепей и сетей позволяет описать любой сложный технологический процесс и осуществлять управление технологией.

Монография утверждена и рекомендована к печати научно-техническим советом ЛЭГИ для студентов по отдельным дисциплинам специальностей 220501 «Управление качеством», 230401 «Прикладная математика», 230102 «Автоматизированные системы обработки информации и управления», а также для аспирантов и преподавателей вузов.

Библиография: 46 назв.

**Рецензенты:**

кафедра математического анализа, алгебры и геометрии Липецкого государственного педагогического университета (заведующий кафедрой, доктор физико-математических наук, профессор **Калитвин А.С.**);

доктор технических наук, профессор **О.Я. Кравец**  
(Воронежский государственный технический университет)

**ISBN 5-900037-47-9**

## Оглавление

Введение .....	5
<b>1. Автоматы .....</b>	<b>6</b>
1.1. Конечные автоматы (КА). Дискретно-детерминированные схемы .....	6
1.2. Описание технологии с помощью конечных автоматов.....	16
1.3. Вероятностные конечные автоматы (ВКА) (P-схемы). .....	31
(Дискретно-стохастические модели) .....	31
1.4. Представление технологического процесса в виде вероятностного автомата.....	36
<b>2. Дискретные пространственные модели .....</b>	<b>42</b>
2.1. Одномерные однонаправленные линейные итеративные цепи.....	43
2.2. Одномерные двунаправленные линейные итеративные цепи.	44
2.3. Двумерные однонаправленные линейные итеративные цепи.	46
2.4. Двумерные двунаправленные линейные итеративные цепи ...	50
2.5. Двумерные полуторанаправленные линейные итеративные цепи.....	53
2.6. Другие двумерные цепи.....	54
2.7. Трехмерные однонаправленные линейные итеративные цепи	56
2.8. Многомерные однонаправленные линейные итеративные цепи.....	56
2.9. Использование итеративных цепей для описания технологического процесса.....	58
<b>3. Дискретные пространственно-временные модели .....</b>	<b>64</b>
3.1. Одномерные однонаправленные линейные итеративные сети	64
3.2. Одномерные двунаправленные итеративные сети .....	69
3.3. Одномерные линейные клеточные автоматы.....	70
3.4. Клеточные конечные автоматы (ККА).....	70
3.5. Линейные клеточные автоматы. Переход к ассоциированному уравнению .....	74
3.6. Аналитическое представление ККА.....	81
3.7. Использование дискретных пространственно-временных моделей при расчете себестоимости металлопродукции.....	91
<b>4. Дискретные конечно-аргументные модели .....</b>	<b>98</b>
4.1. Одномерные циклические линейные итеративные цепи (N-цепи) .....	100
4.2. Одномерные полициклические линейные итеративные цепи ((m)-цепи).....	102
4.3. Двумерные полициклические линейные итеративные цепи (двумерные (m)-цепи).....	103
4.4. Одномерные полициклические линейные итеративные сети (решетчатые модели).....	105
<b>5. Сети Петри.....</b>	<b>107</b>
5.1. Описание сетей Петри .....	107
5.2. Использование сетей Петри при расчете себестоимости продуктов металлургического производства.....	113
<b>Заключение.....</b>	<b>119</b>
<b>Библиографический список.....</b>	<b>120</b>

### **Внимание!**

*В конец данного файла вынесены  
некоторые фрагменты основного текста:*

Таблица 1.1 (с. 10),   Таблица 1.2 (с. 12),  
Таблица 1.3 (с. 13),   Таблица 1.23 (с. 27),  
Таблица 1.35 (с. 40),   Рис. 2.16 (с. 61),  
Таблица 3.1 (с. 68),   Таблица 3.2 (с. 83).

## Введение

Клеточная интерпретация в математической теории дискретно-аргументных систем играет роль, подобную теории геометрической интерпретации в математическом анализе или гидродинамической интерпретации в математической теории скалярных и векторных полей. Основанное на этой интерпретации представление дискретно-аргументных систем столь удобно, наглядно и естественно, что в обзоре [1] значительное количество таких систем объединено под названием «линейные клеточные машины».

Сущность клеточной интерпретации состоит в том, что подавляющее большинство известных к настоящему времени дискретно-аргументных моделей допускает представление в виде тех или иных клеточных конфигураций в широком смысле слова. При этом используются всего два типа простейших клеток: клетка без памяти, или комбинационная цепь, и клетка с памятью, или дискретно-временная динамическая система. Они являются элементарными «дискретно-аргументными блоками» в структурной теории дискретно-аргументных систем, которая может быть разработана подобно являющейся уже классической структурной теории сосредоточенных систем или возникшей в последнее время структурной теории сложных распределенных систем [2]. В [2] предложено под «элементарным распределенным блоком» понимать всякую распределенную систему, для которой известен в явном виде оператор (аналитический или алгоритмический), однозначно относящий каждому входному сигналу некоторый выходной сигнал. В соответствии с таким определением «элементарные блоки» могут описывать весьма сложные краевые задачи для уравнений математической физики.

Дискретно-аргументный подход можно использовать для описания сложных технологических процессов, характеризующихся многостадийностью обработки.

Технологический процесс можно представить в виде клеток, каждая из которых соответствует отдельному технологическому агрегату или переделу. Описав в виде входов, состояний и выходов характеристики сырья, технологических параметров и свойств готовой продукции, можно моделировать технологии обработки любых видов продукции. Использование конечных и вероятностных автоматов, итеративных цепей и сетей позволяет описать любой сложный технологический процесс и осуществлять управление технологией.

## 1. Автоматы

### 1.1. Конечные автоматы (КА).

#### Дискретно-детерминированные схемы

На основе теории автоматов система представляется в виде автомата, перерабатывающего дискретную информацию и меняющего свои внутренние состояния лишь в допустимые моменты времени  $t \in Z_0 = \{0, 1, 2, \dots\}$  [3–5].

Автомат можно представить как некоторое устройство (черный ящик), на которое передаются входные сигналы и снимаются выходные.

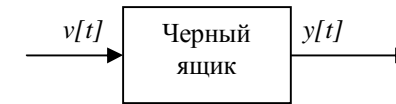


Рис. 1.1. Устройство «чёрный ящик»

Но так как у автомата по определению существует внутренняя характеристика (состояние), то это делает «черный» ящик «серым».

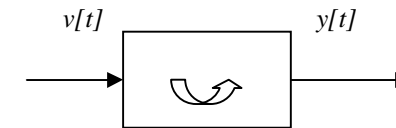


Рис. 1.2. Устройство «серый ящик»

Конечный автомат – автомат, у которого множества значений внутренних состояний, входных и выходных сигналов конечны.

Конечный автомат можно представить как математическую схему (F-схему), характеризующуюся шестью элементами [4,5]:

1). Конечное множество входных сигналов  $V$  (входной алфавит).

Входные сигналы  $v[t]$  в любой момент времени принимают значения из этого входного алфавита  $v[t] \in V, t \in Z_0$ .

2). Конечный внутренний алфавит  $X$ .

Состояние автомата  $x[t]$  принимает значения из этого внутреннего алфавита:  $x[t] \in X, t \in Z_0$

3). Конечный выходной алфавит  $Y$ .

$y[t] \in Y, t \in Z_0$

4). Начальное состояние (состояние в начальный момент времени).

$$x_0 = x[0]$$

5). Функция переходов.

Определяет состояние конечного автомата в некоторый момент времени  $t$ , если в предшествующий момент он был в каком-то состоянии  $x[t-1]$  и на вход поступил сигнал  $v[t]$ :

$$x[t] = j(x[t-1], v[t]). \quad (1.1)$$

При заданном начальном состоянии конечный автомат может функционировать, даже если нет входного воздействия (автономный автомат).

б). Функции выходов определяется типом конечного автомата (рассмотрим автоматы Мили и Мура).

а). Конечный автомат Мили.

Для автомата Мили выход определяется состоянием автомата в предыдущий момент времени  $x[t-1]$  и сигналом, поступающим на вход в данный момент  $v[t]$ :

$$y[t] = y(x[t-1], v[t]). \quad (1.2)$$

б). Конечный автомат Мура.

Для автомата Мура выход определяется состоянием автомата в тот же момент времени и не зависит от входного сигнала:

$$y[t] = y(x[t]). \quad (1.3)$$

Конечный автомат определяется как кортеж (шестерка)

$$F = \langle V, X, Y, x_0, j, y \rangle. \quad (1.4)$$

Конечный автомат функционирует в дискретном автоматном времени, моментами которого являются такты, т.е. примыкающие друг к другу равные интервалы времени, каждому из которых соответствуют постоянные значения входного и выходного сигналов и внутренние состояния.

Конечный автомат в каждый момент  $t = 0, 1, 2, \dots$  дискретного времени находится в определенном состоянии  $x[t]$ , причем, в начальный момент времени  $t = 0$  он всегда находится в начальном состоянии  $x_0$ . В момент  $t$ , будучи в состоянии  $x[t]$ , автомат способен воспринять на входном канале сигнал  $v[t]$  и выдать на выходном канале сигнал  $y[t]$ , переходя в состояние  $x[t+1]$ .

Работа конечного автомата происходит по схеме: на каждом  $t$ -м такте на вход автомата, находящегося в состоянии  $x[t]$ , подается некоторый сигнал  $v[t]$ , на который он реагирует переходом в  $[t+1]$ -м такте в новое состояние  $x[t+1]$  и выдачей некоторого выходного сигнала.

В общем случае конечный автомат имеет много входов, состояний и выходов.

$v[t] = (v_1[t], \dots, v_m[t])$  – вектор составленный из скалярных входов ( $m$  – число входов).

$x[t] = (x_1[t], \dots, x_n[t])$ ,  $n$  – число состояний.

$y[t] = (y_1[t], \dots, y_L[t])$ ,  $L$  – число выходов.

В этом случае алфавиты представляют собой прямые произведения более простых алфавитов, т.е. каждая координата принимает значение в своем конечном алфавите.

Аналогично для состояний и выходов.

$$\begin{cases} v = v_1 \times \dots \times v_m \\ x = x_1 \times \dots \times x_n \\ y = y_1 \times \dots \times y_L \end{cases} \quad (1.5)$$

В этом случае функции переходов и выходов допускают развернутую запись:

функция переходов для автоматов Мили и Мура.

$$x_i[t] = j_i(x_1[t-1], \dots, x_n[t-1]; v_1[t], \dots, v_m[t]), \quad (1.6)$$

$$i = 1, \dots, n;$$

функция выходов для автомата Мили:

$$y_i[t] = y_j(x_1[t-1], \dots, x_n[t-1]; v_1[t], \dots, v_m[t]), \quad (1.7)$$

$$j = 1, \dots, l;$$

функция выходов для автомата Мура:

$$y_i[t] = y_j^0(x_1[t], \dots, x_n[t]), \quad (1.8)$$

$$j = 1, \dots, l.$$

Автомат Мили – более общий автомат, чем автомат Мура. У каждого конечного автомата Мура есть конечный автомат Мили его интерпретирующий.

Если ограничиться рассмотрением двоичных или булевых автоматов, то алфавиты входов, состояний и выходов составят произведения:

$$V = \{0, 1\} \times \dots \times \{0, 1\} = \{0, 1\}^m = \{s_a, a = 1, \dots, 2^m\}, \quad (1.9)$$

$$X = \{0, 1\}^n = \{x_b, b = 1, \dots, 2^n\} \quad (1.10)$$

$$Y = \{0, 1\}^l = \{t_j, j = 1, \dots, 2^l\}, \quad (1.11)$$

где  $s_a, x_b, t_j$  – элементы прямого произведения.

Тогда функции переходов и выходов запишутся в виде:

запись КА Мура (всегда штрих будет означать предшествующий момент времени)

$$\begin{cases} x'_b = j(x_b, s_a) \\ t'_g = y^0(x_b) \end{cases}; \quad (1.12)$$

запись КА Мили

$$\begin{cases} x'_b = j(x_b, s_a) \\ t'_g = y(x_b, s_a) \end{cases}. \quad (1.13)$$

Алфавиты входов, состояний и выходов могут иметь различную значимость.

$a_{ij_i}$  – составляющие алфавитов входов;

$j_i = 1, \dots, J_i$ , где  $J_i$  – значность алфавита  $i$ -го выхода;

$i = 1, \dots, m$  – номера входов;

$b_{kj_k}$  – составляющие алфавитов состояний;

$j_k = 1, \dots, J_k$ , где  $J_k$  – значность алфавита  $k$ -го состояния;

$k = 1, \dots, n$  – номера состояний;

$C_{pj_p}$  – составляющие алфавитов выходов;

$j_p = 1, \dots, J_p$ , где  $J_p$  – значность алфавита  $p$ -го выхода;

$p = 1, \dots, L$ , где  $p$  – номер выхода.

Так как в общем случае конечный автомат имеет несколько входов, состояний и выходов, то алфавиты представляют собой прямые произведения более простых алфавитов.

Таким образом:

◆ конечный входной алфавит

$$V = \{a_{11}, \dots, a_{1J_1}\} \times \dots \times \{a_{m1}, \dots, a_{mJ_m}\} = \{S_a, a = 1, \dots, A; A = \prod_{i=1}^m J_i\}, \quad (1.14)$$

где  $S_a$  – вариант сочетания алфавитов входов  $v[t]$ ;

◆ конечный алфавит состояний

$$X = \{b_{11}, \dots, b_{1J_1}\} \times \dots \times \{b_{n1}, \dots, b_{nJ_n}\} = \{X_b, b = 1, \dots, B; B = \prod_{i=1}^n J_i\}, \quad (1.15)$$

где  $X_b$  – вариант сочетания алфавитов состояний  $x[t]$ ;

◆ конечный выходной алфавит

$$Y = \{C_{11}, \dots, C_{1J_1}\} \times \dots \times \{C_{L1}, \dots, C_{LJ_L}\} = \{t_g, g = 1, \dots, C; C = \prod_{p=1}^L J_p\}, \quad (1.16)$$

где  $t_g$  – вариант сочетания алфавитов выходов  $y[t]$ .

Так как количество значений входов, состояний и выходов конечно, то функции переходов и выходов могут быть представлены в табличной форме. Таблицы переходов и выходов могут быть представлены таблицами с одним входом (табл. 1.1).

Все возможные значения входов для удобства можно записывать в столбец блоками. В этом случае каждому сочетанию алфавитов входов  $v[t]$  ставятся в соответствие все возможные комбинации состояний  $x_b$ .

Выход конечного автомата Мили определяется состоянием автомата в предыдущий момент времени и сигналом, поступившим на вход в данный момент. При представлении конечного автомата Мили таблицей с одним входом, выходы заносятся в столбец  $(y_1, \dots, y_l)$ .

Так как входы записаны блоками, а для каждого блока представлена полная комбинация состояний, то, учитывая, что выход у конечного автомата Мура зависит от состояний и не зависит от входов, для любого блока входов полная комбинация состояний повторяется. Следовательно, выход для любого блока – блок некоторой комбинации выходов  $t'_g$ , определяемый комбинацией состояний. Выходы конечного автомата Мура зависят от предыдущих состояний, поэтому при представлении автомата Мура таблицей с одним входом выходы заносятся в столбец  $(y'_1, \dots, y'_l)$ .

Таблицы переходов и выходов могут быть представлены таблицами с двумя входами (табл. 1.2 и 1.3).

У конечного автомата Мура нет таблицы выходов, аналогичной автомату Мили. Это объясняется тем, что выходы непосредственно зависят от состояний. Поэтому для автомата Мура таблица выходов сводится к строке состояний значений выходов соответствующих определенному сочетанию состояний, и располагается над значениями состояний таблицы переходов. Получена отмеченная таблица переходов конечного автомата Мура (табл. 1.4).

С каждым конечным автоматом Мура связан интерпретирующий его конечный автомат Мили, описываемый не отмеченной таблицей, а двумя таблицами с двумя входами. Таблица переходов автомата Мура аналогична таблице переходов автомата Мили (табл. 1.2). Интерпретирующая функция выходов определяется как суперпозиция функции переходов Мура с его функцией выходов.

Таблица выходов интерпретирующего автомата определяется по следующей схеме:

1. Из таблицы переходов (табл. 1.2) по сочетанию значений входов  $s_a$  и предыдущих состояний  $x'_b$  определяется новое сочетание состояний  $x_b$ .
2. Из отмеченной таблицы переходов (табл. 1.4) для сочетания  $x'_b$ , совпадающего с сочетанием  $x_b$ , определяется сочетание выходов  $t'_g$ .
3. В таблицу выходов интерпретирующего автомата для рассмотренных  $s_a$  и  $x'_b$  заносится сочетание выходов  $t_g$ , совпадающее с сочетанием  $t'_g$ .

Таблица 1.4

Отмеченная таблица переходов конечного автомата Мура

$y'_1, \dots, y'_l$		$\dots c'_{1j_1}, \dots, c'_{pj_p}, \dots, c'_{lj_l} (t'_g) \dots$		
$x'_1, \dots, x'_n$	$b'_{11}, \dots, b'_{k1}, \dots, b'_{n1}$	$\dots$	$b'_{1j_1}, \dots, b'_{kj_k}, \dots, b'_{nj_n}$	$\dots$
	$(x'_1)$		$(x'_b)$	$(x'_B)$
$y_1, \dots, y_m$	$a_{11}, \dots, a_{i1}, \dots, a_{m1}$		$c_{1j_1}, \dots, c_{pj_p}, \dots, c_{lj_l}$	
	$(S_1)$		$(t_g)$	
	$\dots$			
	$a_{1j_1}, \dots, a_{ij_i}, \dots, a_{mj_m}$			
	$(S_a)$			
	$\dots$			
	$a_{1j_1}, \dots, a_{ij_i}, \dots, a_{mj_m}$			
	$(S_A)$			

Конечные автоматы могут быть представлены с помощью матрицы соединений (табл. 1.5).

Таблица 1.5

Матрица соединений конечного автомата

$x_1, \dots, x_n$		$b_{11}, \dots, b_{k1}, \dots, b_{n1}$	$\dots$	$b_{1j_1}, \dots, b_{kj_k}, \dots, b_{nj_n}$	$\dots$	$b_{1j_1}, \dots, b_{kj_k}, \dots, b_{nj_n}$
$x'_1, \dots, x'_n$	$(x_1)$			$(x_b)$		$(x_B)$
	$(x'_1)$	$\dots$		$\dots$		$\dots$
	$(x'_b)$	$\dots$		$\frac{s_{a_1}}{t_{g_1}} \vee \dots \vee \frac{s_{a_i}}{t_{g_i}}$ или 0		$\dots$
	$(x'_B)$	$\dots$		$\dots$		$\dots$

Матрица соединений строится как квадратная матрица того же порядка  $B$ , что и число сочетаний автомата.

$$B = \prod_{i=1}^n j_k \cdot \tag{1.17}$$

Столбцы и строки матрицы соответствуют различным состояниям автомата. Элементами матрицы служат дроби, в числителе которых – значения входов, обеспечивающие заданный переход состояний, а в знаменателе – соответст-

вующие значения выходов. На пересечении строки  $x'_b$  и столбца  $x_b$  располагается сочетание значений входов  $S_a$ , вызывающее переход автомата из состояния  $x'_b$  в состояние  $x_b$ . В знаменателе указывается сочетание значений выходов  $t_g$ , которое появится при осуществлении данного перехода. Если указанных дробей несколько, то ставится их дизъюнкция, а если таких дробей нет, то ставится ноль. Для удобства составления таблицы соединений все возможные сочетания входов и выходов получают свои коды (номера или буквы), что позволяет не загромождать матрицу. Информация о кодах входов и выходов приводится отдельно.

### Представление конечных автоматов графами

Конечный автомат можно представить в виде ориентированного графа. Граф автомата представляет собой набор вершин, соответствующих различным состояниям автомата и соединяющих вершины дуг графа, соответствующих тем или иным переходам автомата (направление дуги соответствует переходу из одного состояния в другое).

Дуги соответствуют элементам матрицы соединений.

Граф интерпретирующего автомата Мили для автомата Мура совпадает с графом исходного автомата Мура. Свойством такого графа является то, что на всех дугах, сходящихся в некоторую вершину, буква выходного алфавита – одна и та же.

Конечный автомат можно представить графом другого типа – ориентированным деревом. Корнем этого дерева служит начальное состояние автомата (вершина первого ранга). Из нее проводится столько дуг, сколько букв во входном алфавите (дуги первого ранга помечаются так же, как в матрице соединений и графе автомата). Каждая дуга входит в вершину второго ранга. Из каждой вершины второго ранга проводятся, в свою очередь, дуги второго ранга, входящие в вершины третьего ранга и т.д. Тем самым, дерево автомата – ориентированный граф, в каждую вершину которого, кроме корня, входит одна дуга и который не содержит контуров. Примеры графов представлены на рис. 1.3-1.4, а дерева – на рис. 1.5.

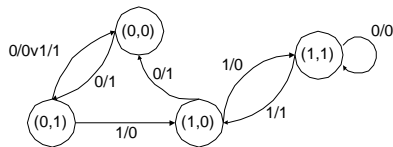


Рис. 1.3. Граф автомата Мили

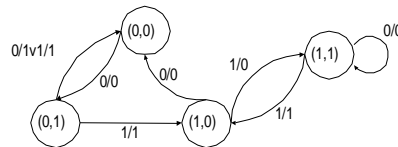


Рис. 1.4. Граф автомата Мура

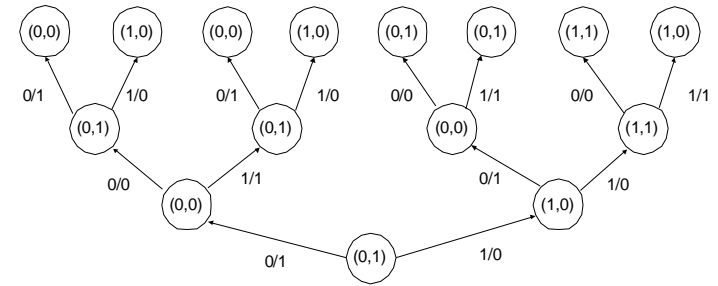


Рис. 1.5. Дерево автомата Мили

### Представление конечных автоматов при помощи многочленов Жегалкина

Состояния и выходы двоичных автоматов Мили и Мура удобно описывать с помощью многочленов Жегалкина [5-7]. Рассмотрим автомат с двумя состояниями и выходами и одним входом, и получим многочлены Жегалкина для каждого состояния и каждого выхода.

Для двоичного конечного автомата Мили многочлены Жегалкина будут иметь вид:

$$x_1 = a_0 \dot{A} a_1 \ddot{A} x'_1 \dot{A} a_2 \ddot{A} x'_2 \dot{A} a_3 \ddot{A} v \dot{A} a_{12} \ddot{A} x'_1 \ddot{A} x'_2 \dot{A} a_{13} \ddot{A} x'_1 \ddot{A} v \dot{A} a_{23} \ddot{A} \dot{A} x'_2 \ddot{A} v \dot{A} a_{123} \ddot{A} x'_1 \ddot{A} x'_2 \ddot{A} v; \quad (1.18)$$

$$x_2 = b_0 \dot{A} b_1 \ddot{A} x'_1 \dot{A} b_2 \ddot{A} x'_2 \dot{A} b_3 \ddot{A} v \dot{A} b_{12} \ddot{A} x'_1 \ddot{A} x'_2 \dot{A} b_{13} \ddot{A} x'_1 \ddot{A} v \dot{A} \dot{A} b_{23} \ddot{A} x'_2 \ddot{A} v \dot{A} b_{123} \ddot{A} x'_1 \ddot{A} x'_2 \ddot{A} v; \quad (1.19)$$

$$y_1 = c_0 \dot{A} c_1 \ddot{A} x'_1 \dot{A} c_2 \ddot{A} x'_2 \dot{A} c_3 \ddot{A} v \dot{A} c_{12} \ddot{A} x'_1 \ddot{A} x'_2 \dot{A} c_{13} \ddot{A} x'_1 \ddot{A} v \dot{A} c_{23} \ddot{A} \dot{A} x'_2 \ddot{A} v \dot{A} c_{123} \ddot{A} x'_1 \ddot{A} x'_2 \ddot{A} v; \quad (1.20)$$

$$y_2 = d_0 \dot{A} d_1 \ddot{A} x'_1 \dot{A} d_2 \ddot{A} x'_2 \dot{A} d_3 \ddot{A} v \dot{A} d_{12} \ddot{A} x'_1 \ddot{A} x'_2 \dot{A} d_{13} \ddot{A} x'_1 \ddot{A} v \dot{A} \dot{A} d_{23} \ddot{A} x'_2 \ddot{A} v \dot{A} d_{123} \ddot{A} x'_1 \ddot{A} x'_2 \ddot{A} v. \quad (1.21)$$

Для двоичного конечного автомата Мура многочлены Жегалкина для состояний аналогичны многочленам для автомата Мили. Для выходов автомата Мура многочлены будут иметь вид:

$$y_1 = c'_0 \dot{A} c'_1 \ddot{A} x_1 \dot{A} c'_2 \ddot{A} x_2 \dot{A} c'_{12} \ddot{A} x_1 \ddot{A} x_2; \quad (1.22)$$

$$y_2 = d'_0 \dot{A} d'_1 \ddot{A} x_1 \dot{A} d'_2 \ddot{A} x_2 \dot{A} d'_{12} \ddot{A} x_1 \ddot{A} x_2. \quad (1.23)$$

## 1.2. Описание технологии с помощью конечных автоматов

Технологический процесс характеризуется многостадийностью обработки – от сырья до готовой продукции, при этом полуфабрикат проходит множество технологических переделов.

В общем случае производственный процесс можно представить в виде конечного автомата, который имеет несколько входов –  $V$  (сырье, полуфаб-



рикат), состояний –  $X$  (технологических параметров по каждому агрегату) и выходов –  $Y$  (свойств готовой продукции).

Вектор, составленный из скалярных входов  $V[t] = (V_1[t], \dots, V_L[t])$ , где  $L$  – число входов, аналогично для состояний (технологических факторов):

$$x[t] = (x_{11}[t], \dots, x_{1M_1}[t], x_{21}[t], \dots, x_{2M_2}[t], x_{k1}[t], \dots, x_{kM_k}[t])$$

и выходов:

$$y[t] = (y_1[t], \dots, y_R[t]),$$

где  $M_i$  – число технологических факторов  $i$ -го передела;  $K$  – число технологических переделов;  $R$  – число показателей качества.

Функции переходов и выходов допускают развернутую запись:

$$x_{ij}[t] = j_i(x_{11}[t], \dots, x_{1M_1}[t], x_{21}[t], \dots, x_{2M_2}[t], \dots, x_{(i-1),1}[t], \dots, x_{(i-1),M_{i-1}}[t], v_1[t], \dots, v_L[t]). \quad (1.24)$$

Последующие:  $x_{(i+1),1}[t], \dots, x_{(i+1),M_{i+1}}[t], \dots, x_{K1}[t], \dots, x_{kM_k}[t]$  считаем равными нулю, т.к. процесс еще не реализован (рис. 1.6).

Можно исследовать безусловные вероятности, с которыми каждый из входов  $s_a \in V$  может оказаться начальным. Это позволяет оценить характеристики поступающего сырья с помощью вероятностей и законов распределения случайных величин.

Алфавиты входов, состояний и выходов могут иметь различную значность (количество диапазонов изменения).

Пусть  $L$  – количество характеристик, образующих входной сигнал. Каждая входная величина может иметь свою значность (количество дискретных непересекающихся значений). В качестве примера входной величины можно взять один из элементов химического состава сляба. Пусть содержание углерода в стали изменяется в интервале от 0,05 до 0,08%. Если разбить на составляющие  $\{0,05; 0,06; 0,07; 0,08\}$ , то получим значность алфавита равным 4. Допустим, что это первая входная величина, тогда ( $J_1 = 4$ ), а  $a_{i1}$  –  $i$ -я составляющая алфавита 1-го входа. Таким образом, конечный входной алфавит (см. 1.15):

$$V = \{a_{11}, \dots, a_{1J_1}\} \times \dots \times \{a_{L1}, \dots, a_{LJ_L}\} = \{s_a, a = 1, \dots, A; A = \prod_{i=1}^L J_i\}, \quad (1.25)$$

где  $s_a$  – вариант сочетания алфавитов входов  $v[t]$  любого передела.

В более общем случае каждый агрегат характеризуется набором технологических величин, определяющих процесс обработки, сырьём и затратами на агрегате.

Технологический процесс может осуществляться в виде параллельных и последовательных потоков (рис. 1.6 и 1.7).

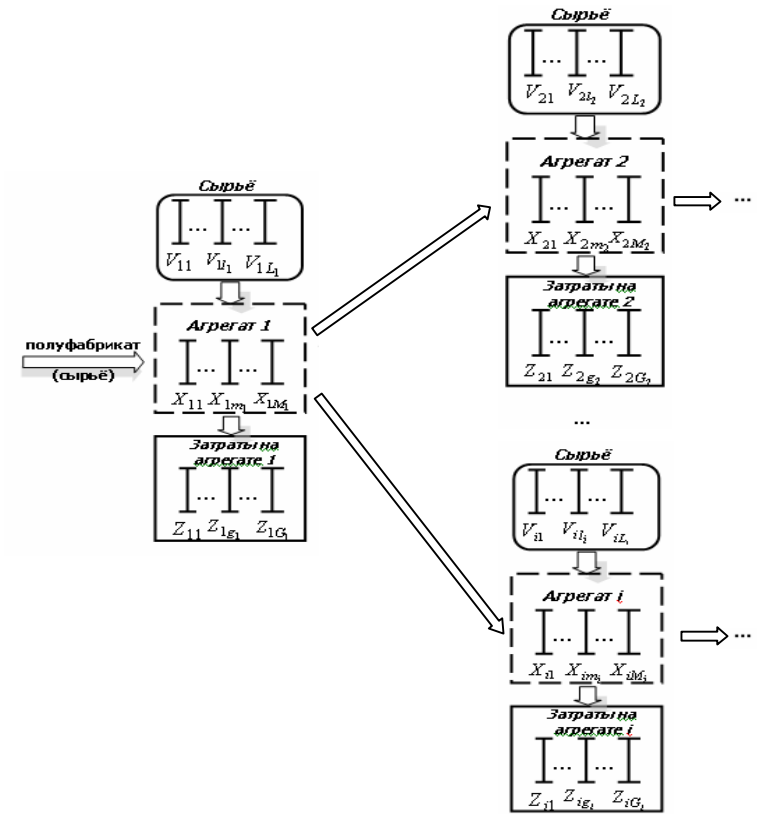


Рис. 1.6. Представление технологического процесса в виде параллельных потоков

$X_{imi}$  –  $m_i$  технологический фактор  $i$ -го передела (агрегата);  $M_i$  – количество технологических факторов на  $i$ -м переделе;  $Z_{ig_i}$  –  $g_i$  элемент затрат  $i$ -го передела (агрегата);  $G_i$  – количество элементов затрат на  $i$ -м переделе;  $V_{il_i}$  –  $l_i$ -я входная величина  $i$ -го передела (агрегата);  $L_i$  – количество входных элементов (сырьё)

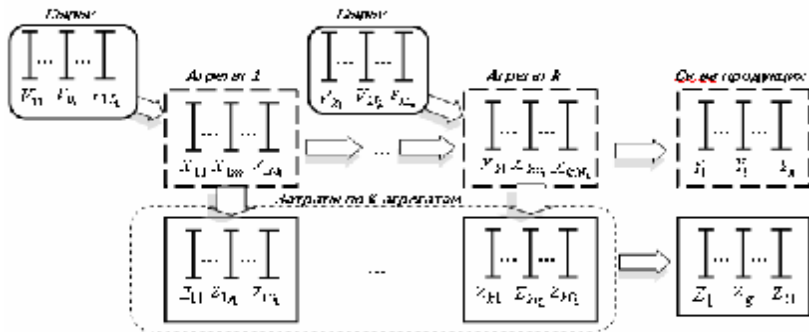


Рис. 1.7. Представление технологического процесса в виде последовательных потоков

$i = 1, \dots, K$  – число агрегатов;  $l = 1, \dots, R$  – число показателей качества;  $V_{in_j}$  –  $n_j$ -й вход для  $i$ -го агрегата,  $j = 1, \dots, N$ ;  $X_{im_j}$  –  $m_j$ -й технологический фактор  $i$ -го агрегата,  $j = 1, \dots, M$ ;  
 $Z_{ig_j}$  –  $g_j$ -й элемент затрат на  $i$ -м агрегате,  $j = 1, \dots, G$ ;  $Y_l$  –  $l$ -й показатель качества

Так как в общем случае производственный процесс, представленный в виде конечного автомата, имеет несколько входов, состояний (технологических факторов), разделенных по переделам или агрегатам, и выходов, а алфавиты входов, состояний и выходов могут иметь разную значность, то их составляющие можно представить следующим образом:

$k = 1, \dots, K$  – номер передела (агрегата);

$a_{klk_j l_k}$  – составляющая алфавитов входов, где  $j_{l_k} = 1, \dots, J_{l_k}$ , где

$J_{l_k}$  значность алфавита  $l$ -го входа на  $k$ -м агрегате;

$l_k = 1, \dots, L_k$  – номер входа (элементов сырья, полуфабрикатов) на  $k$ -м агрегате;

$J_{l_k}$  – значность алфавита  $l$ -го входа на  $k$ -м агрегате;

$b_{k,m_k j_{m_k}}$  – составляющая алфавитов состояний (технологических факторов) для  $k$ -го агрегата;

$j_{m_k} = 1, \dots, J_{m_k}$ , где  $J_{m_k}$  – значность алфавита  $m$ -го технологического фактора на  $k$ -м агрегате;

$m_k = 1, \dots, M_k$  – номера факторов на  $k$ -м переделе;

$C_{rj_r}$  – составляющая алфавитов  $r$ -го выхода;

$j_r = 1, \dots, J_r$  – значность алфавита  $p$ -го выхода;

$r = 1, \dots, R$  – номер выхода.

Таким образом, конечный входной алфавит:

$$V = \{a_{111}, \dots, a_{1J_1}\} \times \dots \times \{a_{lL_1}, \dots, a_{lJ_{L_1}}\} \times \dots \times \{a_{k11}, \dots, a_{kJ_1}\} \times \dots \times \{a_{kL_k}, \dots, a_{kJ_{L_k}}\} =$$

$$= \{s_a, a_k = 1, \dots, A_k; A_k = \prod_{i=1}^{L_k} J_i\} \quad (1.26)$$

где  $s_a$  – вариант сочетаний алфавитов входов  $V[t]$ ,  $A_k$  – максимальное количество сочетаний алфавитов входов на  $k$ -м агрегате.

Конечный внутренний алфавит (алфавит технологических факторов)

$$X = \{b_{111}, \dots, b_{1J_1}\} \times \dots \times \{b_{M_11}, \dots, b_{M_1J_{M_1}}\} \times \dots \times \{b_{k11}, \dots, b_{kJ_1}\} \times \dots \times \{b_{kM_k1}, \dots, b_{kM_kJ_{M_k}}\} =$$

$$= \{x_{b_k}, b_k = 1, \dots, B_k; B_k = \prod_{i=1}^{M_k} J_i\} \quad (1.27)$$

где  $x_{b_k}$  – вариант сочетания алфавитов на  $k$ -м агрегате;  $K$  – число агрегатов;  $m_k$  – число факторов на  $k$ -м агрегате.

Вариант сочетания алфавитов на 1-м агрегате:

$$x_{b_1} = \{b_{111}, \dots, b_{1J_1}\} \times \dots \times \{b_{1M_11}, \dots, b_{1M_1J_1}\} = \{x_{b_1}, b_1 = 1, \dots, B_1; B_1 = \prod_{i=1}^{M_1} J_i\} \quad (1.28)$$

Вариант сочетания алфавитов на  $k$ -м агрегате:

$$x_{b_k} = \{b_{k11}, \dots, b_{kJ_1}\} \times \dots \times \{b_{kM_k1}, \dots, b_{kM_kJ_{M_k}}\} = \{x_{b_k}, b_k = 1, \dots, B_k; B_k = \prod_{i=1}^{M_k} J_i\} \quad (1.29)$$

Вариант сочетания алфавитов на последнем агрегате:

$$x_{b_K} = \{b_{K11}, \dots, b_{KJ_1}\} \times \dots \times \{b_{KM_k1}, \dots, b_{KM_kJ_{M_k}}\} = \{x_{b_K}, b_K = 1, \dots, B_K; B_K = \prod_{i=1}^{M_K} J_i\} \quad (1.30)$$

$s_a$  – вариант сочетания алфавитов  $V[t]$  на всех агрегатах:

$$s_a, a = 1, \dots, A; A = \prod_{i=1}^K A_i, \quad (1.31)$$

$x_b$  – вариант сочетания алфавитов  $x[t]$  на всех агрегатах:

$$x_b, b = 1, \dots, B; B = \prod_{i=1}^K B_i. \quad (1.32)$$

Конечный выходной алфавит:

$$Y = \{c_{11}, \dots, c_{1J_1}\} \times \dots \times \{c_{R1}, \dots, c_{RJ_R}\} = \{t_g, g = 1, \dots, G; G = \prod_{i=1}^R J_i\}, \quad (1.33)$$

где  $t_g$  – вариант сочетания алфавитов входов  $y[t]$ .

На каждом агрегате и переделе можно оценить затраты на производство. Статьи затрат определяются калькуляцией для каждого передела. Обозначим  $z_{ig_i}$  –  $g$ -й элемент затрат на  $i$ -м агрегате. Так как затраты изменяются в зависимости от сортамента и условий производства, то их также можно описать с помощью конечных автоматов.

Пусть  $d_{kgkjg_k}$  – составляющая алфавита  $g$ -го элемента затрат на  $k$ -м агрегате.

$j_{gk} = 1, \dots, J_{gk}$ , где  $J_{gk}$  – значность алфавита  $g$ -го элемента затрат на  $k$ -м агрегате.

$g_k = 1, \dots, G_k$  – номера элементов затрат на  $k$ -м агрегате (переделе).

В соответствии с рис. 1.6 разложение входов и состояний (технологических факторов) по агрегатам, элементов затрат и выходных показателей можно представить в виде, представленном на рис. 1.7.

Как отмечалось ранее, каждая входная величина может иметь свою значность (количество непересекающихся диапазонов), аналогичные выводы можно сделать о состояниях (технологических факторах по переделам или агрегатам) и выходах. Таким образом, алфавиты входов, состояний и выходов могут иметь различную значность.

Для формирования значности алфавитов каждого входа, технологического фактора или выхода необходимо рассмотреть законы распределения изучаемых величин. Если исследуемая величина может быть представлена в виде вариационного ряда, то значность ее алфавита будет принимать значения из соответствующего вариационного ряда. В случае если величина подчиняется какому-либо закону распределения и может быть представлена в виде гистограммы, то необходимо сформировать алфавит этой величины таким образом, чтобы она принимала значения из этого внутреннего алфавита. Наиболее простой способ – разбить диапазоны изменения исследуемого фактора на равные участки, однако в этом случае вероятности попадания в каждый из них могут существенно различаться, т.е. получим неравномерно

распределенный алфавит данной величины (алфавит с различной частотой появления отдельных составляющих).

При необходимости получения диапазонов, вероятность попадания в которые примерно одинакова (равномерный алфавит), отрезок изменения исследуемой величины разбивается на ряд неодинаковых по длине полуинтервалов.

Для примера возьмем  $m_k$ -ю технологическую величину  $k$ -го передела ( $X_{km_k}$ ). Отрезок изменения входной величины разбивается на ряд полуинтервалов:  $b_{km_k 1}, b_{km_k 2}, \dots, b_{km_k j_{m_k}}, \dots, b_{km_k J_{m_k}}$ , где  $k$  – номер передела,  $m_k$  – фактор  $r$ -го передела,  $j_{m_k} = 1, \dots, J_{m_k}$  – полуинтервалы, образующие алфавит данного фактора). Численные значения  $j_{m_k}$  выбираются таким образом, чтобы любому полуинтервалу принадлежало достаточное число точек (опытов), полученных при исследовании значений технологической величины  $X_{km_k}$ .

Обозначим количество попаданий в полуинтервал  $n_{km_k j_{m_k}}$ , причем:

$$\sum_{j_{m_k}=1}^{J_{m_k}} n_{km_k j_{m_k}} = n, \quad (1.34)$$

где  $n$  – объем выборки.

Так как количество значений входов, состояний и выходов конечны, то функции переходов (из одного состояния в другое) могут быть представлены табличной форме. При этом можно использовать 2 варианта таблиц.

Первый вариант состоит в представлении перехода с предыдущего состояния в последующее с учетом входных воздействий, т.е. по входным данным для 1-го передела определяется состояние на данном переделе ( $V_1 \text{ передел} \rightarrow X_1 \text{ передел}$ ), затем по данным о состоянии на первом переделе и входах для второго передела определяются состояния на 2-м переделе ( $V_2 \text{ передел}, X_1 \text{ передел} \rightarrow X_2 \text{ передел}$ ) и т.д. В этом случае все значения входов можно представить в таблице в виде блоков, записываемых в столбец. И каждому сочетанию алфавитов входов  $V_k$  ставятся в соответствие все возможные комбинации состояний  $x_{b_{k-1}}$ . Таким образом, для  $k$ -го передела таблица переходов будет иметь вид, представленный в табл. 1.20.

Таблица 1.20

Таблица переходов для k-го передела с одним входом

$V_{k1}, \dots, V_{kL_k}$	$x_{k-1,1}, \dots, x_{k-1, M_{k-1}}$	$x_{k1}, \dots, x_{k, M_k}$
$a_{k11} \dots a_{kl_1} \dots a_{kL_k 1}$ $s_{1(k)}$	$b_{(k-1)11}, \dots, b_{(k-1)m_{k-1}1}, \dots, b_{(k-1)M_{k-1}1}$ $(x_{1(k-1)})$	
.....	.....	
$s_{1(k)}$	$b_{(k-1)1j_1}, \dots, b_{(k-1)m_{k-1}j_{m_{k-1}}}, \dots, b_{(k-1)M_{k-1}j_{M_{k-1}}}$ $(x_{b_{1(k-1)}})$	
.....	.....	
	$b_{(k-1)1J_1}, \dots, b_{(k-1)m_{k-1}J_{m_{k-1}}}, \dots, b_{(k-1)M_{k-1}J_{M_{k-1}}}$ $(x_{B_{1(k-1)}})$	
.....	.....	
	$(x_{1(k-1)})$	.....
$a_{k1j_1} \dots a_{kl_j k} \dots a_{kL_k j_{L_k}}$ $s_{a(k)}$	$(x_{a_{1(k-1)}})$	$b_{k1j_1} \dots b_{km_{k-1}j_{m_k}} \dots b_{kM_k j_{M_k}}$ $(x_{b_k})$
.....	$(x_{B_{1(k-1)}})$	.....
.....	.....	
	$(x_{1(k-1)})$	
$a_{k1J_1} \dots a_{kl_j J_k} \dots a_{kL_k J_{L_k}}$ $s_{A(k)}$	$(x_{b_{1(k-1)}})$	
	$(x_{B_{1(k-1)}})$	

Более наглядный вид даст таблица переходов с двумя входами (имеются в виду входы таблиц) (табл. 1.21).

Таблица 1.21

Таблица переходов для k-го передела с двумя входами

$x_{k-1,1}, \dots, x_{k-1, M_{k-1}}$	$b_{(k-1)11}, \dots, b_{(k-1)m_{k-1}1}, \dots, b_{(k-1)M_{k-1}1}$ $(x_{1(k-1)})$	.....	$b_{(k-1)1j_1}, \dots, b_{(k-1)m_{k-1}j_{m_{k-1}}}, \dots, b_{(k-1)M_{k-1}j_{M_{k-1}}}$ $(x_{b_{1(k-1)}})$	.....	$b_{(k-1)1J_1}, \dots, b_{(k-1)m_{k-1}J_{m_{k-1}}}, \dots, b_{(k-1)M_{k-1}J_{M_{k-1}}}$ $(x_{B_{1(k-1)}})$
$v_{k1}, \dots, v_{kL_k}$					
$a_{k11} \dots a_{kl_1} \dots a_{kL_k 1}$ $s_{1(k)}$					
.....					
$s_{1(k)}$					
.....					
$a_{klj_1} \dots a_{kl_j j_k} \dots a_{kL_k j_{L_k}}$ $s_{a(k)}$					
.....					
$s_{a(k)}$					
.....					
$a_{k1J_1} \dots a_{kl_j J_k} \dots a_{kL_k J_{L_k}}$ $s_{A(k)}$					

В табл. 1.20-1.21 представлена зависимость текущих состояний от предыдущих и входных воздействий. При этом предшествующими считаются состояния (факторы), реализованные на предшествующем агрегате.

Более полное представление может дать включение всех ранее реализованных факторов (состояний) и входных воздействий в таблицу. Накопленные на предыдущих агрегатах и переделах факторы можно представить в табл. 1.22.

Таблица 1.22

Таблица переходов с накопленными на предыдущих агрегатах и переделах факторами

$V_{11} \dots V_{1L_1}$	$V_{21} \dots V_{2L_2}$	.....	$V_{(k+1)1} \dots V_{(k+1)L_{(k+1)}}$	$X_{11} \dots X_{1M_1}$	$X_{21} \dots X_{2M_2}$	.....	$X_{k1} \dots X_{kM_k}$	.....	$X_{(k+1)1} \dots X_{(k+1)M_{(k+1)}}$
$s_{1(1)}$	$s_{2(2)}$		$s_{(k+1)}$	$x_{1(1)}$	$x_{1(2)}$		$x_{1(k)}$		$x_{B_{(k+1)}}$
<b>M</b>	<b>M</b>		<b>M</b>	<b>M</b>	<b>M</b>		<b>M</b>		
$s_{A(1)}$	$s_{A(2)}$		$s_{A(k+1)}$	$x_{B_{(1)}}$	$x_{B_{(2)}}$		$x_{B_{(k)}}$		

Общее количество строк в такой таблице, являющейся полным перебором сочетаний алфавитов входов и состояний на всех рассмотренных пределах велико и составляет

$$D = \prod_{i=1, \dots, k+1} S_a \times \prod_{i=1, \dots, k} X_b \quad (1.35)$$

Таким образом, полное заполнение таблицы практически невозможно из-за отсутствия в реальных условиях производства некоторых сочетаний алфавитов.

Однако анализ реального производства позволяет выделить все основные комбинации входов и состояний при различных сочетаниях алфавитов. Эти сочетания и должны быть вынесены в данные таблицы переходов. Причем, использование такого представления дает наглядную картину реализации производственных процессов с выделением их основных траекторий.

При этом, после выделения всех встречающихся сочетаний алфавитов, которые можно назвать технологическими траекториями, число опытов  $n_{SX}$ , приходящихся на каждую траекторию, может изменяться в пределах

$$0 < n_{SX} \leq M, \quad (1.36)$$

где  $M$  – число опытов.

В то же время, число сочетаний алфавитов  $T$ , для которых  $n_{SX} > 0$  (при  $n_{SX} = 0$  сочетание автоматически исключается из таблицы) изменяется от 1 до  $M$ , но не более  $D$ .

Аналогичные подходы можно использовать при составлении таблиц входов, отражающих влияние входных воздействий и технологических параметров (состояний) на выходные характеристики. В качестве выходных величин могут выступать регламентируемые стандартами свойства полуфабриката. Также в качестве таких величин можно рассматривать элементы затрат, расходуемых на данном переделе (рис. 1.8).

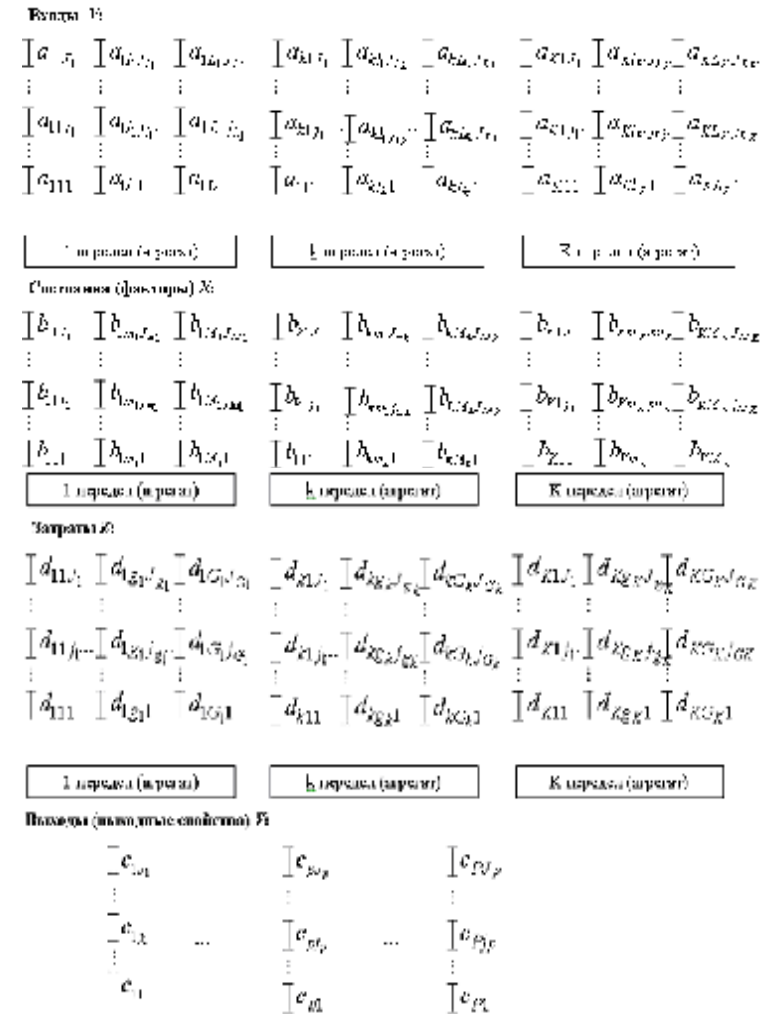


Рис. 1.8. Описание технологии с помощью конечных автоматов

Примечания к рис. 1.8.

1). Значение индексов для входов и состояний (факторов): 1-я цифра – передел, 2-я – номер фактора, 3-я – номер алфавита соответствующего фактора; для выходов: 1-я цифра – номер выходной величины, 2-я – номер алфавита соответствующей выходной величины.

2). При исследовании выходов после каждого агрегата или передела появится 3-я цифра – передела (аналогично входам, состояниям и затратам).

В итоге формируются табл. 1.23-1.25, аналогичные таблицам переходов.

Таблица 1.24

Таблица выходов с двумя входами для  $k$ -го передела

$x_{k-1,1}, \dots, x_{k-1,M_{k-1}}$	$(x_{l_{(k-1)}})$	...	$(x_{b_{(k-1)}})$	...	$(x_{B_{(k-1)}})$
$v_{k1}, \dots, v_{kL_k}$					
$s_{l_{(k)}}$	<b>O</b>		<b>M</b>		<b>N</b>
...	$c_{k1j_1}, \dots, c_{k r_k j_{r_k}}, \dots, c_{k R_k j_{R_k}}$				
$s_{a_{(k)}}$	<b>K</b>	$t_{g^{(k)}}$			<b>K</b>
...	$d_{k_1 j_{1k}}, \dots, d_{k g_k j_{gk}}, \dots, d_{k G_k j_{Gk}}$				
$s_{A_{(k)}}$	<b>N</b>		<b>J</b>	$l_{(k)}$	<b>O</b>

Таблица 1.25

Таблица выходов после  $k$ -го передела

$V_{11} \dots V_{1L}$	$V_{21} \dots V_{2L_2}$		$V_{(k+1)1} \dots V_{(k+1)M_{(k+1)}}$	$X_{11} \dots X_{1M_1}$	$X_{21} \dots X_{2M_2}$		$X_{k1} \dots X_{kM_k}$	$Y_{k1} \dots Y_{kR_k}$	$\sum_{i=1}^K Z_1 \dots \sum_{i=1}^K Z_G$
$s_{l_{(1)}}$	$s_{l_{(2)}}$		$s_{l_{(k+1)}}$	$x_{l_{(1)}}$	$x_{l_{(2)}}$		$x_{l_{(k)}}$	$t_{g^{(k)}}$	$\sum_{i=1}^k z_{l_{(i)}}$
<b>M</b>	<b>M</b>		<b>M</b>	<b>M</b>	<b>M</b>		<b>M</b>		$\sum_{i=1}^k z_{l_{iR}}^* \dots \sum_{i=1}^k z_{iR_i}^* \dots$
$s_{A_{(1)}}$	$s_{A_{(2)}}$		$s_{A_{(k+1)}}$	$x_{B_{(1)}}$	$x_{B_{(2)}}$		$x_{B_{(k)}}$		$\dots \sum_{i=1}^k z_{iG_i}^*$

Примечание к табл. 1.25.

$$s_{l_{(k)}} = a_{k1l_1} \dots a_{kl_k l_k} \dots a_{kL_k l_k};$$

$$s_{a_{(k)}} = a_{k1j_1} \dots a_{kl_k j_k} \dots a_{kL_k j_k};$$

$$s_{A_{(k)}} = a_{k1J_1} \dots a_{kl_k J_k} \dots a_{kL_k J_k};$$

$$(x_{l_{(k-1)}}) = b_{(k-1)11} \dots b_{(k-1)m_{k-1}1} \dots b_{(k-1)M_{k-1}1};$$

$$(x_{b_{(k-1)}}) = b_{(k-1)1j_{k-1}} \dots b_{(k-1)m_{k-1}j_{mk-1}} \dots b_{(k-1)M_{k-1}j_{Mk-1}};$$

$$(x_{B_{(k-1)}}) = b_{(k-1)1J_{k-1}} \dots b_{(k-1)m_{k-1}J_{mk-1}} \dots b_{(k-1)M_{k-1}J_{Mk-1}};$$

$$\begin{aligned} (x_{1_k}) &= b_{k11}, \dots, b_{km_{k-1}}, \dots, b_{kM_{k-1}}; \\ (x_{b_k}) &= b_{k1j_{1k-1}}, \dots, b_{km_{k-1}j_{mk-1}}, \dots, b_{kM_{k-1}j_{Mk-1}}; \\ (x_{B_{(k-1)}}) &= b_{k1j_{1k-1}}, \dots, b_{km_{k-1}j_{mk-1}}, \dots, b_{kM_{k-1}j_{Mk-1}} \\ \tau_{\gamma^{(k)}} &= c_{k1j_1}, \dots, c_{krj_r}, \dots, c_{kRj_R} \\ \sum_{i=1}^k j_{i(i)}^* &= \sum_{i=1}^k d_{i1j_{1i}}^*, \dots, \sum_{i=1}^k d_{iG_i j_{G_i}}^*, \dots, \sum_{i=1}^k d_{iG_i j_{G_i}}^* \end{aligned}$$

где  $j_{g_i}^*$  – номер использованного выходного алфавита затрат на  $i$ -м перделе для  $g$ -го элемента.

При описании технологического процесса можно использовать матрицы соединений. Матрица соединений строится как квадратная матрица того же порядка  $B$ , что и число состояний технологического процесса, представленного в виде автомата. Однако, так как не все состояния реализуемы, то получим матрицу меньшего порядка  $T$ .

Столбцы и строки матрицы соответствуют различным состояниям автомата. Элементами такой матрицы служат дроби, в числителе которых – значения входов, обеспечивающие заданный переход состояний, а в знаменателе – соответствующие значения выходов (табл. 1.26). На пересечении строки  $x_{b_{(k-1)}}$  и столбца  $x_{b_{(k)}}$  располагается сочетание значений входов

$S_{a_k}$ , вызывающее переход автомата из состояния  $x_{b_{(k-1)}}$  в состояние  $x_{b_{(k)}}$ . В знаменателе указывается сочетание значений выходов  $t_{gk}$  и  $j_{1k}$ , которое появится при осуществлении данного перехода. Если указанных дробей несколько (т.е. если из одного состояния  $(x_{b_{(k-1)}})$  в другое  $(x_{b_{(k)}})$

можно перейти при различных сочетаниях входов  $S_{a_k}$ , то ставится их дизъюнкция, а если таких переходов нет, то ставится ноль. Для удобства составления таблиц соединений все возможные сочетания входов и выходов можно представить в виде кодов, состояний из номеров или букв, что позволяет не загромождать матрицу. Информация о кодах входов и выходов может быть приведена в отдельной таблице (табл. 1.26).

Матрица соединений для  $k$ -го пердела

$x_{k,1}, \dots, x_{k,M_{k-1}}$ $x_{k-1,1}, \dots, x_{k-1,M_{k-1}}$	$(x_{1_{(k)}})$		$(x_{b_{(k)}})$		$(x_{B_{(k)}})$
$(x_{1_{(k-1)}})$ ...	<b>K</b>	<b>O</b>	<b>M</b> $S_{a_{(k)}}$	<b>N</b>	<b>K</b>
$(x_{b_{(k-1)}})$ ...		$(a_{k1j_1} \dots a_{klj_k} \dots a_{kLj_k})$ $t_{g^{(k)}}; j_{1_{(k)}}$			
$(x_{B_{(k-1)}})$	<b>N</b>	$(c_{k1j_1} \dots c_{krj_r} \dots c_{kRj_R})$	$(d_{k1j_{1k}} \dots c_{kG_k j_{G_1}})$	<b>M</b>	<b>O</b>

Примечание к таблице 1.26.

$$S_{a_{(k)}} = a_{k1j_1} \dots a_{klj_k} \dots a_{kLj_k};$$

$$(x_{1_{(k-1)}}) = b_{(k-1)11}, \dots, b_{(k-1)m_{k-1}1}, \dots, b_{(k-1)M_{k-1}1};$$

$$(x_{b_{(k-1)}}) = b_{(k-1)1j_{1k-1}}, \dots, b_{(k-1)m_{k-1}j_{mk-1}}, \dots, b_{(k-1)M_{k-1}j_{Mk-1}};$$

$$(x_{B_{(k-1)}}) = b_{(k-1)1j_{1k-1}}, \dots, b_{(k-1)m_{k-1}j_{mk-1}}, \dots, b_{(k-1)M_{k-1}j_{Mk-1}}$$

$$(x_{1_k}) = b_{k11}, \dots, b_{km_{k-1}1}, \dots, b_{kM_{k-1}1};$$

$$(x_{b_k}) = b_{k1j_{1k-1}}, \dots, b_{km_{k-1}j_{mk-1}}, \dots, b_{kM_{k-1}j_{Mk-1}};$$

$$(x_{B_{(k-1)}}) = b_{k1j_{1k-1}}, \dots, b_{km_{k-1}j_{mk-1}}, \dots, b_{kM_{k-1}j_{Mk-1}}$$

Чтобы сформировать алфавит каждого фактора можно воспользоваться одним из подходов:

1) *деление по среднему*.

Для исследуемого фактора находится среднее значение по выборке, которое можно определить до ближайшего фактического значения данного фактора. В результате чего все опыты будут разбиты приблизительно на две одинаковые по объему части. Затем каждый из полуинтервалов разбивается на два новых подинтервала. Недостатком данного подхода является кратное изменение количества полуинтервалов, а следовательно, получаемых элементов алфавита (2, 4, 8 и т.д.);

2) *деление на заданное количество полуинтервалов*.

Разделив объем выборки на задаваемое количество полуинтервалов, определяем среднее число опытов, приходящихся на каждый полуинтервал  $M$ .

Затем осуществляется нормирование всех элементов выборки. «Двигаясь» по выборке от минимального к максимальному значению и расширяя границы полуинтервала, суммируя все опыты, попадающие в данный полуинтервал. При достижении среднего числа опытов, приходящихся на полуинтервал, фиксируем его границы и переходим к следующему. В итоге формируются полуинтервалы, с помощью которых определяется значность алфавита данного фактора. Если какое-либо значение фактора фиксируется многократно и число его появлений близко к  $M(X_{km_k})$ , то данное значение фактора может выступать в роли элементов алфавита, как для вариационного ряда.

### 1.3. Вероятностные конечные автоматы (ВКА) (P-схемы). (Дискретно-стохастические модели)

В общем виде вероятностный автомат можно определить как дискретный потактный преобразователь информации с памятью, функционирование которого в каждом тексте зависит только от состояния памяти в нем и может быть описано стохастически [8-10].

Ранее конечный автомат был рассмотрен как детерминированный, т.е. каждой паре  $(x_b, s_a)$  (значений состояния и поступившего входа) однозначно ставилась в соответствие пара  $(x'_b, t_g)$  (новое состояние и выход) с помощью функций переходов и выходов.

Вероятностный конечный автомат – такой автомат, который не относит однозначно каждой паре  $(x'_b, s_a)$  некоторую из пар  $(x'_b, t_g)$ , а задает лишь условные вероятности появления пар  $(x'_b, t_g)$ , при условии, что реализовалась пара  $(x'_b, s_a)$  ( $s_a$  – вариант сочетания алфавитов входов  $u[t]$ ,  $x'_b, x'_b$  – варианты сочетания алфавитов состояний  $x[t]$  в предыдущий и последующие моменты времени соответственно;  $t_g$  – вариант сочетания алфавитов выходов  $y[t]$ ):

$$P(x'_b, t_g / x'_b, s_a) = P^{<b', a>}_{b, i}, \quad (1.37)$$

где  $P^{<b', a>}_{b, i}$  – вероятность перехода автомата в состояние  $x'_b$  и появления выходного сигнала  $t_g$  при условии, что автомат находится в состоянии  $x'_b = x'_b$  и на его вход поступил сигнал  $s_a$ .

Кроме того, для вероятностного конечного автомата не задается однозначно начальное состояние  $x_{(0)}$ , а задаются лишь безусловные вероятности, с которыми каждое из состояний  $x'_b$  может оказаться начальным:

$$P_{(0)}(x_b) = P_{b, (0)}. \quad (1.38)$$

Последнее условие задает начальное безусловное распределение вероятностей на конечном внутреннем алфавите  $X$  (табл. 1.27).

При этом должны выполняться требования:

$$1) \quad 0 \leq R_{b(0)} \leq 1; \quad (1.39)$$

$$2) \quad \sum_{x_b \in C} R_{b(0)} = 1. \quad (1.40)$$

Таблица 1.27

Вероятности, с которыми каждое из состояний  $x_b \in X$  может оказаться начальным

$X$	$x_1, \dots, x_b, \dots, x_B$
$P_{(0)}$	$P_{1, (0)}, \dots, R_{b(0)}, \dots, P_{B, (0)}$

При этом должны выполняться требования:

$$1) \quad 0 \leq R_{b(0)} \leq 1; \quad (1.41)$$

$$2) \quad \sum_{x_b \in X} P_{b, (0)} = 1. \quad (1.42)$$

$P_{b, (0)}$  может равняться 0 или 1 при некотором  $x_b \in X$ , и в силу (1.42) только одна вероятность равна 1, а все остальные равны 0.

Описание функционирования вероятностного конечного автомата можно трактовать так, что для каждой пары  $(x'_b, s_a)$  из множества  $X \times V$  задается совместное условное распределение вероятностей. Так как внутренний алфавит  $X$  и выходной алфавит  $U$  конечны, то имеем пару дискретных случайных величин, которая должна быть задана своим совместным законом распределения. Число таких распределений, представляемых в виде табл. 1.28, равно числу возможных пар  $(x'_b, s_a)$ . При этом должны выполняться требования:

$$1) \quad 0 \leq P^{(ba)}_{ba} \leq 1; \quad (1.43)$$

$$2) \quad \sum_{x_b \in C} \sum_{t_g \in U} P^{(ba)}_{ba} = 1. \quad (1.44)$$

Для полного описания вероятностного конечного автомата следует задать такие законы распределения для всех пар  $(x'_b, s_a)$ . Конечный автомат можно рассматривать как частный случай вероятностного конечного автомата, где соответствующие распределения вероятностей будут вырожденными (т.е. в ряду вероятностей лишь одна 1, остальные 0).



Таблица 1.28

Совместный закон распределения состояний и выходов

		$P_g$		
		$P_1^{ba}, \dots, P_g^{ba}, \dots, P_c^{ba}$		
$P_b$	$X$	$Y$	$t_1, \dots, t_g, \dots, t_c$	
	$X_1$	$X_b$	$X_B$	
$P_1^{\beta\alpha}$	$\vdots$	$P_{11}^{\beta\alpha}, \dots, P_{1\gamma}^{\beta\alpha}, \dots, P_{1c}^{\beta\alpha}$		
$P_\beta^{\beta\alpha}$	$\vdots$	$P_{\beta 1}^{\beta\alpha}, \dots, P_{\beta\gamma}^{\beta\alpha}, \dots, P_{\beta c}^{\beta\alpha}$		
$P_B^{\beta\alpha}$	$\vdots$	$P_{B1}^{\beta\alpha}, \dots, P_{B\gamma}^{\beta\alpha}, \dots, P_{Bc}^{\beta\alpha}$		

Совместные законы распределения всегда порождают частные законы распределений. И для состояний и выходов выполняются следующие соотношения:

$$1) P^{<b', a>}_b = \sum_{t_g \in U} P^{<b', a>}_{b, i}; \quad (1.45)$$

$$2) 0 \leq P^{<b', a>}_b \leq 1; \quad (1.46)$$

$$3) \sum_{x_b \in C} P^{<b', a>}_b = 1. \quad (1.47)$$

Эти соотношения задают вероятности появления состояния  $x_b$  при условии, что было состояние  $x_b^I$  и на вход пришло  $s_t$ .

Аналогично для выходов:

$$1) P_g^{<ba>} = \sum_{x_b \in C} P_{ba}^{<ba>}; \quad (1.48)$$

$$2) 0 \leq P_g^{<ba>} \leq 1; \quad (1.49)$$

$$3) \sum_{t_g \in U} P_g^{<ba>} = 1 \quad (1.50)$$

(вероятность появления выхода  $t_g$  при условии, что было состояние  $x_b^I$  и на вход пришло  $s_t$ ).

Если у вероятностного конечного автомата выход зависит от входа и предыдущего состояния и выполняется условие:

$$P^{<b', a>}_{b, i} = P^{<b', a>}_b * P^{<b', a>}_i, \quad (1.51)$$

$$s_t \in V, \quad t_g \in Y, \quad x_b^I, x_b \in X,$$

то этот автомат называется вероятностным автоматом Мили.

Это требование означает выполнение условия независимости распределений для нового состояния автомата и его выходного сигнала, т.е. означает, что новое состояние и новый выход определяются не как пара, а независимо друг от друга.

Подобно конечному автомату Мура возможен вероятностный конечный автомат, у которого выход не зависит от входа, а зависит только от текущего состояния. Тогда для  $P^{<b', a>}_{b, i}$  задается  $P^{<b', a>}_b$ , для каждого  $x_b$  задается  $P^{<b>}_i$ .

Вероятностный конечный автомат, у которого выход не зависит от входа, а зависит только от текущего состояния, называется вероятностным автоматом Мура. В этом случае для каждой пары  $(x_b^I, s_t)$  задается  $P_b^{<ba>}$ , для каждого  $x_b$  задается  $P_g^{<b>}$  и должно выполняться условие

$$P^{<b', a>}_{b, i} = P^{<b', a>}_b \times P^{<b>}_i. \quad (1.52)$$

Вероятностный конечный автомат называется автономным, если все случайные законы распределения  $P^{<b', a>}_{b, i}$  одинаковы при различных вариантах сочетания алфавитов входов, т.е. не зависят от входов:

$$P^{<b', a>}_{b, i} = P^{<b'>}_{b, i}, \quad "s_a \hat{I} V. \quad (1.53)$$

То есть, у автономного вероятностного автомата выходной алфавит состоит из единственной буквы и тогда количество совместных законов распределения не  $|X| \times |V|$ , а только  $|X|$  (сколько состояний, столько и таблиц).

Если входной сигнал вероятностного конечного автомата определяется детерминировано, т.е. не участвует в совместных законах распределения, которые в этом случае сводятся к законам распределения в количестве  $|X|$  ( $X$  – внутренний алфавит), то такой автомат называется  $Y$ -детерминированным вероятностным конечным автоматом.

Таким образом, для автономного  $Y$ -детерминированного вероятностного конечного автомата достаточно задать в вероятностном смысле только закон распределения  $x_b \hat{I} X$  для каждого  $x_b' \hat{I} X$ , т.е.  $P^{<b'>}_b, x_b, x_b' \hat{I} X$ .

Это означает, что вместо ранее рассмотренных наборов совместных законов распределения следует задать один совместный закон распределения для пары  $x_b, x_b'$  плюс начальное распределение. Такой вероятностный конечный автомат называется марковской цепью.

Рассмотрим случайный дискретный процесс или случайную последовательность  $x[t]$ ,  $t \in Z_0 = \{0, 1, 2, \dots\}$ , характеризуемую следующими свойствами:

1. В любой момент времени  $t \in Z_0$  состояние процесса  $x[t]$  может принять значение из конечного множества состояний  $X = \{x_b\}$ , т.е.  $x[t]$  равно одному из  $x_b$  в любой момент времени;

2. Вероятность перейти в состояние  $x_b$  в любой момент  $t$ , при условии, что в момент  $t - 1$  реализовалось состояние  $x_b^1$ , зависит только от состояния  $x_b^1$  и не зависит от предшествующих состояний.

Такая случайная последовательность называется конечной марковской цепью. Она дискретна как по времени, так и по состояниям. Переход конечной марковской цепи из состояния в момент времени  $t - 1$  в состояние в момент времени  $t$  полностью описывается переходной матрицей.

Матрица образована условными вероятностями, но, в отличие от рассмотренных ранее условных вероятностей, в этой отсутствуют входы и выходы, а значит каждое состояние зависит только от предыдущего состояния. Матрица имеет размер  $|X^*/X|$ . Ее строки занумерованы предшествующими состояниями, а столбцы – последующими. В каждой строке записаны вероятности перехода во все возможные последующие состояния из  $x_b^1$ , и значит, сумма всех элементов строки равна единице.

$$\sum_{x_b} P^{[t-1, t]}_{x_b, x_b} = 1. \quad (1.54)$$

Такая матрица называется стохастической (табл. 1.29).

Таблица 1.29

Переходная матрица неоднородной марковской цепи

$X^1$ \ X	$X_1, \dots, X_b, \dots, X_B$
$X_1^1$	$P\langle X_1^1 X_1 \rangle, \dots, P\langle X_1^1 X_b \rangle, \dots, P\langle X_1^1 X_B \rangle$
$\cdot$	.....
$\cdot$	.....
$\cdot$	.....
$X_b^1$	$P\langle X_b^1 X_1 \rangle, \dots, P\langle X_b^1 X_b \rangle, \dots, P\langle X_b^1 X_B \rangle$
$\cdot$	.....
$\cdot$	.....
$X_B^1$	$P\langle X_B^1 X_1 \rangle, \dots, P\langle X_B^1 X_b \rangle, \dots, P\langle X_B^1 X_B \rangle$

Если все переходные матрицы совпадают при любом  $t \in Z$ , т.е.

$$P^{[t-1, t]} \equiv P, \quad t_0 \in Z, \quad (1.55)$$

то такая марковская цепь называется однородной.

#### 1.4. Представление технологического процесса в виде вероятностного автомата

В реальных условиях многостадийного производства реализация технологического процесса не дает стопроцентную гарантию получения продукции строго заданного качества. Это объясняется влиянием значительного качества технологических величин, часть из которых может не регламентироваться стандартами. Кроме того, сложно выдержать слишком «узкие» допустимые режимы обработки при учёте погрешностей приборов.

То есть, если технология представлена в виде конечных автоматов, то использование любой технологической цепочки, образованной сочетанием строго определенных элементов алфавитов входов и состояний в большинстве случаев не обеспечивает гарантированное попадание в заданные элементы алфавитов выходных величин. Чтобы учесть данную ситуацию необходимо использовать для описания вероятностные автоматы.

Предварительное исследование технологического процесса позволяет проанализировать законы распределения факторов сквозной технологии и определить безусловные вероятности, с которыми каждое из состояний  $x_{b_k}$  ( $k = 1, \dots, K$ ), встречается в реальном производстве, и, следовательно, обычно используется. Данные вероятности можно анализировать с помощью гистограмм или диапазонной оценки для отдельных факторов и входных величин, а также и для их совместного появления.

Для отдельного элемента  $X_i$  можно создать табл. 1.32.

Таблица 1.32

Вероятности реализации элементов алфавита  $i$ -го фактора

$X_i$	$a_{ij}$	$\dots$	$a_{ij_i}$	$\dots$	$a_{ij_i}$
$P(a_i)$	$Pa_{ij}$	$\dots$	$Pa_{ij_i}$	$\dots$	$Pa_{ij_i}$

Как отмечалось выше, сочетания алфавитов технологических факторов формируют технологические траектории. Реализация каждой такой траектории позволяет получать определенный уровень свойств и оценивать затраты на её реализацию. Причём, для каждой траектории затраты можно рассчитать с вероятностью близкой к единице. Следовательно, это позволяет использовать детерминированные конечные автоматы для описания влияния технологии на затраты.

Поскольку реализация конкретной технологии не обеспечивает стро-  
гое получение конечных свойств определенного качества (т.е. не позволяет  
получить единственное сочетание алфавитов выходов), то необходимо для  
каждой технологической цепочки описать свой вероятностный автомат.

Описание функционирования вероятностного конечного автомата  
можно трактовать так, что на любом  $k$ -м переделе для каждой пары  
 $(x_{b(k-1)}, s_{a(k)})$  задается совместное условное распределение вероятностей  
новых состояний и выходов. Т. к. внутренний алфавит (алфавит технологи-  
ческих факторов) и выходной алфавит конечны, то имеем пару дискретных  
случайных величин, которая должна быть задана своим совместным законом  
распределения. Число таких распределений на каждом агрегате или переделе  
равно числу возможных пар  $(x_{b(k-1)}, s_{a(k)})$ . Закон совместного распреде-  
ления можно представить в виде табл. 1.33.

Таблица 1.33

Совместный закон распределения состояний и выходов на  $k$ -ом переделе

		$P_g$				
		$P_{I_k}^{<x_{b(k-1)}, s_{a(k)}>}$	...	$P_{g(k)}^{<x_{b(k-1)}, s_{a(k)}>}$	...	$P_{G_k}^{<x_{b(k-1)}, s_{a(k)}>}$
$P_b$	$y_k$	$t_{I_k}$	...	$t_{g_k}$	...	$t_{G_k}$
	$x_k$					
$P_{I_k}^{<x_{b(k-1)}, s_{a(k)}>}$	$X_{I_k}$	$P_{I(k)}^{<x_{b(k-1)}, s_{a(k)}>}$	...	$P_{I(g(k))}^{<x_{b(k-1)}, s_{a(k)}>}$	...	$P_{I(G_k)}^{<x_{b(k-1)}, s_{a(k)}>}$
$P_{b_k}^{<x_{b(k-1)}, s_{a(k)}>}$	$X_{b_k}$	$P_{b(k)}^{<x_{b(k-1)}, s_{a(k)}>}$	...	$P_{b(g(k))}^{<x_{b(k-1)}, s_{a(k)}>}$	...	$P_{b(G_k)}^{<x_{b(k-1)}, s_{a(k)}>}$
$P_{B_k}^{<x_{b(k-1)}, s_{a(k)}>}$	$X_{B_k}$	$P_{B(k)}^{<x_{b(k-1)}, s_{a(k)}>}$	...	$P_{B(g(k))}^{<x_{b(k-1)}, s_{a(k)}>}$	...	$P_{B(G_k)}^{<x_{b(k-1)}, s_{a(k)}>}$

При этом должны выполняться требования:

$$0 \leq P_{bg(k)}^{<x_{b(k-1)}, s_{a(k)}>} \leq 1 ; \quad (1.70)$$

$$\sum_{x_{b_k} \in X_k} \sum_{t_{g_k} \in Y_k} P_{bg(k)}^{<x_{b(k-1)}, s_{a(k)}>} = 1 . \quad (1.71)$$

Для состояний и выходов верны следующие соотношения:

$$1. P_{b(k)}^{<x_{b(k-1)}, s_{a(k)}>} = \sum_{t_{g_k} \in Y_k} P_{bg(k)}^{<x_{b(k-1)}, s_{a(k)}>} ; \quad (1.72)$$

$$2. 0 \leq P_{b(k)}^{<x_{b(k-1)}, s_{a(k)}>} \leq 1 ; \quad (1.73)$$

$$3. \sum_{x_{b_k} \in X_k} P_{b(k)}^{<x_{b(k-1)}, s_{a(k)}>} = 1 . \quad (1.74)$$

Т.е. эти вероятности задают вероятности появления состояний  $x_{b_k}$   
(технологических режимов) при условии, что был реализован режим  $x_{b_{k-1}}$   
и на вход передела или агрегата поступают входы  $s_{a_k}$ .

Аналогично для выходов:

$$1. 0 \leq P_{g(k)}^{<x_{b(k-1)}, s_{a(k)}>} \leq 1 ; \quad (1.75)$$

$$2. P_{g(k)}^{<x_{b(k-1)}, s_{a(k)}>} = \sum_{x_{b_k} \in X_k} P_{bg(k)}^{<x_{b(k-1)}, s_{a(k)}>} ; \quad (1.76)$$

$$3. \sum_{t_{g_k} \in Y_k} P_{g(k)}^{<x_{b(k-1)}, s_{a(k)}>} = 1 . \quad (1.77)$$

Если на каком-либо переделе или агрегате нет входных воздействий  
или они постоянны для всех технологических режимов, то в этом случае ис-  
пользуется автономный вероятностный конечный автомат (автомат, совме-  
стные законы распределения которого не зависят от входов). В этом случае:

$$P_{bg(k)}^{<x_{b(k-1)}, s_{a(k)}>} = P_{bg(k)}^{<x_{b(k-1)}>} \quad \text{для всех } s_{a_k} \in V_u . \quad (1.78)$$

Если выходной сигнал вероятностного конечного автомата определя-  
ется детерминировано (пример с затратами), т.е. не участвует в совместном  
законе распределения, то используется  $Y$ -детерминированный вероятностный  
конечный автомат.

$Y$ -детерминированный автомат можно использовать на переделах (аг-  
регатах), после которых уровень выходных свойств не оценивается.

Таким образом, для каждого варианта сочетаний алфавитов состоя-  
ний, полученных до предыдущего агрегата ( $x_{b(k)}$ ) (автономного  
 $Y$ -детерминированного вероятностного конечного автомата) можно задать  
вероятность  $P_{b_k}^{<b_{k-1}>}$ . Такое описание переходов (такой автомат) можно  
считать Марковской цепью.

При таких допущениях случайный дискретный процесс характеризу-  
ется следующими свойствами:

1. На любом переделе (агрегате) состояние процесса  $x_k$  может принять значение из конечного множества состояний  $X_k = \{x_{b(k)}\}$ , т.е.  $x_k$  равно одному из  $x_{b(k)}$  на любой стадии обработки продукции.

2. Вероятность перехода в состояние  $x_{b(k)}$  на  $k$ -м агрегате при условии, что на  $k-1$ -м агрегате реализовалось состояние  $x_{b(k-1)}$ , зависит только от состояния  $x_{b(k-1)}$  и не зависит от предыдущих переделов.

Такая последовательность называется конечной Марковской цепью. Она дискретна как по агрегатам, так и по состояниям (технологическим факторам).

Переход конечной Марковской цепи из состояния агрегата  $k-1$  в состояние агрегата  $k$  можно описать переходной матрицей:

$$P^{[k-1,k]} = \begin{bmatrix} P^{[k-1,k]} \\ x_{b(k-1)} \cdot x_{b(k)} \end{bmatrix} = \begin{bmatrix} P^{[k-1,k]} \left( \begin{array}{c} x_{b(k)} \\ \hline x_{b(k-1)} \end{array} \right) \end{bmatrix} \quad (1.79)$$

$$x_{b(k-1)} \in X_{k-1}, x_{b(k)} \in X_k$$

Строки матрицы занумерованы предыдущими состояниями, а столбцы – последующими (табл. 1.34).

Таблица 1.34

Переходная таблица из агрегата  $k-1$  в агрегат  $k$

$X^{k-1}$ \ $X^k$	$\xi_{1(k)}$ .....	$\xi_{\beta(k)}$ .....	$\xi_{B(k)}$
$\xi'_{1(k-1)}$	$P^{[k-1,k]}_{x_1^1 x_1}$ .....	$P^{[k-1,k]}_{x_1^1 x_b}$ .....	$P^{[k-1,k]}_{x_1^1 x_{B1}}$
$\xi'_{\beta(k-1)}$	$P^{[k-1,k]}_{\xi'_{\beta} b \xi_1}$ .....	$P^{[k-1,k]}_{x_b^1 x_b}$ .....	$P^{[k-1,k]}_{x_b^1 x_B}$
$\xi'_{B(k-1)}$	$P^{[k-1,k]}_{\xi'_{B} \xi_1}$ .....	$P^{[k-1,k]}_{x_B^1 x_b}$ .....	$P^{[k-1,k]}_{x_B^1 x_B}$

В реальных условиях данный подход можно использовать при определении затрат на производство. Уровень свойств в большинстве случаев является величиной случайной или, по крайней мере, не является строго детерминированной. Следовательно, можно для каждого режима задать вероятности получения того или иного уровня свойств, т.е. вероятности реализации различных сочетаний алфавитов выходных свойств.

В итоге информацию можно представить в виде табл. 1.35.

Каждая строка табл. 1.35 представляет собой отдельную реализацию технологии (технологическую цепочку). Зная распределение вероятностей выходных свойств можно выделить такие сочетания алфавитов входных величин и состояний (технологических параметров), которые обеспечивают максимальный уровень требуемых свойств. Для этого выбирается сочетание алфавитов выходных величин, отвечающих требованиям стандартов. Например, после реализации технологии обработки на  $k + 1$  агрегатах, сочетание

$$S^+ = c_{(k+1)j_1} \dots c_{(k+1)r_{k+1}} \dots c_{(k+1)R_{k+1}j_{R_{k+1}}} \cdot \quad (1.80)$$

Необходимо подобрать технологическую цепочку (набор сочетаний алфавитов входных величин и состояний), обеспечивающую максимальную вероятность попадания в  $S^+$ .

Для этой цели можно использовать критерий связи технологии и свойств, основанный на принципах теории информации [11,12].

Технологическая цепочка, имеющая максимальное значение критерия связи, может быть выбрана в качестве оптимальной технологии. Если максимальное значение критерия достигается при объединении нескольких технологических цепочек, описанных в табл. 1.35, то их комбинация может использоваться как оптимальное технологическое пространство.

## 2. Дискретные пространственные модели

В этом разделе приведены разнообразные пространственные структуры, образованные из клеток без памяти, взаимосвязанных некоторым регулярным образом. Их «динамика» трактуется как поведение относительно дискретного пространства  $S = Z_u^d$  или  $Z^d$  той или иной размерности  $d \geq 1$ .

Опишем элементарные клетки (рис. 2.1) [13]. Клетка без памяти, или линейная комбинационная цепь, не совершает эволюций во времени и пространстве или относительно какого бы то ни было аргумента. Она не является динамической системой в полном смысле этого слова, а представляет собой «чистый преобразователь вход-выход». Ее вектор входа  $U \in K^l$  и вектор выхода  $Y \in K^m$  связаны линейным преобразованием, определяемым матрицей над некоторым числовым полем  $K$ :

$$Y = L \cdot U \quad (2.1)$$

так что клетка без памяти – функциональная система.



Рис. 2.1. Клетки без памяти (а) и с памятью (б)

Следует подчеркнуть, что клетка без памяти «вне времени и пространства» является идеализацией: она может представлять достаточно богатую пространственно-временную структуру, но временными задержками в (2.1) пренебрегаем, а пространственная структура «зашивается» в матрицу  $L$ .

Клетка с памятью, или обычная линейная стационарная дискретно-временная динамическая система, описывается уравнениями состояний, входов и выходов, где  $x \in K^n$  – вектор состояния,  $y \in K^m$  – вектор выхода,  $u \in K^l$  – вектор входа:

$$\begin{aligned} x(t) &= \Phi x(t-1) + \Psi u(t), \quad x(0) = x_0, \\ y(t) &= Hx(t) + Gu(t), \quad t \in T = Z_0 = \{0, 1, 2, \dots\}. \end{aligned} \quad (2.2)$$

Она сложнее клетки без памяти в том отношении, что хотя ее пространственная структура «зашита» в матрицы уравнений (2.2), временными

задержками полностью не пренебрегаем – считая лишь, что все они кратны одной, принимаемой за единичную, и известным приемом сводим их к ней, а описание динамики – к уравнениям (2.2); при этом переменными состояниями считаем входные сигналы элементов задержки. В [14,15] подробно описано «защивание» пространственной структуры системы в матрицы уравнений (2.2) в предположении, что переменными состояниями считаются выходные сигналы элементов задержки; при этом уравнения (2.2) принимают альтернативную форму (второе уравнение – тоже):

$$x(t+1) = \Phi x(t) + \Psi u(t), \quad (2.3)$$

являющуюся традиционной, или

$$x(t) = \Phi x(t-1) + \Psi u(t-1); \quad (2.4)$$

наоборот, (2.2) можно записать

$$\begin{aligned} x(t+1) &= \Phi x(t) + \Psi u(t+1) = A(0)x(t) + B(1)u(t+1), \\ y(t) &= Hx(t) = C(0)x(t) \end{aligned} \quad (2.5)$$

(для дальнейшего удобства – в несколько иных обозначениях,  $G$  опущено).

## 2.1. Одномерные однонаправленные линейные итеративные цепи

Каждая из одномерных однонаправленных линейных итеративных цепей представляет собой бесконечную вправо цепочку клеток без памяти (рис. 2.2).

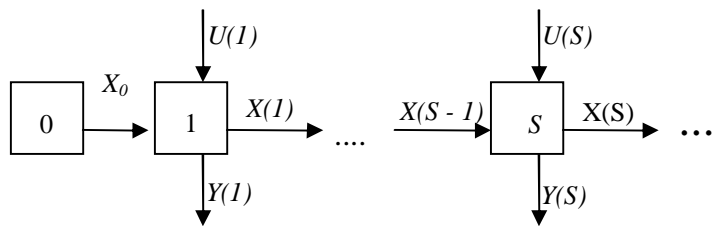


Рис. 2.2. Одномерная однонаправленная линейная итеративная цепь

Клеткам присваиваются номера  $s \in Z_0$ . Каждая клетка с номером  $s \in Z_1 = \{1, 2, \dots\}$  имеет внешние вход  $u(s) \in K^l$  и выход  $y(s) \in K^m$  и внутренние (по отношению к цепи как к целому) вход  $x(s-1) \in K^n$  (выход клетки с номером  $s-1$ ) и выход  $x(s) \in K^n$  (вход клетки с номером  $s+1$ ). Клетка с номером 0 (краевая) имеет внутренний выход  $x_0$ . Клетки идентичны (т.е. цепь пространственно однородна) и в соответствии с (2.1) каждая описывается уравнением

$$\begin{bmatrix} x(s) \\ y(s) \end{bmatrix} = L \begin{bmatrix} x(s-1) \\ u(s) \end{bmatrix}, \quad s \in Z_1. \quad (2.6)$$

Разбивая матрицу  $L$  на блоки, получаем уравнения состояний и выходов одномерной однонаправленной линейной итеративной цепи:

$$\begin{aligned} x(s) &= \Phi x(s-1) + \Psi u(s), \quad x(0) = x_0, \\ y(s) &= Hx(s-1) + Gu(s), \quad s \in Z_1. \end{aligned} \quad (2.7)$$

Эти уравнения формально идентичны уравнениям стационарной линейной дискретно-временной системы (2.2), с заменой в последних времени  $t$  на пространственную переменную  $s$ , или, в операторной форме, – с заменой оператора  $x$  временного сдвига на оператор  $z$  пространственного сдвига. Аргументная каузальность, априорная для временной системы, для пространственной определяется тем, что постулирован выбор одного из двух возможных направлений на пространственной оси (в следующем примере это уже не предполагается). Содержательные интерпретации рассмотренных цепей очевидны. Методы их исследования полностью заимствуются из хорошо разработанной теории дискретно-временных систем [16]. Одномерные однонаправленные цепи – класс моделей не менее богатый, чем класс временных систем, при том, что только лишь последние являются объектом изучения в классической теории управления. То же касается каждого из приводимых ниже примеров, а в совокупности этим подчеркивается широта класса дискретно-аргументных моделей.

## 2.2. Одномерные двунаправленные линейные итеративные цепи

Каждая из одномерных двунаправленных линейных итеративных цепей представляет собой (потенциально) бесконечную в обе стороны цепочку клеток без памяти (рис. 2.3).

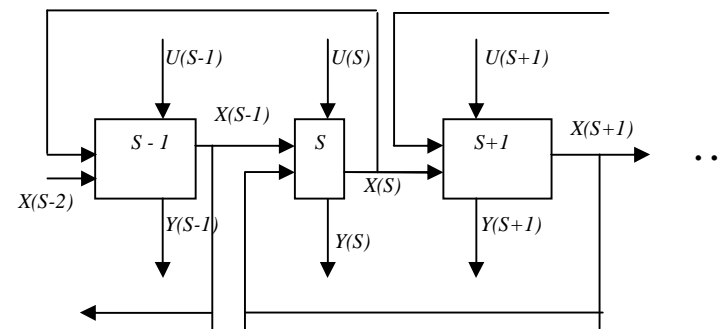


Рис. 2.3. Одномерная двунаправленная линейная итеративная цепь

1-го типа

Клеткам присвоены номера  $s \in Z = \{\dots, -1, 0, 1, \dots\}$ . В отличие от однонаправленной цепи, каждая клетка  $s$  имеет два внутренних входа  $x(s-1)$  и  $x(s+1)$ , тогда как ее внутренний выход  $x(s)$  подается как на клетку  $s+1$ , так и на клетку  $s-1$ . Это отличие особенно наглядно проявляется при сравнении шаблонов соседства однонаправленной и двунаправленной цепей (рис. 2.4).



Рис. 2.4. Шаблоны соседства Однонаправленной (а) и двунаправленной (б) цепей

Двунаправленность определяет аргументную некаузальность цепи, что соответствует классу «систем ближайшего соседа». Обычно такие цепи рассматриваются на конечном отрезке, например  $[-N, N]$ : крайние клетки  $-N, N$  имеют внутренние выходы  $x(-N)$  и  $x(N)$ . Клетки идентичны (цепь пространственно однородна), и в соответствии с (2.6) каждая описывается уравнением

$$\begin{bmatrix} x(s) \\ y(s) \end{bmatrix} = L \begin{bmatrix} x(s-1) \\ x(s+1) \\ u(s) \end{bmatrix}, \quad s \in Z. \quad (2.8)$$

Разбивая матрицу  $L$  на блоки, получаем уравнение цепи

$$\begin{aligned} x(s) &= \Phi_1 x(s-1) + \Phi_2 x(s+1) + \Psi u(s), \\ y(s) &= H_1 x(s-1) + H_2 x(s+1) + G u(s), \quad s \in Z. \end{aligned} \quad (2.9)$$

В дальнейшем основное внимание уделяем уравнению динамики. Описанную цепь относим к цепи I типа (рис. 2.5) (в отличие от цепи II типа, каждая клетка  $S$  которой имеет два внутренних выхода  $x_1(s)$  и  $x_2(s)$ , подаваемых на клетки  $s+1$  и  $s-1$  соответственно):

$$\begin{bmatrix} x_1(s) \\ x_2(s) \\ y(s) \end{bmatrix} = L \begin{bmatrix} x_1(s-1) \\ x_2(s+1) \\ u(s) \end{bmatrix}, \quad s \in Z \quad (2.10)$$

или

$$\begin{aligned} x_1(s) &= \Phi_{11} x_1(s-1) + \Phi_{12} x_2(s+1) + \Psi_1 u(s), \\ x_2(s) &= \Phi_{21} x_1(s-1) + \Phi_{22} x_2(s+1) + \Psi_2 u(s), \quad s \in Z, \end{aligned} \quad (2.11)$$

положим здесь  $x(s) = (x_1^T(s), x_2^T(s))^T$ . Тогда (2.11) можно записать в виде

$$x(s) = \begin{bmatrix} \Phi_{11} & 0 \\ \Phi_{21} & 0 \end{bmatrix} x(s-1) + \begin{bmatrix} 0 & \Phi_{12} \\ 0 & \Phi_{22} \end{bmatrix} x(s+1) + \begin{bmatrix} \Psi_1 \\ \Psi_2 \end{bmatrix} u(s), \quad (2.12)$$

так что цепь II типа получается как частный случай цепи I типа при специальном задании матриц  $F_1, F_2, Y$ .

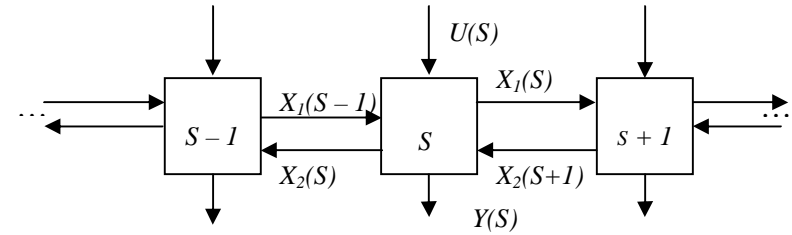


Рис. 2.5. Одномерная двунаправленная линейная итеративная цепь II типа

Подобное разбиение вектора состояния на составляющие (в данном случае – составляющая «вправо»  $x_1$  и составляющая «влево»  $x_2$ ) встретится нам и в дальнейшем. Операторная запись уравнений двунаправленной цепи использует операторы пространственного сдвига «назад»  $Z$  и «вперед»  $Z^{-1}$ , удовлетворяющих соотношению  $ZZ^{-1} = 1$ , так что модель аргументно-некаузальна, как уже отмечалось выше.

Содержательная интерпретация двунаправленных итеративных цепей, как и в однонаправленных, относится к таким областям как обработка сигналов, кодирование и декодирование информации, построение вычислительных систем [1]. Большое внимание моделям (2.9) уделяется в вычислительной математике, где они интерпретируются как широко известный класс разностных схем [17].

### 2.3. Двумерные однонаправленные линейные итеративные цепи

Каждая из двумерных однонаправленных линейных итеративных цепей представляет собой бесконечную вправо и вверх таблицу клеток без памяти (рис. 2.6).

Клетки занумерованы парами чисел  $(s_1, s_2) \in Z_0^2$ . В отличие от одномерной цепи, каждая клетка имеет два внутренних входа  $x(s_1-1, s_2)$  и  $x(s_1, s_2-1)$ , тогда как ее внутренний выход  $x(s_1, s_2)$  подается на две клетки  $(s_1+1, s_2)$  и  $(s_1, s_2+1)$  (соответствующий шаблон соседства представлен на рис. 2.7).

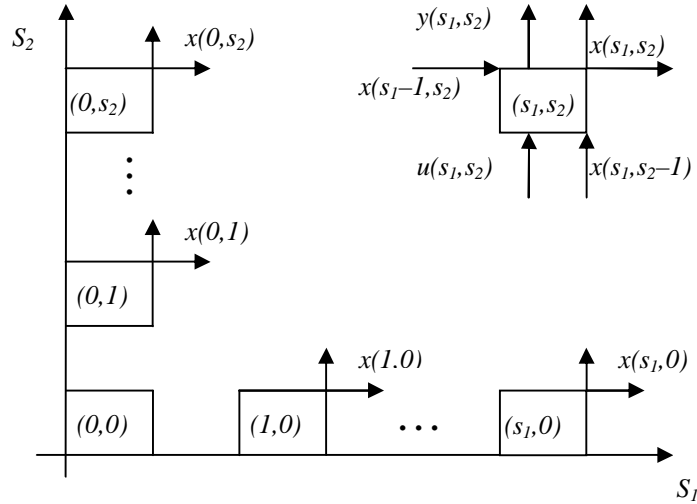


Рис. 2.6. Двумерная однонаправленная линейная итеративная цепь I типа

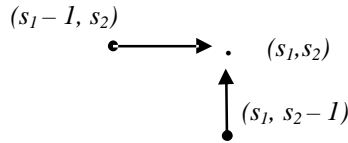


Рис. 2.7. Шаблон соседства двумерной однонаправленной цепи

Клетки  $(s_1, 0)$  и  $(0, s_2)$  (краевые) имеют внутренние выходы  $x_0(s_1, 0)$  и  $x_0(0, s_2)$ . Как и ранее, клетки идентичны (цепь пространственная и одно-родная) и каждая описывается уравнением

$$\begin{bmatrix} x(s_1, s_2) \\ y(s_1, s_2) \end{bmatrix} = L \begin{bmatrix} x(s_1 - 1, s_2) \\ x(s_1, s_2 - 1) \\ u(s_1, s_2) \end{bmatrix}, \quad s_1, s_2 \in Z_1^2 \quad (2.13)$$

так, что уравнение цепи:

$$\begin{aligned} x(s_1, s_2) &= \Phi_1 x(s_1 - 1, s_2) + \Phi_2 x(s_1, s_2 - 1) + \Psi_1 u(s_1, s_2), \\ y(s_1, s_2) &= H_1 x(s_1 - 1, s_2) + H_2 x(s_1, s_2 - 1) + \Psi_2 u(s_1, s_2), \end{aligned} \quad (2.14)$$

где  $(s_1, s_2) \in Z_1^2 = Z_1 \times Z_1$ .

Этой цепи I типа соответствует цепь II типа (рис. 2.8), в которой каждая клетка  $(s_1, s_2)$  имеет два внутренних выхода  $x_1(s_1, s_2) \in R^{n_1}$  и  $x_2(s_1, s_2) \in R^{n_2}$ , подаваемых на клетки  $(s_1 + 1, s_2)$  и  $(s_1, s_2 + 1)$  соответственно:

$$\begin{bmatrix} x_1(s_1, s_2) \\ x_2(s_1, s_2) \\ y(s_1, s_2) \end{bmatrix} = L \begin{bmatrix} x_1(s_1 - 1, s_2) \\ x_2(s_1, s_2 + 1) \\ u(s_1, s_2) \end{bmatrix}, \quad (s_1, s_2) \in Z_1^2, \quad (2.15)$$

так что  $x = (x_1^T, x_2^T)^T \in R^n$ ,  $n = n_1 + n_2$ ,

$$\begin{aligned} x_1(s_1, s_2) &= \Phi_{11} x_1(s_1 - 1, s_2) + \Phi_{12} x_2(s_1, s_2 - 1) + \Psi_1 u(s_1, s_2), \\ x_2(s_1, s_2) &= \Phi_{21} x_1(s_1 - 1, s_2) + \Phi_{22} x_2(s_1, s_2 - 1) + \Psi_2 u(s_1, s_2), \\ y(s_1, s_2) &= H_1 x_1(s_1 - 1, s_2) + H_2 x_2(s_1, s_2 - 1) + G u(s_1, s_2). \end{aligned} \quad (2.16)$$

(2.16) – частный случай (2.14) при специальном выборе матриц  $\Phi_1, \Phi_2, \Psi$ . Состояние двумерной цепи II типа оказывается разложенным на «горизонтальную»  $x_1$  и «вертикальную»  $x_2$  составляющие. Двумерная однонаправленная цепь аргументно-каузальна в соответствии с постулированным выбором одного из множества направлений на плоскости (именно направление «северо-восток»). Это соответствует классу «многомерных систем». Операторная запись уравнения цепи использует операторы горизонтального  $z_1$  и вертикального  $z_2$  пространственных сдвигов, причем  $z_1 z_2 = z_2 z_1$ .

Модель (2.14) является частным случаем модели

$$\begin{aligned} x(s_1, s_2) &= \Phi_1 x(s_1 - 1, s_2) + \Phi_2 x(s_1, s_2 - 1) + \Phi_3 x(s_1 - 1, s_2 - 1) + \\ &+ \Psi u(s_1, s_2) \end{aligned} \quad (2.17)$$

при  $\Phi_3 = 0$ . Покажем, что это уравнение (2.17) может быть, в свою очередь, приведено к виду (2.14). Для этого приведем (2.17) к (2.16). Положим

$$\begin{aligned} x_1(s_1, s_2) &= x(s_1, s_2), \\ x_2(s_1, s_2) &= \Phi_2 x(s_1, s_2) + \Phi_3 x(s_1 - 1, s_2). \end{aligned} \quad (2.18)$$

Тогда непосредственно из (2.17) получаем

$$x_1(s_1, s_2) = \Phi_1 x_1(s_1 - 1, s_2) + x_2(s_1, s_2 - 1) + \Psi u(s_1, s_2), \quad (2.19)$$

что соответствует первому уравнению (2.16) при  $\Phi_{11} = \Phi_1, \Phi_{12} = I, \Psi_1 = \Psi$ .

Подставим в выражение для  $x_2(s_1, s_2)$  полученное выражение для  $x_1(s_1, s_2)$ :

$$\begin{aligned} x_2(s_1, s_2) &= \Phi_3 x_1(s_1 - 1, s_2) + \Phi_2 [\Phi_1 x_1(s_1 - 1, s_2) + x_2(s_1, s_2 - 1) + \\ &+ \Psi u(s_1, s_2)] = (\Phi_3 + \Phi_2 \Phi_1) x_1(s_1 - 1, s_2) + \Phi_2 x_2(s_1, s_2 - 1) + \Phi_2 \Psi u(s_1, s_2). \end{aligned} \quad (2.20)$$



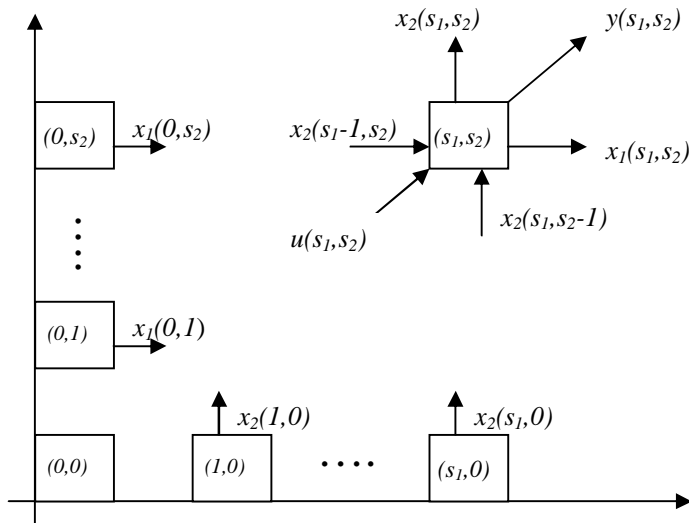


Рис. 2.8. Двумерная однонаправленная линейная итеративная цепь II типа

Это соответствует второму уравнению (2.16) при  $\Phi_{21} = \Phi_3 + \Phi_2 \Phi_1$ ,  $\Phi_{22} = \Phi_2$ ,  $\Psi_2 = \Phi_2 \Psi$ , так что (2.17) приведено к (2.16). Теперь положим  $\bar{x}(s_1, s_2) = [x_1^T(s_1, s_2) x_2^T(s_1, s_2)]^T$  и запишем:

$$\begin{aligned} \bar{x}(s_1, s_2) &= \begin{bmatrix} \Phi_1 & 0 \\ \Phi_3 + \Phi_2 \Phi_1 & 0 \end{bmatrix} \bar{x}(s_1 - 1, s_2) + \\ &+ \begin{bmatrix} 0 & I \\ 0 & \Phi_2 \end{bmatrix} \bar{x}(s_1, s_2 - 1) + \begin{bmatrix} \Psi \\ \Psi_2 \Psi \end{bmatrix} u(s_1, s_2) = \\ &= \bar{\Phi}_1 \bar{x}(s_1 - 1, s_2) + \bar{\Phi}_2 \bar{x}(s_1, s_2 - 1) + \bar{\Psi} u(s_1, s_2). \end{aligned} \quad (2.21)$$

Таким образом, все три формы (2.14), (2.16), (2.17) модели двумерной однонаправленной цепи могут быть преобразованы одна в другую.

В литературе широко изучались альтернативные варианты этих моделей. Они известны под следующими названиями [1]:

а) модель Хабиби-Аттази, Форназини-Маркезини

$$x(s_1 + 1, s_2 + 1) = \Phi_0 x(s_1, s_2) + \Phi_1 x(s_1 + 1, s_2) + \Phi_2 x(s_1, s_2 + 1) + \Psi u(s_1, s_2);$$

б) модель Россера

$$\begin{aligned} x_1(s_1 + 1, s_2) &= \Phi_{11} x_1(s_1, s_2) + \Phi_{12} x_2(s_1, s_2) + \Psi_1 u(s_1, s_2), \\ x_2(s_1, s_2 + 1) &= \Phi_{21} x_1(s_1, s_2) + \Phi_{22} x_2(s_1, s_2) + \Psi_2 u(s_1, s_2); \end{aligned}$$

в) обобщенная модель Форназини-Маркезини

$$x(s_1 + 1, s_2 + 1) = \Phi_1 x(s_1 + 1, s_2) + \Phi_2 x(s_1, s_2 + 1) + \Psi_1 u(s_1 + 1, s_2) + \Psi_2 u(s_1, s_2 + 1).$$

Они могут быть преобразованы одна в другую подобно тому, как это сделано выше для рассматриваемых здесь моделей. С другой стороны, последняя из указанных моделей может быть записана в виде (2.14) с несущественными отличиями:

$$x(s_1, s_2) = \Phi_1 x(s_1, s_2 - 1) + \Phi_2 x(s_1 - 1, s_2) + \Psi_1 u(s_1, s_2 - 1) + \Psi_2 u(s_1 - 1, s_2). \quad (2.22)$$

Полную двумерную модель запишем в виде:

$$\begin{aligned} x(s_1 + 1, s_2 + 1) &= A(0, 0) x(s_1, s_2) + A(1, 0) x(s_1 + 1, s_2) + \\ &+ A(0, 1) x(s_1, s_2 + 1) + B(0, 0) u(s_1, s_2) + B(1, 0) u(s_1 + 1, s_2) + \\ &+ B(0, 1) u(s_1, s_2 + 1) + B(1, 1) u(s_1 + 1, s_2 + 1). \end{aligned} \quad (2.23)$$

Эта модель включает все рассматривавшиеся до сих пор двумерные модели и характеризуется наличием члена  $B(1, 1)u(s_1 + 1, s_2 + 1)$ , отсутствовавшего в моделях Хабиби-Аттази, Форназини-Маркезини (и их модификациях), Россера и др. Этот член впервые был в модели мультиразмерных систем в [18, 19], что в отражено [1, 20, 21].

Основную двумерную модель (2.11) запишем в виде:

$$\begin{aligned} \bar{x}(s_1 + 1, s_2 + 1) &= \bar{A}(1, 0) \bar{x}(s_1 + 1, s_2) + \bar{A}(0, 1) \bar{x}(s_1, s_2 + 1) + \\ &+ \bar{B}(1, 1) u(s_1 + 1, s_2 + 1). \end{aligned} \quad (2.24)$$

Именно благодаря наличию члена  $\bar{B}(1, 1)u(s_1 + 1, s_2 + 1)$  полная двумерная модель (а значит, и все ее частные случаи) может быть приведена к основной.

## 2.4. Двумерные двунаправленные линейные итеративные цепи

Каждая из двумерных двунаправленных линейных итеративных цепей представляет собой (потенциально) бесконечную во все стороны таблицу клеток без памяти (рис. 2.9).

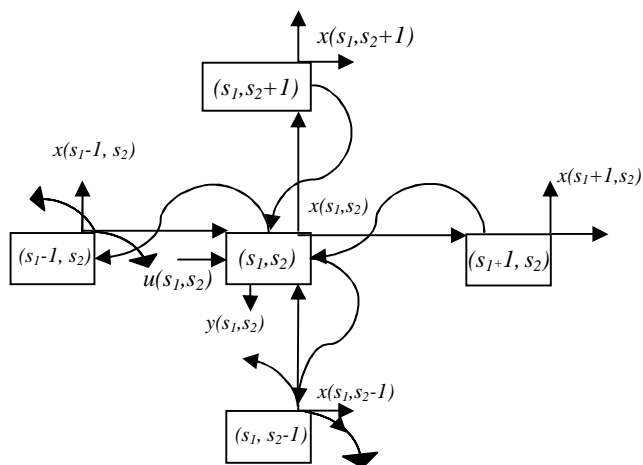


Рис. 2.9. Двумерная двунаправленная линейная итеративная цепь I типа

Клетки занумерованы парами чисел  $(s_1, s_2) \in Z^2 = Z \times Z$ . В отличие от однонаправленной цепи каждая клетка  $(s_1, s_2)$  имеет четыре входа от четырех ближайших соседей, а ее выход подается на четырех ближайших соседей. Этим и объясняется закрепившееся за такими системами название «системы ближайшего соседа». Особенно наглядно сказанное иллюстрируется шаблоном соседства (рис. 2.10).

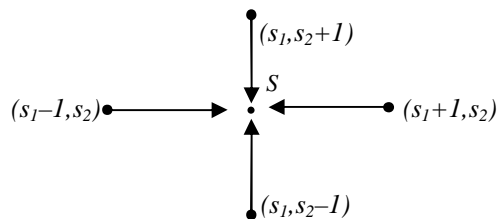


Рис. 2.10. Шаблон соседства двумерной двунаправленной цепи

Двунаправленность в горизонтальных и вертикальных направлениях определяет аргументную некаузальность цепи. Обычно такие цепи рассматриваются на конечном участке плоскости, например  $[-N_1, N_1] \times [-N_2, N_2]$ . Краевые клетки имеют только внутренние входы. Как и в предыдущих пунктах, клетки идентичны, каждая описывается уравнением

$$\begin{bmatrix} x(s_1, s_2) \\ y(s_1, s_2) \end{bmatrix} = L \begin{bmatrix} x(s_1 - 1, s_2) \\ x(s_1, s_2 - 1) \\ x(s_1 + 1, s_2) \\ x(s_1, s_2 + 1) \\ u(s_1, s_2) \end{bmatrix}, (s_1, s_2) \in Z_2. \quad (2.25)$$

После разбиения матрицы L на блоки, уравнения динамики такой цепи записываются в виде

$$x(s_1, s_2) = \Phi_1 x(s_1 - 1, s_2) + \Phi_2 x(s_1, s_2 - 1) + \Phi_3 x(s_1 + 1, s_2) + \Phi_4 x(s_1, s_2 + 1) + \Psi u(s_1, s_2), \quad (2.26)$$

$(s_1, s_2 \in Z^2)$ .

Описанной цепи I типа соответствует цепь второго типа (рис. 2.11), отличающаяся тем, что каждая клетка имеет не один, а четыре выхода, подаваемых каждый на одну из ближайших соседних клеток.

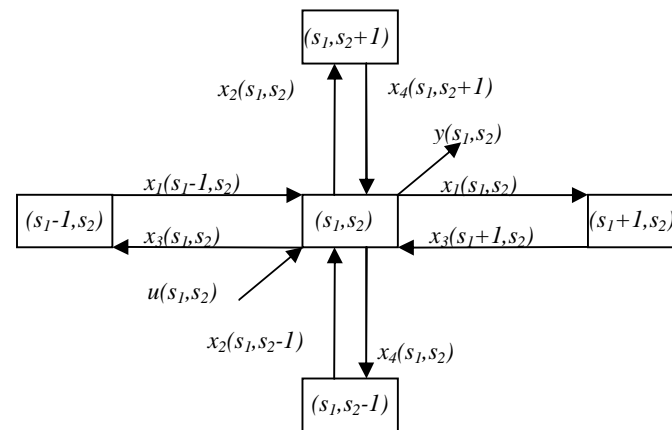


Рис. 2.11. Двумерная двунаправленная линейная итеративная цепь II типа

Схема такой цепи проще, а уравнения записываются в виде:

$$\begin{aligned}
 x_1(s_1, s_2) &= \Phi_{11}x_1(s_1 - 1, s_2) + \Phi_{12}x_2(s_1, s_2 - 1) + \Phi_{13}x_3(s_1 + 1, s_2) + \\
 &\quad + \Phi_{14}x_4(s_1, s_2 + 1) + \Psi_1u(s_1, s_2); \\
 x_2(s_1, s_2) &= \Phi_{21}x_1(s_1 - 1, s_2) + \Phi_{22}x_2(s_1, s_2 - 1) + \Phi_{23}x_3(s_1 + 1, s_2) + \\
 &\quad + \Phi_{24}x_4(s_1, s_2 + 1) + \Psi_2u(s_1, s_2); \\
 x_3(s_1, s_2) &= \Phi_{31}x_1(s_1 - 1, s_2) + \Phi_{32}x_2(s_1, s_2 - 1) + \Phi_{33}x_3(s_1 + 1, s_2) + \\
 &\quad + \Phi_{34}x_4(s_1, s_2 + 1) + \Psi_3u(s_1, s_2); \\
 x_4(s_1, s_2) &= \Phi_{41}x_1(s_1 - 1, s_2) + \Phi_{42}x_2(s_1, s_2 - 1) + \\
 &\quad + \Phi_{43}x_3(s_1 + 1, s_2) + \Phi_{44}x_4(s_1, s_2 + 1) + \Psi_4u(s_1, s_2).
 \end{aligned} \tag{2.27}$$

Можно непосредственно убедиться, что цепь II типа является частным случаем цепи I типа при специальном выборе матрицы  $\Phi_1, \dots, \Phi_4, \Psi$ . Операторная запись уравнений двумерных двунаправленных цепей используют операторы горизонтальных (влево и вправо) и вертикальных (вверх и вниз) сдвигов  $Z_1, Z_1^{-1}, Z_2, Z_2^{-1}$ , удовлетворяющих соотношениям  $Z_1Z_1^{-1} = I, Z_2Z_2^{-1} = I$ , что еще раз говорит об аргументной некаузальности этой модели.

Содержательную интерпретацию рассмотренных цепей можно найти в областях цифровой обработки изображений (случайных двумерных полей) и численного решения дифференциальных уравнений с частными производными. Именно в последней области хорошо известен шаблон «крест» (рис. 2.10). На этом основана «модель ближайшего соседа» [1, 22] для цифровой обработки изображений, обладающих определенными преимуществами перед другими. В [22] рассмотрен скалярный ( $n = l = 1$ ) и изотропный, ( $\Phi_1, \dots, \Phi_4 = 1$ ) вариант модели (2.26); подробно изучен скалярный вариант, для которого  $f_1 = f_2 = b, f_3 = f_4 = g$ , и отмечено, что изотропный случай  $b = g$  представляет особый интерес; указаны и интерпретации, связанные со случайными блужданиями в пространстве, а также со статической физикой.

## 2.5. Двумерные полуторанаправленные линейные итеративные цепи

Ограничимся двумя вариантами двумерных полуторанаправленных линейных итеративных цепей, для простоты представленных только шаблонами соседства (рис. 2.12).

Первая из них описывается уравнением (частный случай (2.26))

$$x_1(s_1, s_2) = \Phi_1x(s_1 - 1, s_2) + \Phi_2x(s_1, s_2 - 1) + \Phi_3x(s_1, s_2 + 1) + \Psi u(s_1, s_2), \tag{2.28}$$

в котором  $s_1 \in Z_0, s_2 \in Z$ . Так как здесь  $Z_2Z_2^{-1} = 1$ , то цепь аргументно некаузальна. Вторая цепь описывается уравнением

$$\begin{aligned}
 x_1(s_1, s_2) &= \Phi_1x(s_1 - 1, s_2) + \Phi_2x(s_1 - 1, s_2 - 1) + \\
 &\quad + \Phi_3x(s_1 - 1, s_2 + 1) + \Psi u(s_1, s_2),
 \end{aligned} \tag{2.29}$$

в котором снова  $s_1 \in Z_0, s_2 \in Z$ . Так как здесь нет соотношения типа  $d_{n=1}$ , то система аргументно-каузальна. Стоит отметить, что в данном случае имеется специальное соотношение  $Z_1Z_2 \cdot Z_1Z_2^{-1} = Z^2$ . Соотношения подобного типа не встречались в рассмотренных ниже примерах.

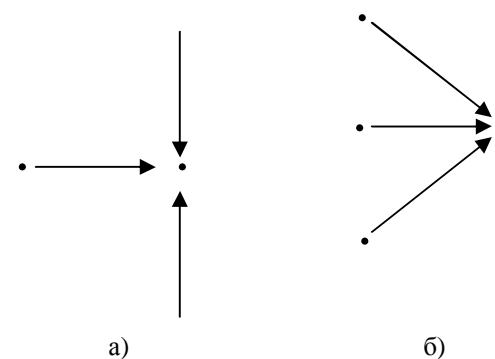


Рис. 2.12. Шаблоны соседства двумерных полуторанаправленных цепей

Двумерные полуторанаправленные цепи рассматривались в связи с задачами обработки изображений под названием «полукаузальных» процессоров. Как показано выше, по классификации, принятой в данной работе, они могут быть как аргументно-каузальными, так и аргументно-некаузальными. Шаблоны, изображенные на рис. 2.12, как и встречавшиеся ранее (а также рассмотренные в следующем пункте), хорошо известны в теории разностных схем.

## 2.6. Другие двумерные цепи

Не выписывая здесь соответствующих уравнений, можно провести сравнительный анализ шаблонов соседства некоторых цепей (рис. 2.13).

На рис. 2.13а показан шаблон цепи (2.22). Такое описание двумерной однонаправленной цепи избыточно, так как сводится к (2.19), т.е. шаблону, показанному на рис. 2.7, хотя последний и является частным случаем шаблона на рис. 2.13а. Другие частные случаи также показаны на рис. 2.13 в, с. Соот-

вестующие им цепи как частные случаи могут быть приведены к (2.19). Все эти цепи аргументно-каузальны. Альтернативные варианты шаблонов показаны на рис. 2.13d, f (каждому соответствуют и свои частные случаи).

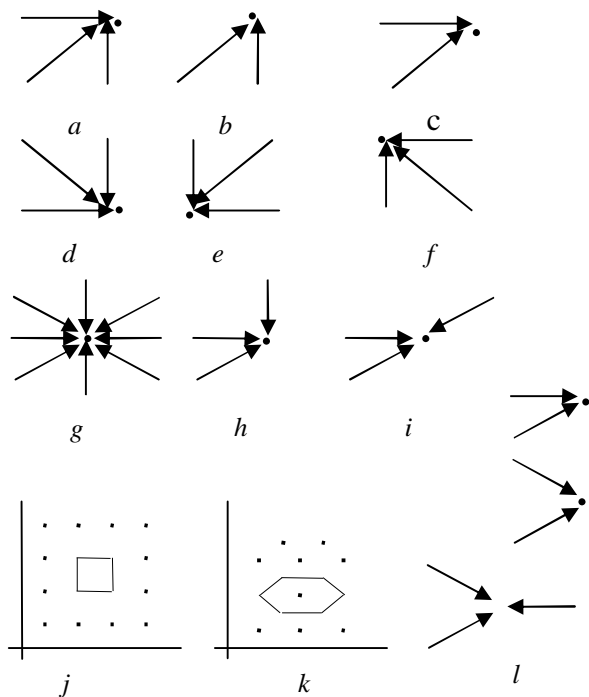


Рис. 2.13. Шаблоны соседства различных цепей

Все соответствующие цепи аргументно-каузальны, однонаправлены, меняется лишь направление, соответственно «юго-восток», «юго-запад», «северо-запад» (цепи (2.22) отвечает направление «северо-восток»). Цепь с шаблоном, показанным на рис. 2.13, может быть приведена к (2.22) подобно тому, как (2.22) приведена к (2.19). Она аргументно-некаузальна, хотя среди ее частных случаев (а она, по существу, охватывает все рассмотренные выше цепи) есть как каузальные, так и некаузальные. Иногда имеет значение то, что она может быть представлена как некоторая комбинация четырех каузальных цепей. На рис. 2.13 h показаны шаблоны еще двух цепей, первая из них каузальна, а вторая – нет. Из проведенного краткого сравнения можно сделать следующие выводы: цепь некаузальна всякий раз, когда на шаблоне встречается два взаимнопротивоположных направления: цепь избыточна (подобно (2.22)), когда среди направлений есть линейно-зависимые. Иными словами, избыточность связана с наличием специальных тождеств

между операторами. Результаты подобного анализа допускают дальнейшее развитие.

Все рассмотренные выше двумерные цепи были связаны с ортогональной (прямоугольной, в частности – квадратной) дискретизацией плоскости. В [22] обращено внимание на возможность использования других способов дискретизации, в частности – гексагональной, и предложены соответствующие «гексагональные модели». Ортогональная и гексагональная дискретизация сравниваются на рис. 2.13j, k. Некоторые шаблоны гексагональных цепей показаны на рис. 2.13. К ним также применима классификация на аргументно-каузальные и аргументно-некаузальные.

## 2.7. Трехмерные однонаправленные линейные итеративные цепи

Полную трехмерную модель запишем в виде

$$\begin{aligned}
 x(s_1 + 1, s_2 + 1, s_3 + 1) = & A(0, 0, 0)x(s_1, s_2, s_3) + \\
 & + A(1, 0, 0)x(s_1 + 1, s_2, s_3) + A(0, 1, 0)x(s_1, s_2 + 1, s_3) + \\
 & + A(0, 0, 1)x(s_1, s_2, s_3 + 1) + A(1, 1, 0)x(s_1 + 1, s_2 + 1, s_3) + \\
 & + A(1, 0, 1)x(s_1 + 1, s_2, s_3 + 1) + A(0, 1, 1)x(s_1, s_2 + 1, s_3 + 1) + \\
 & + B(0, 0, 0)u(s_1, s_2, s_3) + B(1, 0, 0)u(s_1 + 1, s_2, s_3) + \\
 & + B(0, 1, 0)u(s_1, s_2 + 1, s_3) + B(0, 0, 1)u(s_1, s_2, s_3 + 1) + \\
 & + B(1, 1, 0)u(s_1 + 1, s_2 + 1, s_3) + B(1, 0, 1)u(s_1 + 1, s_2, s_3 + 1) + \\
 & + B(0, 1, 1)u(s_1, s_2 + 1, s_3 + 1) + B(1, 1, 1)u(s_1 + 1, s_2 + 1, s_3 + 1).
 \end{aligned} \tag{2.30}$$

Основную трехмерную модель запишем в виде

$$\begin{aligned}
 \bar{x}(s_1 + 1, s_2 + 1, s_3 + 1) = & \bar{A}(1, 1, 0)\bar{x}(s_1 + 1, s_2 + 1, s_3) + \bar{A}(1, 0, 1)\bar{x}(s_1 + 1, s_2, s_3 + 1) + \\
 & + \bar{A}(0, 1, 1)\bar{x}(s_1, s_2 + 1, s_3 + 1) + \\
 & + \bar{A}(0, 1, 1)x(s_1, s_2 + 1, s_3 + 1) + \bar{B}(1, 1, 1)u(s_1 + 1, s_2 + 1, s_3 + 1).
 \end{aligned} \tag{2.31}$$

Замечания, сделанные ранее о двумерных моделях, справедливы и здесь. Впервые предложенная в [18,19] в общем случае основная трехмерная модель описывалась уравнением

$$\begin{aligned}
 x(s_1, s_2, s_3) = & \Phi_1 x(s_1 - 1, s_2, s_3) + \Phi_2 x(s_1, s_2 - 1, s_3) + \\
 & + \Phi_3 x(s_1, s_2, s_3 - 1) + \Psi u(s_1, s_2, s_3).
 \end{aligned} \tag{2.32}$$

## 2.8. Многомерные однонаправленные линейные итеративные цепи

Пусть  $s \in Z^d$ ,  $\underline{s} = (s_1, \dots, s_d)$ ,  $s_j \in Z$ ,  $j = 1, \dots, d$ ;  $b \in B = \{0, 1\}^d$  – множество всех  $d$ -мерных двоичных векторов с координатами 0 или 1:  $\underline{b}$  –

число единиц в векторе  $\underline{b}$  (вес Хемминга),  $0 \leq |\underline{b}| \leq d$ ,  $\underline{1} = (1, \dots, 1)$ ,  $|\underline{1}| = d$ .

Неявную полную  $d$ -мерную модель определим уравнением

$$\sum_{\underline{b} \in B} \{A(\underline{b})x(\underline{s} + \underline{b}) + B(\underline{b})u(\underline{s} + \underline{b})\} = 0, \quad (2.33)$$

а явную полную  $d$ -мерную модель – уравнением

$$x(\underline{s} + 1) = \sum_{\underline{b} \in B \setminus \{1\}} \{A(\underline{b})x(\underline{s} + \underline{b}) + B(\underline{b})u(\underline{s} + \underline{b}) + B(1)u(\underline{s} + 1)\}. \quad (2.34)$$

Эта модель включает все рассматривавшиеся до сих пор модели мультиразмерностных систем и характеризуется наличием члена  $B(1)u(\underline{s} + 1)$ , отсутствовавшего в ранее рассмотренных общих моделях. Этот член впервые был введен в модели мультиразмерностных систем [18,19], где рассматривалась (и была развита в [20] до основной модели дискретно-аргументных систем) основная  $d$ -мерная модель, записанная в виде

$$x(\underline{s}) = \sum_{j=1}^d \Phi_j x(s_1, \dots, s_{j-1}, s_j - 1, s_{j+1}, \dots, s_d) + \Psi u(\underline{s}). \quad (2.35)$$

В принятых здесь обозначениях основная  $d$ -мерная модель запишется в виде:

$$x(\underline{s} + 1) = \sum_{|\underline{b}|=d-1} A(\underline{b})x(\underline{s} + \underline{b}) + B(1)u(\underline{s} + 1). \quad (2.36)$$

Основные двумерная и трехмерная модели (2.24) и (2.31) являются частными случаями (2.36) соответственно при  $d = 2$  и  $d = 3$ . Приведение полной  $d$ -мерной модели к основной можно выполнить, развивая методику, подробно изложенную выше для случаев  $d = 2$  и  $d = 3$ . Тем самым и все рассматривавшиеся в литературе частные случаи полной модели могут быть приведены к основной, так что основная модель является в этом смысле наиболее общей, универсальной моделью мультиразмерностных систем.

Ключевым для возможности таких преобразований представляется следующее обстоятельство: правая часть основной модели содержит только члены «максимального веса» относительно состояний  $|\underline{b}| = d - 1$  и относительно входа  $|\underline{1}| = d$ ; все члены полной модели меньшего веса  $0 \leq |\underline{b}| \leq d$  при указанных преобразованиях «впитываются» членами «максимального веса». В этом смысле основная модель обладает «естественной максимальной компактностью» среди всех  $d$ -мерных моделей, тогда как полная модель обладает «естественной максимальной полнотой» – при соблюдении принципа причинности или каузальности для мультиразмерностных и вообще дискретно-временных моделей.

## 2.9. Использование итеративных цепей для описания технологического процесса

Выделенная из табл. 1.35 технологическая цепочка, обеспечивающая оптимальный уровень выходных свойств и описанная с помощью конечного и вероятностного конечного автоматов, может быть использована для управления технологическим процессом с использованием итеративной цепи.

Получив оптимальные диапазоны, состоящие из нескольких сочетаний алфавитов состояний (технологических величин), можно оценить вид образованного пространства (рис. 2.14).

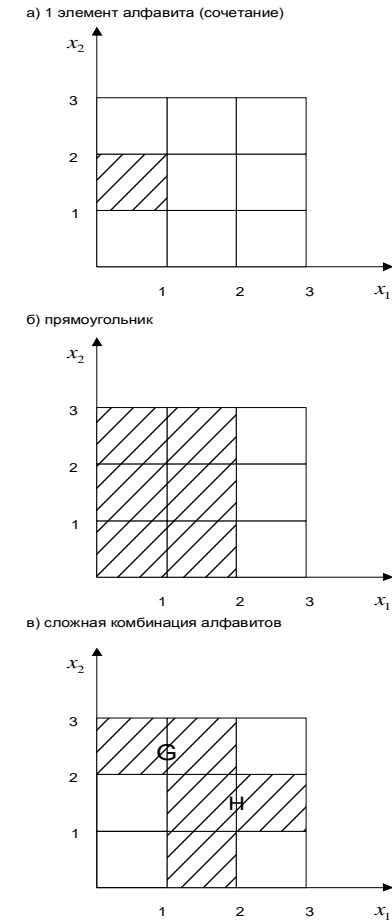


Рис. 2.14. Виды сочетаний алфавитов состояний, образующих оптимальное технологическое пространство

В случае, если оптимальное подпространство состоит из одного сочетания алфавитов состояний (рис. 2.14а), можно сформировать новый алфавит каждого фактора, равный трем (например,  $A, B, V$ . В этом случае  $B_i$  – будет соответствовать оптимальному диапазону, который обеспечивает максимальный уровень свойств).

При объединении нескольких сочетаний алфавитов в прямоугольный параллелепипед (рис. 2.14б) новый алфавит представляет собой объединение (дизъюнкцию) исходных алфавитов. Например, для рис. 2.14б выходы можно описать для факторов  $x_i$ :  $B_1 = \{1 \cup 2\}$ ,  $B_1 = \{3\}$ ,  $A_1 = \{\emptyset\}$ , для фактора  $x_2$ :  $A_2 = \{\emptyset\}$ ,  $B_2 = \{1 \cup 2 \cup 3\}$ ,  $B_2 = \{\emptyset\}$ . В итоге оптимальное сочетание выходов (требуемое продолжение) будет  $B_1, B_2$ .

Для варианта, при котором формируется из сочетаний алфавитов пространство неправильной формы (рис. 2.14 в), необходимо указать объединение отдельных прямоугольных параллелепипедов, аналогичных варианту б. В этом случае формируется пространство как объединение параллелепипедов, каждый из которых идентифицируется отдельной буквой ( $G, H, F...$ ). В рассмотренном примере (рис. 2.14в) можно сформировать:  $G = \{1_1, 2_1, 3_2\}$ ,  $H = \{2_1, 3_1, 2_2\}$  (индекс – номер фактора). В итоге пространство можно представить как объединение:  $G \cup H \cup \{2_1, 1_2\}$ . Полученную технологию можно представить в виде итеративной цепи (рис. 2.15).

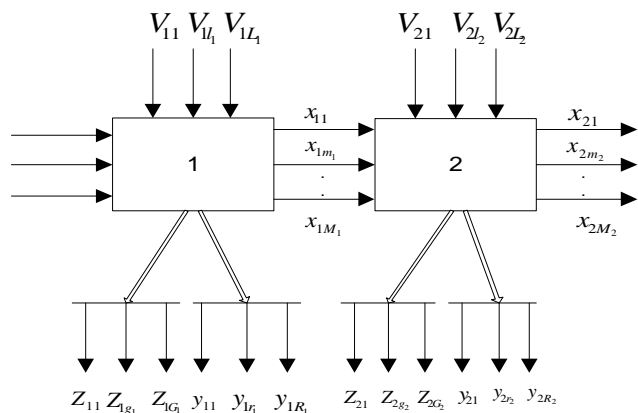


Рис. 2.15. Представление технологии в виде итеративной цепи

Описать цепь можно с помощью уравнений:

$$\begin{aligned} x(k) &= \Phi x(k-1) + yv(k), \\ y(k) &= Hx(k-1) + Gv(k). \end{aligned} \quad (2.37)$$

Данные уравнения применимы, если используется линейная итеративная цепь. В более общем случае (при учете взаимного влияния величин) необходимо перейти к нелинейным итеративным цепям.

Т.к. для каждой клетки цепи (технологического агрегата) используется разное количество входов, состояний и выходов, а функции переходов и выходов не идентичны, то формируется пространственно-неоднородная цепь. Для каждой клетки необходимо формировать свои законы изменения состояний и выходов, описываемые в виде уравнений или таблиц.

В итоге можно сформировать таблицы переходов и выходов для каждой клетки (табл. 1.36).

Таблица 1.36

Таблица переходов и выходов клетки итеративной цепи

$V_{k1} \dots V_{kL_k}$	$x_{(k-1)1} \dots x_{(k-1)M_{k-1}}$	$x_{k1} \dots x_{kM_k}$	$y_{k1} \dots y_{kR_k}$

Таким образом, технологический прогресс описан в виде клеток, каждая из которых соответствует отдельному технологическому агрегату или переделу. Однако, внутри передела технологические параметры реализуются одновременно или последовательно. Чтобы удобно было описывать реализацию процесса и осуществить его оперативное управление, необходимо для каждой клетки цепи, соответствующей одному агрегату или переделу, сформировать внутреннюю итеративную цепь, учитывающую одновременную или последовательную реализацию технологических величин.

Количество внутренних входов и выходов каждой клетки полученной цепи может быть равным одному или  $m_{i_k}$ . Варианты клеток представлены на рис. 2.17.

Причем

$$M_{i_{k-1}} = \sum_{j=1}^{i_{k-1}} m_j, \quad (2.38)$$

где  $m_j$  – количество внутренних выходов  $j$ -й клетки.

Каждый внутренний вход и выход клетки имеет 3 индекса (номер передела (номер клетки внешней цепи), номер клетки внутренней цепи, номер фактора, соответствующий его порядковой реализации). На вход в первую клетку внутренней цепи поступают данные о факторах, реализованных на предыдущем переделе (клетке внешней цепи).

Клетки внутренней цепи неоднородны. Для каждой клетки необходимо сформировать свою таблицу переходов, которая имеет вид табл. 1.37.

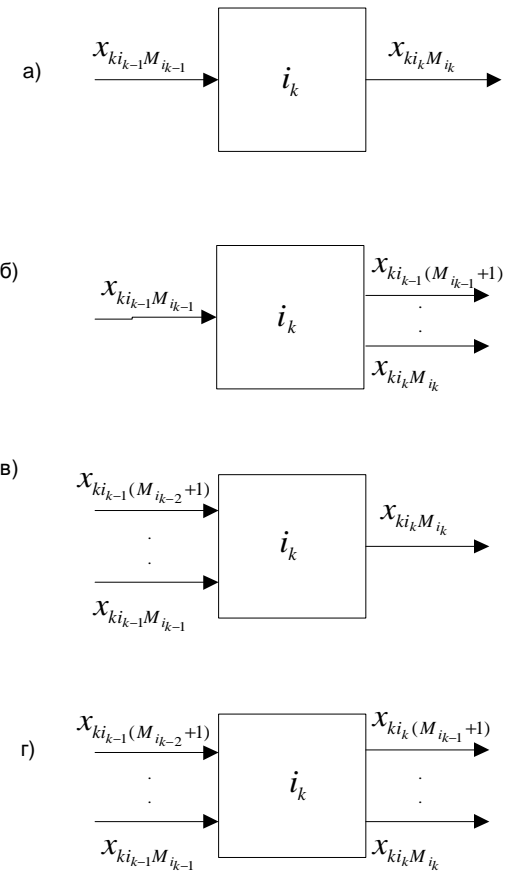


Рис. 2.17. Варианты внутренних входов и выходов клеток внутренней итеративной цепи

Таблица 1.37

Таблица переходов клетки внутренней итеративной цепи

$x_{ki_{k-1}(M_{ik-2}+1)}, \dots, x_{ki_{k-1}M_{ik-1}}$	$x_{ki_k(M_{ik-1}+1)}, \dots, x_{ki_kM_{ik}}$

Входами таблицы могут служить элементы алфавитов факторов, реализованных в предыдущей клетке. При этом рассматриваются сочетания факторов, выделенные как оптимальные (см. рис. 2.14). Для остальных сочетаний факторов осуществляется выход в цепь верхнего уровня, где решается

вопрос о выборе нового, рационального сочетания алфавитов факторов, реализация которых должна быть осуществлена на последующих переделах. Для этого вновь рассматривается работа вероятностного конечного автомата и выбирается продолжение, обеспечивающие максимальную вероятность получения требуемых свойств для данного вида продукции. В случае невозможности получения требуемого уровня свойств рассматривается вариант перевода производимой продукции в другой вид продукции или в брак до завершения технологического процесса.

Таким образом, в таблице переходов внутренней цепи (табл. 1.37) выходами могут служить элементы алфавитов факторов, описываемых в данной клетке (см. рис. 2.14а). (Для приведенного варианта – это сочетания:  $x^*_1 = 1$ ;  $x^*_2 = 2$ ).

При выборе оптимального продолжения, состоящего из комбинации алфавитов (рис. 2.14б), в качестве выходов указывается новый алфавит, представляющий собой объединение (дизъюнкцию) алфавитов. (Для примера:  $x^*_1 = \{1 \cup 2\}$ ,  $x^*_2 = \{1 \cup 2 \cup 3\}$  или  $\{B_1 \cup B_2\}$ ).

Для варианта 2.14в необходимо указать объединение отдельных прямоугольников, аналогичных варианту 2.14б. (Для приведенного варианта, как отмечалось выше,  $G \cup H \cup \{2_1, 1_2\}$ ).

### 3. Дискретные пространственно-временные модели

В этом разделе продолжаем пользоваться клеточной интерпретацией дискретно-аргументных систем. Теперь в качестве элементарных клеток используем клетки с памятью. Рассмотренные здесь системы в обзоре [1] именуется «линейными последовательно-клеточными машинами».

#### 3.1. Одномерные однонаправленные линейные итеративные сети

Одномерные однонаправленные линейные итеративные цепи отличаются от соответствующих цепей (подраздел 2.1) тем, что представляют собой бесконечную вправо цепочку клеток с памятью (с этим отличием сохраняется схема рис. 3.1, представленная на рис. 2.14).

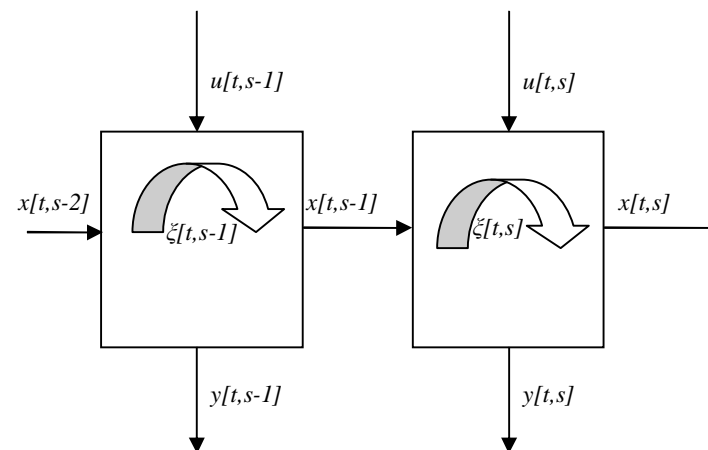


Рис. 3.1. Одномерная однонаправленная линейная итеративная сеть

Сигналы в такой сети зависят уже не только от номера клеток  $s \in Z_0$ , но и от времени  $t \in Z_0$ . Обозначим состояние клетки  $s$  в момент  $t$  через  $x(t, s)$  и используем ее описание (2.2). Кроме внешних входа  $u(t, s)$  и выхода  $y(t, s)$ , она имеет в соответствии с подразделом 2.1 внутренние вход  $x(t, s-1)$  и выход  $x(t, s)$ . Пусть клетки идентичны, так что сеть аргументно-инвариантна: время-стационарна (как (2.2)) и пространственно однородна (как (2.6)). Сочетая (2.2) и (2.7), запишем вместо (2.6) уравнения



$$\begin{aligned} x(t, s) &= \Phi x(t-1, s) + \Psi \begin{bmatrix} x(t, s-1) \\ u(t, s) \end{bmatrix}, \\ \begin{bmatrix} x(t, s) \\ y(t, s) \end{bmatrix} &= H x(t-1, s) + G \begin{bmatrix} x(t, s-1) \\ u(t, s) \end{bmatrix}, \quad t, s \in Z_0. \end{aligned} \quad (3.1)$$

После разбиения матриц  $Y, H, G$  на блоки получаем уравнения одномерной однонаправленной линейной итеративной сети:

$$\begin{aligned} x(t, s) &= \Phi x(t-1, s) + \Psi_1 x(t, s-1) + \Psi_2 u(t, s), \\ x(t, s) &= H_1 x(t-1, s) + G_{11} x(t, s-1) + G_{12} u(t, s), \\ y(t, s) &= H_2 x(t-1, s) + G_{21} x(t, s-1) + G_{22} u(t, s). \end{aligned} \quad (3.2)$$

Эти уравнения решаются при начальных условиях  $x(0, s) = x_0(s), s \in Z_0$  и краевых условиях  $x(t, 0) = x_0(t), t \in Z_0$ . Таким образом, описание одномерной однонаправленной сети формально вполне идентично описанию двумерной двунаправленной цепи II типа (2.16), но сеть эволюционирует не относительно двумерного пространственного аргумента (как цепь), а относительно одномерного пространственного и временного аргументов. Вектор состояния сети составлен из  $x(t, s)$ - и  $y(t, s)$ -векторов состояний обеих эволюций. Операторная запись уравнений сети использует операторы временного  $x$  и пространственного  $z$  сдвигов, причем сеть аргументно-каузальна. Она имеет следующую особенность: состояние каждой клетки в момент  $t$  зависит не только от ее состояния в момент  $t-1$ , но и от состояния соседней клетки в тот же момент  $t$ . Шаблон пространственно-временного соседства сети соответствует показанному на рис. 2.7.

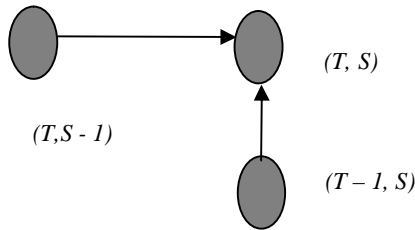


Рис. 3.2. Шаблон соседства одномерной однонаправленной сети

Содержательная интерпретация рассмотренных сетей связана с построением вычислительных систем. Как и в примерах предыдущего раздела, следует указать на связь с разностными схемами.

Рассмотренные уравнения описывают линейные итеративные сети. Они являются частным случаем уравнения для нелинейных сетей. При этом коэффициенты при членах, описывающих совместное влияние состояний и выходов, принимаются равными нулю.

Если сеть описывается с помощью двоичного алфавита, то для представления нелинейных одномерных однонаправленных итеративных сетей можно использовать многочлены Жегалкина.

Например, состояние клетки  $s$  в момент времени  $t$  определяется по формуле:

$$\begin{aligned} x[t, s] &= \Phi_0 \oplus \Phi_1 \otimes x[t-1, s] \oplus \Phi_2 \otimes x[t, s-1] \oplus \Phi_3 \otimes u[t, s] \oplus \Phi_{12} \otimes \\ &\otimes x[t-1, s] \otimes x[t, s-1] \oplus \Phi_{13} \otimes x[t-1, s] \otimes u[t, s] \oplus \Phi_{23} \otimes x[t, s-1] \otimes \\ &\otimes u[t, s] \oplus \Phi_{23} \otimes x[t-1, s] \otimes x[t, s-1] \otimes u[t, s]. \end{aligned} \quad (3.3)$$

В случаях, если алфавиты входов, состояний и выходов не являются двоичными, процесс получения уравнений состояний и выходов для нелинейной итеративной сети становится более трудоемким. Одним из вариантов описания функционирования таких сетей являются представление функций переходов и выходов в табличной форме. Занеся в таблицу полную комбинацию возможных исходов, можно полностью описать работу одномерной однонаправленной нелинейной итеративной сети.

### Работа одномерной однонаправленной сети

В каждый момент времени  $t = 0, 1, 2 \dots$  дискретного времени каждая клетка  $s$  сети находится в определенном состоянии  $\zeta[t, s]$  и способна воспринимать на внешнем входном канале сигнал  $u[t, s]$ , а на внутреннем входном канале сигнал  $x[t, s-1]$ . На эти сигналы клетки реагируют переходом в  $[t+1]$  такте в новое состояние  $\zeta[t+1, s]$  и выдачей некоторого внешнего выходного сигнала  $y[t+1, s]$  и внутреннего выходного сигнала  $x[t+1, s]$ , который является внутренним входом для клетки  $s+1$ .

В общем случае одномерная однонаправленная сеть имеет несколько внутренних и внешних входов, состояний, внешних и внутренних выходов.

Вектор, составленный из скалярных внешних входов:

$$u[t, s] = (u_1[t, s], \dots, u_m[t, s]), \quad \text{где } m - \text{число внешних входов клетки } s+1;$$

$$x[t, s] = (x_1[t, s], \dots, x_n[t, s]), \quad \text{где } n - \text{число внутренних входов клетки } s+1;$$

$$x[t, s] = (x_1[t, s], \dots, x_l[t, s]), \quad \text{где } l - \text{число внутренних состояний клетки } s;$$

$$y[t, s] = (y_1[t, s], \dots, y_k[t, s]), \quad \text{где } k - \text{число внешних выходов клетки } s.$$

Алфавиты входов, состояний и выходов могут иметь различную значность.

$a_{j_i}$  – составляющие алфавитов внешних входов;

$J_i = 1, \dots, J_i, J_i$  – значность алфавита  $i$ -го внешнего входа;

$i = 1, \dots, m$  – номер входов;

$b_{j_r}$  – составляющие алфавитов внутренних входов и выходов;

$J_r = 1, \dots, J_r, J_r$  – значность алфавита  $r$ -го внутреннего входа или выхода;

$r = 1, \dots, n$  – номера внутренних входов или выходов.

$c_{p_j}$  – составляющие алфавитов состояний.

$j_p = 1, \dots, J_p, J_p$  – значность алфавита  $p$ -го состояния.

$p = 1, \dots, l$  – номера состояний;

$d_{q_i}$  – составляющие алфавитов внешних выходов;

$j_q = 1, \dots, J_q, J_q$  – значность алфавита  $q$ -го внешнего входа;

$q = 1, \dots, k$  – номера внешних входов.

В этом случае алфавиты представляются собой прямые произведения более простых алфавитов. Таким образом:

$$U = \{a_{11}, \dots, a_{1J_1}\} \times \{a_{21}, \dots, a_{2J_2}\} \times \dots \times \{a_{m1}, \dots, a_{mJ_m}\} = \{s_a, a = 1, \dots, A; A = \prod_{i=1}^m J_i\},$$

$S_a$  – вариант сочетания алфавитов внешних входов  $u[t, s]$  клетки  $s$ ;

$$X = \{b_{11}, \dots, b_{1J_1}\} \times \{b_{21}, \dots, b_{2J_2}\} \times \dots \times \{b_{l1}, \dots, b_{lJ_l}\} = \{c_b, b = 1, \dots, B; B = \prod_{r=1}^n J_r\},$$

$x_b$  – вариант сочетания алфавитов внутренних выходов  $x[t, s]$  клетки  $s$ ;

$$\Xi = \{c_{11}, \dots, c_{1J_1}\} \times \{c_{21}, \dots, c_{2J_2}\} \times \dots \times \{c_{l1}, \dots, c_{lJ_l}\} = \{y_u, u = 1, \dots, N; N = \prod_{p=1}^l J_p\},$$

$y_u$  – вариант сочетания алфавитов состояний  $x[t, s]$  клетки  $s$ .

$$Y = \{d_{11}, \dots, d_{1J_1}\} \times \{d_{21}, \dots, d_{2J_2}\} \times \dots \times \{d_{l1}, \dots, d_{lJ_l}\} = \{t_g, g = 1, \dots, \Gamma; \Gamma = \prod_{q=1}^k J_q\},$$

$t_g$  – вариант сочетания алфавитов внешних выходов  $y[t, s]$  клетки  $s$ .

Переходы и выходы одномерной однонаправленной нелинейной итеративной сети представлены в табл. 3.1.

### 3.2. Одномерные двунаправленные итеративные сети

Уравнения (2.2) и (2.9) запишем для одномерной двунаправленной итеративной сети (рис. 3.3) в виде:

$$\begin{aligned} x(t, s) &= \Phi x(t-1, s) + \Psi_1 x(t, s-1) + \Psi_2 x(t, s+1) + \Psi_3 u(t, s), \\ x(t, s) &= H_1 x(t-1, s) + G_{11} x(t, s-1) + G_{12} x(t, s+1) + G_{13} u(t, s), \\ y(t, s) &= H_2 x(t-1, s) + G_{21} x(t, s-1) + G_{22} x(t, s+1) + G_{23} u(t, s). \end{aligned} \quad (3.4)$$

Это описание формально вполне идентично описанию двумерной полуторонаправленной цепи (2.28), шаблон пространственно-временного соседства соответствует показанному на рис. 2.12а. Для операторов  $x, Z, Z^{-1}$ , участвующих в описании сети, выполняется тождество  $ZZ^{-1} = 1$ , так что сеть аргументно-некаузальна. Это следует подчеркнуть, так как можно было ожидать, что такая сеть как пространственно-временная система автоматически каузальна благодаря наличию временной составляющей аргумента.

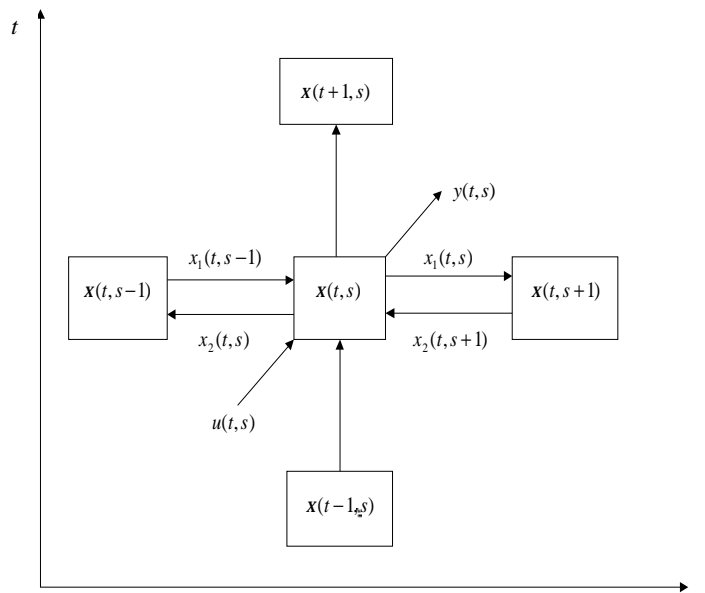


Рис. 3.3. Одномерная двунаправленная итеративная сеть

Сочетая (2.2) с (2.14), (2.26) и т.д., получим описания двумерных сетей как обобщений одномерных. Они не приводятся здесь в связи с громоздкостью и тем, что не имеют каких-либо явных особенностей (на уровне описания) по сравнению с уже рассмотренными сетями.

### 3.3. Одномерные линейные клеточные автоматы

Одномерные линейные клеточные автоматы – это частный случай подробно рассмотренных в подглаве 3.4 классических (двумерных) клеточных автоматов Дж. фон Неймана. Шаблон соседства такого автомата представлен на рис. 3.4. Одномерные клеточные автоматы описываются уравнением

$$x(t, s) = \Phi_1 x(t-1, s) + \Phi_2 x(t-1, s-1) + \Phi_3 x(t-1, s+1) + \Psi u(t, s) \quad (3.5)$$

и формально идентичны двумерным полуторонаправленным цепям (2.29).

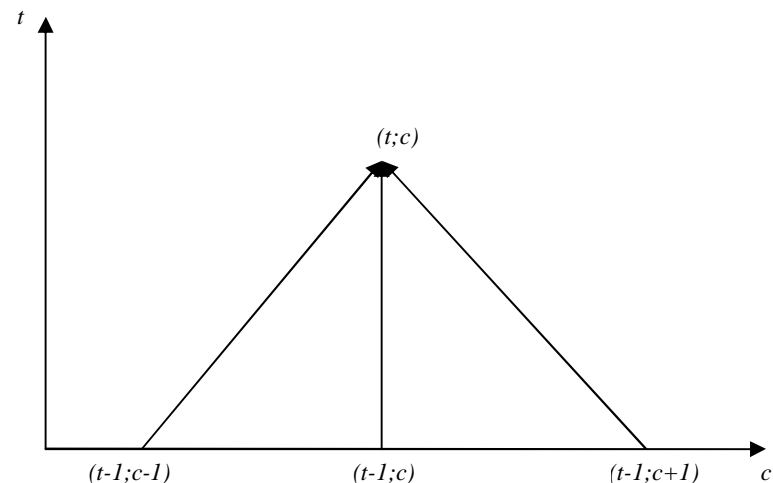


Рис. 3.4. Шаблон соседства одномерного клеточного автомата

Как и последние, одномерные автоматы аргументно-некаузальны, причем операторы  $x, xZ, xZ^{-1}$  удовлетворяет специальному соотношению  $xZ \cdot xZ^{-1} = x^2$ .

### 3.4. Клеточные конечные автоматы

Клеточные автоматы введены Дж. фон Нейманом [23] в связи с задачами автоматического конструирования и самовоспроизведения. Последние могут быть отнесены к задачам управления клеточными автоматами, но не

при помощи внешних управляющих воздействий, как в классической теории управления сосредоточенными системами, а при помощи начальных распределений состояний клеток клеточного пространства. Это определяет некоторые своеобразные аспекты теории клеточных автоматов, мотивирует постановку указанных задач в классе линейных клеточных автоматов и поиск решений в рамках алгебраических методов пространства состояний.

Клеточные автоматы эволюционируют в дискретном времени  $t \in T = \mathbb{Z}_0 = \{0, 1, 2, \dots\}$  и дискретном двумерном клеточном пространстве

$$c = (c_1, c_2) \in C = \mathbb{Z} \times \mathbb{Z}, \mathbb{Z} = \{\dots, -1, 0, 1, \dots\}.$$

$(\overline{C_1}, \overline{C_2})$  – координаты,

$C$  – клеточное пространство,

$\mathbb{Z} = \{\dots, -1, 0, 1, \dots\}$  – множество любых целых чисел.

Клеточное пространство представляется в виде плоскости–решетки (рис. 3.5).

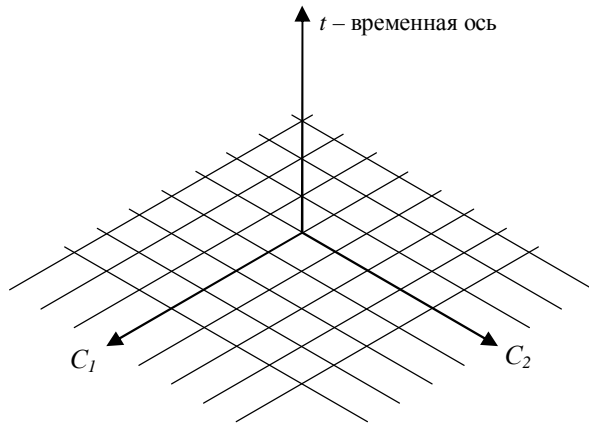


Рис. 3.5. Клеточное пространство клеточного конечного автомата

В каждый момент времени  $t$  каждая клетка  $\overline{C}$  может находиться в определенном состоянии:  $x(t; \overline{C}) = x(t; \overline{C_1}, \overline{C_2}) \in X$  – внутренний алфавит.

Если  $x$  – конечное множество, то клеточный автомат – конечный, но может таким и не быть.

Распределение состояний всех клеток в некоторый момент  $t$  называется конфигурацией и обозначается:

$$x(t) = \{x(t; C_1, C_2), (C_1, C_2) \in C\} = x. \quad (3.6)$$

Пара  $(C_1, C_2)$  пробегает все клеточное пространство.

Соответственно, данная конфигурация –  $x$ , предшествующая –  $x'$ . Как для автомата задается начальное состояние, так и для клеточного конечного автомата (ККА) задается начальная конфигурация, т.е. конфигурация в какой-то начальный момент времени:

$$x_0 = x(t_0). \quad (3.7)$$

Каждая клетка  $c$  может находиться в некотором состоянии  $x(c)$ ; распределение состояний клеток клеточного пространства называется конфигурацией и обозначается через  $\xi = \{x(c), c \in C\}$ . Если  $x(t; c) = x(t; c_1, c_2)$  – состояние клетки  $c$  в момент времени  $t$ , то предположенное в [23] правило переходов запишется в виде локального управления состояний

$$x(t; c_1, c_2) = j(x(t-1; c_1, c_2), x(t-1; c_1-1, c_2), x(t-1; c_1, c_2-1), x(t-1; c_1+1, c_2), x(t-1; c_1, c_2+1)). \quad (3.8)$$

Здесь  $j$  – переходная функция состояний клеточного автомата. Она не зависит в явном виде от времени  $t$  и пространства  $c$ , так что клеточный автомат является стационарной и однородной динамической системой. Это обосновано в [23]. Кроме того, в (3.8) в явном виде не входят внешние управляющие воздействия, так что клеточный автомат – автономная система.

Задается входное воздействие в каждый момент времени в каждой входной точке  $C$ :  $v(t; C) \in V$  – входной алфавит.

В этом случае для определения ККА в дискретном времени и пространстве задается переходная функция состояний клеточного автомата:

$$x(t; C_1, C_2) = j(x(t-1; C_1, C_2); x(t-1; C_1-1, C_2); x(t-1; C_1+1, C_2); x(t-1; C_1, C_2-1); x(t-1; C_1, C_2+1); v(t; C_1, C_2)), \quad (3.9)$$

где  $v(t; C) \in V$  – входное воздействие в каждый момент времени  $t$  в каждой точке  $C$  ( $V$  – входной алфавит).

Состояние в данный момент времени  $t$  в клетке  $C$  зависит от состояния в этой клетке в предыдущий момент времени и, кроме того, от состояния в предыдущий момент в других клетках клеточного пространства.

Графически характер перехода можно изобразить на рис. 3.6.

Выделенные точки образуют «шаблон соседства» данного ККА:

$$P(C) = P(C_1, C_2) = \{(C_1, C_2); (C_1-1, C_2); (C_1, C_2-1); (C_1+1, C_2); (C_1, C_2+1)\}. \quad (3.10)$$

Ввиду стационарности и однородности достаточно задать шаблон  $P$  клеточного автомата:

$$P = \{(0, 0); (-1, 0); (0, -1); (1, 0); (0, 1)\}. \quad (3.11)$$

Можно представить:

$$P(C) = C + P, \quad (3.12)$$

где  $P(C)$  – сдвиг шаблона  $P$  (из клетки 0) в клетку  $C$ .

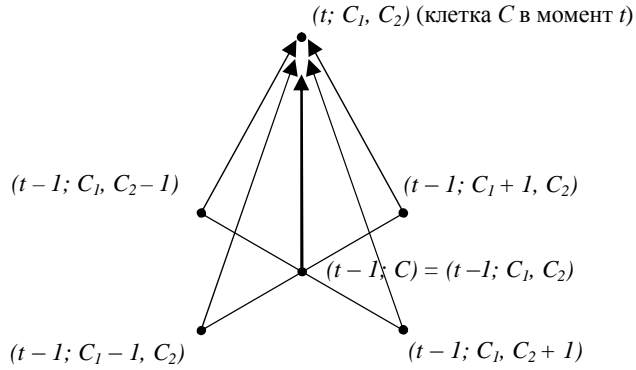


Рис. 3.6. Шаблон соседства клеточного конечного автомата

Для полного определения поведения клеточного автомата в дискретном пространстве – времени дополнительно к (3.8) должна быть задана начальная конфигурация  $x_0 = \{x_0(c), c \in C\}$  в некоторый начальный момент времени  $t_0 \in T$ , так что  $x_0 = \{x(t_0; c) = x_0(c), c \in C\}$ . Таким образом, клеточный автомат может рассматриваться как шестерка  $\langle T; C; P; j; t_0; x_0 \rangle$ .

Предполагается [23], что набор возможных состояний клеток включает специальное состояние  $\epsilon$ , называемое состоянием покоя и определяемое тем свойством, что если некоторая клетка  $c$  и все ее соседи (клетки из  $P(c)$ ) в момент времени  $t-1$  находятся в состоянии покоя, то клетка  $c$  в момент времени  $t$  также находится в состоянии покоя. Кроме того, предполагается, что в каждый момент времени все клетки, за исключением конечного числа, находятся в состоянии покоя. Для выполнения последнего условия в силу конечности шаблона  $P$  достаточно предположить, что в начальный момент времени все клетки, за исключением конечного числа, находятся в состоянии покоя. Клетки, которые не находятся в состоянии покоя, называются возбужденными.

Подмножества клеточного пространства будем называть массивами (клеток). Если массив  $A \subset C$  состоит из конечного числа клеток, то их количество  $|A|$  будем называть также площадью этого массива и обозначать через  $S(A)$  [24].

Для шаблона  $P$  клеточного автомата положим, что

$$P(0) = \{0\}, P(1) = P P(2) = P[P(1)] \quad (3.13)$$

– массив клеток, соседних с клетками шаблона  $P$  в силу этого шаблона, и вообще для любого

$$r\hat{I}Z_0 P(r) = P[P(r-1)] \quad (3.14)$$

– массив клеток, соседних с клетками массива  $P(r-1)$  в силу шаблона  $P$ . Массив  $P(r)$  назовем диском радиуса  $r$ . Ясно, что  $P(r)\hat{E}P(r-1)$  и при  $r \rightarrow \infty$  диск  $P(r)$  стремится покрыть все клеточное пространство  $C$ . При каждом  $r\hat{I}Z_0$  диск  $P(r)$  состоит из конечного числа клеток; его площадь

$$S(P(r)) = S(r) = 2r(r+1) + 1. \quad (3.15)$$

Конфигурация  $x = \{x(c), c\hat{I}C\}$  является финитной, если  $x(c) = \epsilon$ , за исключением конечного числа клеток. Наименьшее  $r\hat{I}Z_0$  такое, что  $x(c) = \epsilon$  для  $c\hat{I}P(r)$ , назовем радиусом  $r(x)$  финитной конфигурации  $x$ , а  $P(r(x)) = P(x)$  – ее диск-носителем. Ясно, что может быть  $x(c) = \epsilon$  и для некоторых  $c\hat{I}P(x)$ . Любой массив  $C(x)\hat{I}P(x)$ , такой, что  $x(c) = \epsilon$  для  $c\hat{I}C(x)$ , будем называть носителем финитной конфигурации  $x$ . Финитную конфигурацию  $x$  можно записать в виде  $x = \{x(c), c\hat{I}C(x)\}$ . Совокупность значений  $x(c)$ ,  $c\hat{I}C(x)$  будем называть составом конфигурации  $x$ .

Для любой клетки  $a\hat{I}C$  через  $z_a$  можно обозначить сдвиг клеточного пространства  $C$  на  $a$ :

$$z_a(c) = c + a; \quad (3.16)$$

для любой клетки

$$c\hat{I}C; z_a(A) = A + a; \quad (3.17)$$

для любого массива  $A\hat{I}C$ . Сдвиги в клеточном пространстве индуцируют сдвиги финитных конфигураций по правилу:

$$(t, t_0) = \Phi(t, t-1)\Phi(t-1, t-2) \dots \Phi(t_0+1, t_0), t > t_0. \quad (3.18)$$

Конфигурацию клеточного автомата в момент времени  $t\hat{I}Z_0$  обозначается  $x(t) = \{x(t, c), c\hat{I}C\}$ . В силу ранее сказанного конфигурации  $x(t)$  финитны; их радиусы обозначаем через  $r(t)$ , диск-носители – через  $P(t)$ , носители – через  $C(t)$ . Начальным моментом времени  $t_0$  клеточного автомата считается радиус  $r(x_0)$  его начальной конфигурации  $x_0$ . Его эволюция рассматривается во времени для  $t \geq t_0$ . Начальная конфигурация запишется в виде:

$$x_0 = x(t_0) = \{x(t_0; c), c \in C_0 = C(t_0) = C(x_0) \subset P(t_0)\}. \quad (3.19)$$

Тем самым, в принятых обозначениях,  $r(t) = t$ ,  $P(r(t)) = P(t)$  для  $t \geq t_0$ , так что обозначения дисков и диск-носителей конфигураций клеточного автомата согласуются.

### 3.5. Линейные клеточные автоматы.

#### Переход к ассоциированному уравнению

Ограничимся рассмотрением линейных клеточных автоматов (ЛКА), для которых при любых  $t\hat{I}T$  и  $c\hat{I}C$   $x(t; c)\hat{I}K^n$  ( $K$  – некоторое числовое поле,  $n$  – натуральное число – размерность вектора состояния каждой клетки ЛКА, определяющая размерность ЛКА), а уравнение (3.8) записывается в виде:

$$x(t; c_1, c_2) = \Phi_1 x(t-1; c_2, c_2) + \Phi_2 x(t-1; c_1-1, c_2) + \Phi_3 x(t-1; c_1, c_2-1) + A_4 x(t-1; c_1+1, c_2) + A_5 x(t-1; c_1, c_2+1), \quad (3.20)$$

где  $\Phi_1, \dots, \Phi_5$  – постоянные (не зависящие от  $t$  и  $c$ )  $n \times n$ -матрицы над полем  $K$ . Роль состояния покоя играет нулевой вектор  $0 \in K^n$ .

Пусть клетки клеточного пространства упорядочены (занумерованы) так, что клетка  $(0, 0)$  имеет номер 0, клетки диска  $P(t)$  имеют номера, не превосходящие

$$S(t) - 1 = 2t(t + 1), \quad (3.21)$$

клетки массива  $P(t)/P(t-1)$  имеют номера от

$$S(t-1) = 2t(t-1) + 1 \text{ до } S(t) - 1 = 2t(t+1) \quad (3.22)$$

при любом  $t \in Z_0$ . Методы нумерации клеточного пространства играют немаловажную роль, особенно в плане перенесения результатов данной работы на случай клеточных автоматов с другими возможными шаблонами [23-25]. Использование некоторых принципов нумерации при исследовании различных классов систем с дискретным аргументом представлено, например, в [18,20,26].

Нумерующая функция, осуществляющая взаимно-однозначное отображение клеточного пространства  $C = Z \times Z$  на множество  $Z_0$ , определяется при помощи алгоритма (здесь  $k = k(c) \in Z_0$  – номер клетки  $c \in C$ )

$$k = k(c) = 2 \lfloor |c| \rfloor (\lfloor |c| \rfloor + 1) + 1 + 4[c] + \{c\}, \quad (3.23)$$

$$\text{где } |c| = \begin{cases} |c_1| - 1, \\ |c_2| \end{cases}.$$

Каждому  $k \in Z_0$  сопоставляется клетка  $c = c(k) = (c_1(k), c_2(k)) \in C$  (имеющая номер  $k$  при указанной выше нумерации) при помощи алгоритма:

- 1) находим наибольшее  $k = \kappa(k)$ , такое, что  $2n(n-1) + 1 \leq k$ ;
- 2) находим наибольшее  $x = \chi(k) \in \{0, 1, \dots, n-1\}$ , такое, что  $4x \leq k - 2n(n-1) - 1$ ;
- 3) находим  $a = a(k) = k - 2n(n-1) - 1 - 4x$ , если  $k$  четно, то  $a \in \{1, 3\}$ ; если  $k$  нечетно, то  $a \in \{0, 2\}$ ;
- 4) если  $k$  четно, то

$$c_2(k) = \begin{cases} -(x+1) & \text{при } a=1, \\ x+1 & \text{при } a=3, \end{cases} \text{ и}$$

$$c_1(k) = \begin{cases} -(x+1) - n & \text{при } x=1, \\ n - (x+1) & \text{при } x=3; \end{cases} \quad (3.24)$$

5) если  $k$  нечетно, то

$$c_1(k) = \begin{cases} -(x+1) & \text{при } a=0, \\ x+1 & \text{при } a=2, \end{cases} \text{ и}$$

$$c_2(k) = \begin{cases} k - (x+1) & \text{при } a=0, \\ (x+1) - n & \text{при } a=2. \end{cases} \quad (3.25)$$

Клетка  $(0, 0)$  имеет номер 0; клетки шаблона  $P$  – номера 1, 2, 3, 4 в том порядке, в котором они записаны выше. Дальнейшее упорядочение рекуррентно: в диске  $P(t)$  первыми идут  $2t(t-1) + 1$  клеток диска  $P(t-1) \cap P(t)$  в том же порядке, что и в  $P(t-1)$ . Последними идут четыре клетки  $(-t, 0), (0, -t), (t, 0), (0, t)$ ; оставшиеся  $4(t-1)$  клеток, образующие массив  $P(t) \setminus (P(t-1) \cap P(t)) = \tilde{E}\{( -t, 0), (0, -t), (t, 0), (0, t)\}$ , следуют за клетками диска  $P(t-1)$  группами по четыре клетки и представляют собой последние  $t-1$  групп по четыре клетки массива  $P(t-1) \setminus P(t-2)$  (в том же порядке, что и в массиве  $P(t-1) \setminus P(t-2)$ ), причем в каждой группе к координатам клеток прибавляются соответственно координаты клеток  $\{(0, 1), (-1, 0), (0, -1), (1, 0)\}$ .

С описанным упорядочением клеток согласован алгоритм конструирования матриц  $\Phi(t, t-1)$  из матриц  $\Phi_1, \dots, \Phi_5$ , состоящий в выполнении следующих шагов:

- 1) матрица  $\Phi(1, 0)$  строится в виде  $\Phi(1, 0) = (\Phi_1^m \Phi_4^m \Phi_5^m \Phi_2^m \Phi_3^m)^m$ ;
- 2) матрица  $\Phi(2, 0)$  имеет вид (выделена матрица  $\Phi(1, 0)$ ):

$$\Phi(2, 1) = \begin{array}{c} \begin{matrix} & & & & & c = (c_1, c_2) & k(c) \end{matrix} \\ \begin{matrix} \Phi_1 & \Phi_2 & \Phi_3 & \Phi_4 & \Phi_5 & (0, 0) & 0 \\ \Phi_4 & \Phi_1 & 0 & 0 & 0 & (-1, 0) & 1 \\ \Phi_5 & 0 & \Phi_1 & 0 & 0 & (0, -1) & 2 \\ \Phi_2 & 0 & 0 & \Phi_1 & 0 & (1, 0) & 3 \\ \Phi_3 & 0 & 0 & 0 & \Phi_1 & (0, 1) & 4 \\ 0 & \Phi_3 & 0 & 0 & \Phi_4 & (-1, 1) & 5 \\ 0 & \Phi_5 & \Phi_4 & 0 & 0 & (-1, -1) & 6 \\ 0 & 0 & \Phi_2 & \Phi_5 & 0 & (1, -1) & 7 \\ 0 & 0 & 0 & \Phi_3 & \Phi_2 & (1, 1) & 8 \\ 0 & \Phi_4 & 0 & 0 & 0 & (-2, 0) & 9 \\ 0 & 0 & \Phi_5 & 0 & 0 & (0, -2) & 10 \\ 0 & 0 & 0 & \Phi_2 & 0 & (2, 0) & 11 \\ 0 & 0 & 0 & 0 & \Phi_3 & (0, 2) & 12 \end{matrix} \end{array} \quad (3.26)$$

$$\begin{array}{c|ccccc} c = (c_1, c_2): & (0,0) & (-1,0) & (0,-1) & (1,0) & (0,1) \\ \hline k(c): & 0 & 1 & 2 & 3 & 4 \end{array}$$

сбоку и снизу выписаны клетки и их номера, при помощи которых нумеруются строки и столбцы матрицы;

3) матрица  $\Phi(t, t-1)$  для  $t \in \mathbb{Z}$  строится из крупных блоков I-VI по схеме

$$\left[ \begin{array}{cccc|c} & & I & & V \\ - & - & \Phi(t-1, t-2) & - & - \\ & & II & & IV \\ \hline & & III & & VI \end{array} \right]$$

- а) блоки I и II составляют полученную на предыдущем шаге матрицу  $\Phi(t-1, t-2)$  размера  $S(t-1)n \times S(t-2)n$ ;
- б) блок III размера  $(S(t) - S(t-1))n \times S(t-2)n$  – нулевой;
- в) блок IV порядка  $(S(t-1) - S(t-2))n$  блочно-диагонален – на диагонали расположены матрицы  $\Phi_1$ ;
- г) блок V получен симметричным отражением блока II относительно диагонали блоков I и IV с заменой индексов  $2 \leftarrow 4, 3 \leftarrow 5$  у матриц  $\Phi_2, \Phi_3, \Phi_4, \Phi_5$ ;
- д) в верхнем правом углу последнего крупного блока VI размера  $(S(t) - S(t-1))n \times (S(t-1) - S(t-2))n$  помещается средний блок A порядка  $4n$ ;
- е) на выходящей из верхнего левого угла крупного блока VI диагонали расположены средние блоки B порядка  $4n$  (их  $t-1$ );
- ж) на входящей в нижний правый угол крупного блока VI диагонали расположены средние блоки C порядка  $4n$  (их  $t-1$ ).

В пп. д), е), ж) использованы средние блоки

$$A = \begin{bmatrix} 0 & 0 & 0 & \Phi_4 \\ \Phi_3 & 0 & 0 & 0 \\ 0 & \Phi_2 & 0 & 0 \\ 0 & 0 & \Phi_3 & 0 \end{bmatrix}, B = \begin{bmatrix} \Phi_3 & 0 & 0 & 0 \\ 0 & \Phi_4 & 0 & 0 \\ 0 & 0 & \Phi_5 & 0 \\ 0 & 0 & 0 & \Phi_2 \end{bmatrix}, C = \begin{bmatrix} \Phi_4 & 0 & 0 & 0 \\ 0 & \Phi_3 & 0 & 0 \\ 0 & 0 & \Phi_2 & 0 \\ 0 & 0 & 0 & \Phi_3 \end{bmatrix}. \quad (3.27)$$

Пусть  $X(t)\hat{I}K^{S(t)n}$  – составной вектор, образованный записанными в столбец векторами  $x(t; c)$ ,  $c \in \hat{I}P(t)$  в установленном порядке клеток  $X(t) = [x(t; 0)^m x(t; c_1)^m \dots x(t; c_{s(t-1)})^m]^m$ . (3.28)

При этом вектор  $X(t_0) \in K^{S(t_0)n}$  включает подвектор  $X_0(t_0; C_0)\hat{I}K^{S(C_0)n}$ , составленный из векторов  $x_0(c)$ ,  $c \in C_0$ , начальной конфигурации  $x_0$  (кроме них в  $X(t_0)$  входит нулевой вектор  $0\hat{I}K^n$  на местах, отвечающих клеткам  $c \in \hat{I}P(t_0) \setminus C_0$ ) в установленном порядке. Ввиду линейности клеточного автомата

та связь между векторами  $X(t-1)$  и  $X(t)$  линейна. Тем самым с уравнением (3.20) ассоциируется уравнение

$$X(t) = \Phi(t, t-1)X(t-1), \quad t > t_0, \quad (3.29)$$

при начальном условии  $X(t_0)$  это уравнение линейной автономной нестационарной дискретно-временной динамической системы с переменными во времени размерностями векторов и матриц. Оно представляет пространственно-глобальное описание данного ЛКА, исходно заданного локальным уравнением состояний (3.20). Переход от (3.20) к (3.29) может быть назван «упаковкой пространственной структуры» клеточного автомата (в векторы  $X(t)$  и матрицы  $\Phi(t, t-1)$  ассоциированного уравнения).

Матрица  $\Phi(t, t-1)$  – блочная. Она составлена из  $n \times n$ -блоков  $\Phi_1, \dots, \Phi_5$  и нулевых. Ее блочные размеры  $S(t) \times S(t-1)$ , полные размеры  $S(t)n \times S(t-1)n$ . Блочные строки матрицы  $\Phi(t, t-1)$  занумерованы номерами клеток массива  $P(t)$ , блочные столбцы – номерами клеток массива  $P(t-1)$  (в согласии с общей нумерацией клеток клеточного пространства). Решение уравнения (3.29) записывается в виде

$$X(t) = \Phi(t, t_0)X(t_0), \quad t \geq t_0, \quad (3.30)$$

где  $\Phi(t_0, t_0) = I_{S(t_0)n}$  – единичная матрица порядка  $S(t_0)n$ ,

$$\Phi(t, t_0) = \Phi(t, t-1)\Phi(t-1, t-2) \dots \Phi(t_0+1, t_0), \quad t > t_0. \quad (3.31)$$

Матрица  $\Phi(t, t_0)$  имеет размеры  $S(t)n \times S(t_0)n$ .

ЛКА более подробно можно характеризовать как кортеж

$$\langle T; C; P; \Phi_1, \Phi_2, \Phi_3, \Phi_4, \Phi_5; t_0; C_0; x_0 \rangle. \quad (3.32)$$

Пусть заданы  $x^* = \{x^*(c), c \in \hat{I}P(x^*) = C^*\}$  и  $t^* \hat{I}Z_0$  – «желаемая» конфигурация, которую требуется сконструировать при помощи заданного ЛКА, и время, за которое это требуется сделать. Данный ЛКА способен сконструировать конфигурацию  $x^*$  за время  $t^*$ , если в силу (3.20), (3.32)

$$x(t_0 + t^*; c) = x^*(c), \quad c \in \hat{I}P(x^*). \quad (3.33)$$

Ясно, что задача имеет смысл при  $t^* > 0$  и  $t_0 + t^* \in r(x^*)$ . В момент  $t_0 - t^*$  могут оказаться возбужденными и некоторые клетки  $c \in \hat{I}P(x^*)$ , а именно: клетки  $c \in \hat{I}P(x^*) \setminus C^*$ . О состояниях этих клеток, вообще говоря, никаких предположений не делается. Если же потребовать дополнительно, чтобы такие клетки оказались в состоянии покоя, то необходимо говорить о чистом конструировании конфигурации  $x^*$  за время  $t^*$  при помощи данного ЛКА.

Наряду с вектором  $X_0(t_0, C_0)$  вводится в рассмотрение вектор  $X(t_0 + t^*; c^*)\hat{I}K^{S(c^*)n}$  – подвектор составного вектора  $X(t_0 + t^*)\hat{I}K^{S(t_0+t^*)n}$ , полученный из последнего исключением составляющих  $X(t_0 + t^*; c)$ ,  $c \in \hat{I}P(t_0 + t^*) \setminus C^*$  с сохранением установленного порядка оставшихся векторов  $x(t_0 + t^*; c)$ ,  $c \in C^*$ . Пусть далее  $\Phi(t_0 + t^*, t_0; C^*, C_0)$  – матрица, полученная из матрицы (3.31)  $\Phi(t_0 + t^*, t_0)$  исключением блочных строк с номерами клеток  $c \in \hat{I}P(t_0 + t^*) \setminus C^*$  и блочных столбцов с номерами клеток  $c \in \hat{I}P(t_0) \setminus C_0$  с со-

хранением установленного порядка оставшихся блочных строк и столбцов. Эта матрица имеет размеры  $S(C^*)n \times S(C_0)n$ . Тогда в силу (3.30)

$$X(t_0 + t^*; C^*) = \Phi(t_0 + t^*; t_0; C^*, C_0)X_0(t_0, C_0). \quad (3.34)$$

С другой стороны, (3.33) можно записать в виде:

$$X(t_0 + t^*; C^*) = X^*, \quad (3.35)$$

где  $X^* \hat{I} K^{(c^*)n}$  – вектор, ассоциированный описанным выше способом с конфигурацией  $x^*$ . Он может быть записан подробнее в виде  $X^*(r(x^*); C^*)$ , а также, поскольку  $P(r(x^*)) \hat{I} P(t_0 + t^*)$  в силу  $t_0 + t^* \in r(x^*)$ , в виде  $X^*(t_0 + t^*; C^*)$ .

Итак, способность заданного ЛКА сконструировать заданную конфигурацию  $x^*$  за заданное время  $t^*$  равносильна выполнению соотношения

$$\Phi(t_0 + t^*, t_0; C^*, C_0)X_0(t_0; C_0) = X^*(t_0 + t^*; C^*). \quad (3.36)$$

Пусть в (3.36) зафиксировано все, кроме состава конфигураций  $x_0$  и  $x^*$ , т.е. для ЛКА зафиксированы  $\Phi_1, \dots, \Phi_5, t_0$ , а для конфигурации – их носители  $C_0$  и  $C^*$ . Запишем (3.36) сокращенно в виде

$$\Phi_0^* X_0 = X^*. \quad (3.37)$$

Теперь алгебраические критерии разрешимости линейного уравнения (3.37) [27] доставляют критерии способности ЛКА к конструированию.

ЛКА способен сконструировать конфигурацию  $x^*$  за время  $t^*$  (в том смысле, что найдется конфигурация, ассоциированный с которой вектор  $X_0$  удовлетворяет (3.37)) тогда и только тогда, когда ассоциированный с конфигурацией  $x^*$  вектор  $X^*$  и матрица  $\Phi_0^*$  удовлетворяют соотношению

$$\Phi_0^*(\Phi_0^*)X^* = X^*, \quad (3.38)$$

где  $(\Phi_0^*)^+$  – псевдообратная к матрице  $\Phi_0^*$  [27].

ЛКА способен сконструировать любую конфигурацию  $x^*$  с заданным носителем  $C^*$  тогда и только тогда, когда

$$rg \Phi_0^* = S(C^*)n. \quad (3.39)$$

Ясно, что последний критерий имеет смысл лишь при  $S(C^*) \leq S(C_0)$ . В противном случае речь может идти лишь об оптимальном конструировании конфигураций при помощи ЛКА в смысле минимума среднеквадратического отклонения конструируемой конфигурации от желаемой, когда (3.33) заменяется на

$$\Sigma S_{c \hat{I} C^*} \|x(t_0 + t^*; c) - x^*(c)\|^2 @ min, \quad (3.40)$$

где  $\| \dots \|$  – некоторая норма в  $K^n$  (например, обычная евклидова).

Задача стирания конфигурации  $\xi_0$  за время  $t^*$  является частным случаем задачи конструирования, когда требуется, чтобы

$$x(t_0 + t^*; c) \circ 0, c \hat{I} P(t_0 + t^*). \quad (3.41)$$

В этом случае соотношение (3.36) запишется в виде

$$\Phi(t_0 + t^*, t_0; P(t_0 + t^*), C_0)X_0(t_0; C_0) = 0. \quad (3.42)$$

Составные векторы  $X_0$ , ассоциированные с конфигурациями  $x_0$ , которые данный ЛКА способен стереть за время  $t$ , образуют ядро матрицы  $\Phi(t_0 + t^*, t_0; P(t_0 + t^*), C_0)$  в пространстве  $K^{S(C_0)n}$ .

В задаче чистого конструирования конфигурации  $x^*$  к соотношению (3.36) добавляется соотношение

$$\Phi(t_0 + t^*, t_0; P(t_0 + t^*)/C_0)X_0(t_0; C_0) = 0. \quad (3.43)$$

Эта задача разрешима, если множество решений задачи конструирования, если оно непусто, имеет непустое пересечение с ядром матрицы, фигурирующей в (3.43).

Задача воспроизведения конфигурации  $\xi_0$  через время  $t^*$  «в месте»  $a$  клеточного пространства является частным случаем задачи конструирования, когда требуется, чтобы  $x^* = z_a(x_0)$ , т.е.  $x^* = \{x^*(c) = x_0(c - a), c \hat{I} C^* = z_a(C_0)\}$ .

Условие (3.33) запишется в виде  $x(t_0 + t^*; c) = x_0(c - a)$ ,  $c \in z_a(C_0)$ .

По-прежнему должно быть  $t_0 + t^* \in r(z_a(x_0))$ . В этом случае матрица  $\Phi_0^*(a) = \Phi(t_0 + t^*, t_0^*, z_a(C_0), C_0)$  – квадратная, блочного размера  $S(C_0) \times S(C_0)$ . Вектор  $X^*$  может отличаться от  $X_0$  разве что порядком составляющих его векторов  $x_0(c)$ ,  $c \hat{I} C_0$ . Пусть  $I_{S(C_0)n}(a)$  – матрица, в которой  $S(C_0) \times n$ -блоков – единичные, а остальные – нулевые, такая, что  $X^* = I(a)X_0$ . Тогда (3.37) запишется в виде

$$\Phi_0^*(a)X_0 = I(a)X_0 \text{ или } [\Phi_0^*(a) - I(a)]X_0 = 0. \quad (3.44)$$

Таким образом, векторы  $X_0$ , отвечающие конфигурациям  $x_0$ , которые данный ЛКА способен воспроизводить через время  $t^*$  в месте  $a$  клеточного пространства, образуют ядро матрицы  $\Phi_0^*(a) - I(a)$ . ЛКА способен воспроизводить нетривиальные конфигурации тогда и только тогда, когда

$$\det[\Phi(t_0 + t^*, t_0; z_a(C_0), C_0) - I(a)] = 0. \quad (3.45)$$

Состав таких конфигураций определяется решениями уравнения (3.44).

Задача восстановления конфигурации  $\xi_0$  через время  $t^*$  является частным случаем задачи воспроизведения, когда  $a = 0$ . Критерий способности данного ЛКА восстанавливать через время  $t^*$  нетривиальные конфигурации  $x_0$  имеет вид

$$\det(\Phi(t_0 + t^*, t_0; C_0, C_0) - I) = 0, \quad (3.46)$$

а состав таких конфигураций определяется решениями соответствующей однородной системы уравнений типа (3.44).

Задача размножения конфигурации  $x_0$  за время  $t^*$  в местах  $a_1, \dots, a_m$  клеточного пространства является частным случаем задачи конструирования, когда (3.20) записывается в виде

$$x(t_0 + t^*; c) = x_0(c - a_i), c \hat{I} z_{a_i}(C_0), i = 1, \dots, m. \quad (3.47)$$



При этом предполагается, что  $z_{a_i}(C_0) \cap z_{a_j}(C_0) = f$  для любых  $i, j = 1, \dots, m, i \neq j$ , а под носителем  $C^*$  желаемой конфигурации понимается  $U_{i=1}^m z_{a_i}(c_0)$ . Теперь одновременно должны выполняться  $m$  соотношений типа (3.44)

$$[\Phi_0^*(a_i) - I(a_i)]X_0 = 0, \quad i = 1, \dots, m. \quad (3.48)$$

Данный ЛКА способен размножать указанным образом нетривиальные конфигурации  $x_0$  тогда и только тогда, когда элементы определяющих его матриц  $\Phi_1, \dots, \Phi_5$  удовлетворяют одновременно  $m$  полиномиальным уравнениям

$$\det[\Phi(t_0 + t^*, t_0; z_{a_i}^i(C_0), C_0) - I(a_i)] = 0, \quad t = 1, \dots, m. \quad (3.49)$$

Проверка выполнения этого критерия может быть осуществлена, например, методами теории результатов [28]. Состав конфигураций, которые данный ЛКА способен размножать указанным образом, определяется пересечением ядер матриц  $\Phi_0^*(a_i) - I(a_i), i = 1, \dots, m$ .

### 3.6. Аналитическое представление ККА

На выходе клетки ККА образуется выходной сигнал  $y(t; C) \hat{I} Y$  ( $Y$  – выходной алфавит клеточного конечного автомата). Образование выходного сигнала определяется функцией выходов, которая может задаваться в двух вариантах:

а) ККА Миля:

$$y(t; C_1, C_2) = y(x(t-1; C_1, C_2), x(t-1; C_1-1, C_2), x(t-1; C_1+1, C_2), x(t-1; C_1, C_2-1), x(t-1; C_1, C_2+1), n(t; C_1, C_2)); \quad (3.50)$$

б) ККА Мура:

$$y(t; C_1, C_2) = \tilde{y}(x(t); C_1, C_2). \quad (3.51)$$

Если функции переходов и выходов в явном виде не зависят от времени и пространства (в неявном – зависят), а единый шаблон  $P$  путем сдвига определяет шаблон в каждой точке пространства, то ККА представляет собой стационарную (не зависит от  $t$ ) и однородную (не зависит от пространства) систему или однородную структуру.

Если отсутствует зависимость от входа, то ККА – автономная система. Именно в таком виде ККА были введены фон Нейманом как модели вычислительных устройств автоматического конструирования и самовоспроизведения.

В случае конечных алфавитов ККА (как и конечных автоматов) может быть задан при помощи таблиц, однако их объем существенно больше, чем у конечного автомата за счет шаблона соседства (у конечного автомата шаб-

лон из одной точки, у ККА – из пяти). По этой причине практически невозможно представление ККА при помощи графов (т.к. здесь одно состояние зависит от пяти состояний, а не одного).

Например, вершиной графа должно было бы считаться множество из пяти точек.

Рассмотрим аналитическое представление клеточного конечного автомата. Ограничимся автономным ККА без выхода (т.е. сосредоточим внимание на функции перехода) с одномерным состоянием и двоичным внутренним алфавитом  $X = \{0, 1\}$  и любым  $x(t; c) \in X$  (в момент  $t$  в точке  $c$ ). В этом случае переходная функция имеет вид:

$$x(t; c) = j(x(t-1; c_1; c_2); x(t-1; c_1-1; c_2); x(t-1; c_1+1; c_2); x(t-1; c_1; c_2-1); x(t-1; c_1; c_2+1)). \quad (3.52)$$

Табличное представление такой функции должно содержать  $2^5$  строк (тогда как для конечного автомата было две строки (без входа)), так как используется двоичный внутренний алфавит и шаблон соседства содержит пять точек (табл. 3.2).

В случае клеточного конечного автомата принято учитывать следующие ограничения.

1. Предполагается, что набор возможных состояний клеток включает специальное состояние 0, называемое состоянием покоя и определяемое тем свойством, что если некоторая клетка  $C$  и все ее соседи в момент времени  $t-1$  находятся в состоянии покоя, то клетка  $C$  в момент времени  $t$  также находится в состоянии покоя.

2. Кроме того, предполагается, что в каждый момент времени все клетки, за исключением конечного числа, находятся в состоянии покоя. Для выполнения последнего условия, в силу конечности шаблона  $P$ , достаточно предположить, что в начальный момент времени все клетки, за исключением конечного числа, находятся в состоянии покоя. Клетки, которые не находятся в состоянии покоя, называются возбужденными. Конфигурация клеточного конечного автомата, в которой все клетки, кроме конечного их числа, находятся в состоянии покоя, называется финитной.

Чтобы выполнялось данное требование (т.е. все конфигурации были финитны), достаточно начальную конфигурацию задать финитной.

Получив исходную конфигурацию, с использованием функции перехода можно рассчитать конфигурацию автомата в любой момент  $t \geq 0$ , т.е. состояние любой клетки в любой момент времени, следовательно, рассматриваемый ККА детерминированный, а не вероятностный.

В соответствии с табл. 3.2, функция переходов задается как функция пяти переменных двузначной логики.

Примеры начальных конфигураций представлены на рис. 3.7.

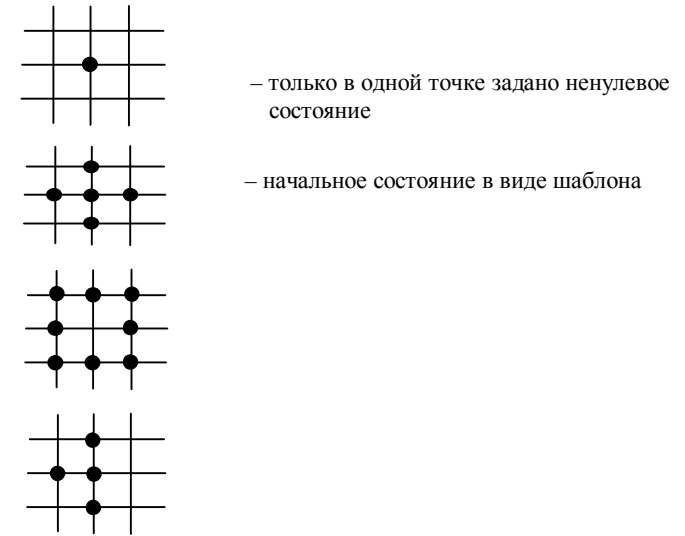


Рис. 3.7. Примеры начальных конфигураций

Для машинного моделирования наиболее удобно представлять функцию переходов в виде многочлена Жегалкина:

$$\begin{aligned}
 X^* = j(x_1, x_2, x_3, x_4, x_5) = & a_0 \oplus a_1 \otimes x_1 \oplus a_2 \otimes x_2 \oplus \dots \oplus a_5 \otimes x_5 \oplus \\
 & \oplus a_{12} \otimes x_1 \otimes x_2 \oplus \dots \oplus a_{45} \otimes x_4 \otimes x_5 \oplus a_{123} \otimes x_1 \otimes x_2 \otimes x_3 \oplus \dots \oplus \\
 & \oplus a_{345} \otimes x_3 \otimes x_4 \otimes x_5 \oplus a_{1234} \otimes x_1 \otimes x_2 \otimes x_3 \otimes x_4 \oplus \dots \oplus a_{2345} \oplus \\
 & \otimes x_2 \otimes x_3 \otimes x_4 \otimes x_5 \oplus a_{12345} \otimes x_1 \otimes x_2 \otimes x_3 \otimes x_4 \otimes x_5 .
 \end{aligned} \quad (3.53)$$

где  $X^*$  – состояние;  $a_0$  – свободный член;  $a_1, \dots, a_{12345}$  – коэффициенты, которые нужно найти.

Число членов (и соответственно коэффициентов) данного многочлена:

$$2^5 = C_5^0 C_5^1 C_5^2 C_5^3 C_5^4 C_5^5 = 1 + 5 + 10 + 10 + 5 + 1 = 32 .$$

Все коэффициенты данного многочлена определяются путем составления по табл. 3.2 и решения системы 32 линейных уравнений с 32 неизвестными.

После получения многочлена Жегалкина расчеты конфигурации ККА могут быть реализованы на ЭВМ.

**Пример 3.1**

Дано:

- 1) переходная функция (табл. 3.3);
- 2) финитная конфигурация.

Найти:

- 1) конфигурацию в следующий момент времени;
- 2) радиус;
- 3) площадь.

Переходная функция ККА

Таблица 3.3

№ п/п	$x(t-1; c_1; c_2)$	$x(t-1; c_1-1; c_2)$	$x(t-1; c_1+1; c_2)$	$x(t-1; c_1; c_2-1)$	$x(t-1; c_1; c_2+1)$	$x(t; c_1; c_2)$
1	0	0	0	0	0	0
2	0	0	0	0	1	1
3	0	0	0	1	0	1
4	0	0	0	1	1	0
5	0	0	1	0	0	0
6	0	0	1	0	1	0
7	0	0	1	1	0	1
8	0	0	1	1	1	1
9	0	1	0	0	0	0
10	0	1	0	0	1	1
11	0	1	0	1	0	1
12	0	1	0	1	1	0
13	0	1	1	0	0	0
14	0	1	1	0	1	0
15	0	1	1	1	0	0
16	0	1	1	1	1	1
17	1	0	0	0	0	1
18	1	0	0	0	1	1
19	1	0	0	1	0	1
20	1	0	0	1	1	1
21	1	0	1	0	0	0
22	1	0	1	0	1	0
23	1	0	1	1	0	1
24	1	0	1	1	1	1
25	1	1	0	0	0	0
26	1	1	0	0	1	0
27	1	1	0	1	0	0
28	1	1	0	1	1	0
29	1	1	1	0	0	0
30	1	1	1	0	1	1
31	1	1	1	1	0	1
32	1	1	1	1	1	1

**многочлен Жегалкина**

Общий вид:

$x(t; c1, c2) = a0$	(*)	(*)	(*)	(*)	(*)	(+)			
a5	(*)	(*)	(*)	(*)	(*)	$x(t-1; c1, c2+1)$ (+)			
a4	(*)	(*)	(*)	(*)	$x(t-1; c1, c2-1)$	(*) (+)			
a45	(*)	(*)	(*)	(*)	$x(t-1; c1, c2-1)$	(*) $x(t-1; c1, c2+1)$ (+)			
a3	(*)	(*)	$x(t-1; c1+1, c2)$	(*)	(*)	(*) (+)			
a35	(*)	(*)	$x(t-1; c1+1, c2)$	(*)	(*)	$x(t-1; c1, c2+1)$ (+)			
a34	(*)	(*)	$x(t-1; c1+1, c2)$	(*)	$x(t-1; c1, c2-1)$	(*) (+)			
a345	(*)	(*)	$x(t-1; c1+1, c2)$	(*)	$x(t-1; c1, c2-1)$	(*) $x(t-1; c1, c2+1)$ (+)			
a2	(*)	$x(t-1; c1-1, c2)$	(*)	(*)	(*)	(*) (+)			
a25	(*)	$x(t-1; c1-1, c2)$	(*)	(*)	(*)	$x(t-1; c1, c2+1)$ (+)			
a24	(*)	$x(t-1; c1-1, c2)$	(*)	(*)	$x(t-1; c1, c2-1)$	(*) (+)			
a245	(*)	$x(t-1; c1-1, c2)$	(*)	(*)	$x(t-1; c1, c2-1)$	(*) $x(t-1; c1, c2+1)$ (+)			
a23	(*)	$x(t-1; c1-1, c2)$	(*)	$x(t-1; c1+1, c2)$	(*)	(*) (+)			
a235	(*)	$x(t-1; c1-1, c2)$	(*)	$x(t-1; c1+1, c2)$	(*)	(*) $x(t-1; c1, c2+1)$ (+)			
a234	(*)	$x(t-1; c1-1, c2)$	(*)	$x(t-1; c1+1, c2)$	(*)	$x(t-1; c1, c2-1)$	(*) (+)		
a2345	(*)	$x(t-1; c1-1, c2)$	(*)	$x(t-1; c1+1, c2)$	(*)	$x(t-1; c1, c2-1)$	(*) $x(t-1; c1, c2+1)$ (+)		
a1	(*)	$x(t-1; c1, c2)$	(*)	(*)	(*)	(*) (+)			
a15	(*)	$x(t-1; c1, c2)$	(*)	(*)	(*)	(*) $x(t-1; c1, c2+1)$ (+)			
a14	(*)	$x(t-1; c1, c2)$	(*)	(*)	(*)	$x(t-1; c1, c2-1)$	(*) (+)		
a145	(*)	$x(t-1; c1, c2)$	(*)	(*)	(*)	$x(t-1; c1, c2-1)$	(*) $x(t-1; c1, c2+1)$ (+)		
a13	(*)	$x(t-1; c1, c2)$	(*)	(*)	$x(t-1; c1+1, c2)$	(*)	(*) (+)		
a135	(*)	$x(t-1; c1, c2)$	(*)	(*)	$x(t-1; c1+1, c2)$	(*)	(*) $x(t-1; c1, c2+1)$ (+)		
a134	(*)	$x(t-1; c1, c2)$	(*)	(*)	$x(t-1; c1+1, c2)$	(*)	$x(t-1; c1, c2-1)$	(*) (+)	
a1345	(*)	$x(t-1; c1, c2)$	(*)	(*)	$x(t-1; c1+1, c2)$	(*)	$x(t-1; c1, c2-1)$	(*) $x(t-1; c1, c2+1)$ (+)	
a12	(*)	$x(t-1; c1, c2)$	(*)	$x(t-1; c1-1, c2)$	(*)	(*)	(*) (+)		
a125	(*)	$x(t-1; c1, c2)$	(*)	$x(t-1; c1-1, c2)$	(*)	(*)	(*) $x(t-1; c1, c2+1)$ (+)		
a124	(*)	$x(t-1; c1, c2)$	(*)	$x(t-1; c1-1, c2)$	(*)	(*)	$x(t-1; c1, c2-1)$	(*) (+)	
a1245	(*)	$x(t-1; c1, c2)$	(*)	$x(t-1; c1-1, c2)$	(*)	(*)	$x(t-1; c1, c2-1)$	(*) $x(t-1; c1, c2+1)$ (+)	
a123	(*)	$x(t-1; c1, c2)$	(*)	$x(t-1; c1-1, c2)$	(*)	$x(t-1; c1+1, c2)$	(*)	(*) (+)	
a1235	(*)	$x(t-1; c1, c2)$	(*)	$x(t-1; c1-1, c2)$	(*)	$x(t-1; c1+1, c2)$	(*)	(*) $x(t-1; c1, c2+1)$ (+)	
a1234	(*)	$x(t-1; c1, c2)$	(*)	$x(t-1; c1-1, c2)$	(*)	$x(t-1; c1+1, c2)$	(*)	$x(t-1; c1, c2-1)$	(*) (+)
a12345	(*)	$x(t-1; c1, c2)$	(*)	$x(t-1; c1-1, c2)$	(*)	$x(t-1; c1+1, c2)$	(*)	$x(t-1; c1, c2-1)$	(*) $x(t-1; c1, c2+1)$ (+)

**Коэффициенты**

a0	= 0	a1	= 1
a5	= 1	a15	= 1
a4	= 1	a14	= 1
a45	= 0	a145	= 0
a3	= 0	a13	= 1
a35	= 1	a135	= 1
a34	= 0	a134	= 1
a345	= 0	a1345	= 0
a2	= 0	a12	= 1
a25	= 0	a125	= 0
a24	= 0	a124	= 0
a245	= 0	a1245	= 0
a23	= 0	a123	= 1
a235	= 0	a1235	= 1
a234	= 1	a1234	= 1
a2345	= 1	a12345	= 0

$$\begin{aligned}
x(t;c_1,c_2) = & \quad (*) \quad (*) \quad (*) \quad (*) x(t-1;c_1,c_2+1) \quad (+) \\
& \quad (*) \quad (*) \quad (*) x(t-1;c_1,c_2-1) \quad (*) \quad (+) \\
& \quad (*) \quad (*) x(t-1;c_1+1,c_2) \quad (*) \quad (*) x(t-1;c_1,c_2+1) \quad (+) \\
& \quad (*) x(t-1;c_1-1,c_2) \quad (*) x(t-1;c_1+1,c_2) \quad (*) x(t-1;c_1,c_2-1) \quad (*) \quad (+) \\
& \quad (*) x(t-1;c_1-1,c_2) \quad (*) x(t-1;c_1+1,c_2) \quad (*) x(t-1;c_1,c_2-1) \quad (*) x(t-1;c_1,c_2+1) \quad (+) \\
x(t-1;c_1,c_2) \quad & (*) \quad (*) \quad (*) \quad (*) \quad (+) \\
x(t-1;c_1,c_2) \quad & (*) \quad (*) \quad (*) \quad (*) x(t-1;c_1,c_2+1) \quad (+) \\
x(t-1;c_1,c_2) \quad & (*) \quad (*) \quad (*) x(t-1;c_1,c_2-1) \quad (*) \quad (+) \\
x(t-1;c_1,c_2) \quad & (*) \quad (*) x(t-1;c_1+1,c_2) \quad (*) \quad (*) \quad (+) \\
x(t-1;c_1,c_2) \quad & (*) \quad (*) x(t-1;c_1+1,c_2) \quad (*) \quad (*) \quad (+) \\
x(t-1;c_1,c_2) \quad & (*) \quad (*) x(t-1;c_1+1,c_2) \quad (*) \quad (*) x(t-1;c_1,c_2+1) \quad (+) \\
x(t-1;c_1,c_2) \quad & (*) \quad (*) x(t-1;c_1+1,c_2) \quad (*) \quad (*) x(t-1;c_1,c_2-1) \quad (+) \\
x(t-1;c_1,c_2) \quad & (*) x(t-1;c_1-1,c_2) \quad (*) \quad (*) \quad (*) \quad (+) \\
x(t-1;c_1,c_2) \quad & (*) x(t-1;c_1-1,c_2) \quad (*) x(t-1;c_1+1,c_2) \quad (*) \quad (*) \quad (+) \\
x(t-1;c_1,c_2) \quad & (*) x(t-1;c_1-1,c_2) \quad (*) x(t-1;c_1+1,c_2) \quad (*) \quad (*) x(t-1;c_1,c_2+1) \quad (+) \\
x(t-1;c_1,c_2) \quad & (*) x(t-1;c_1-1,c_2) \quad (*) x(t-1;c_1+1,c_2) \quad (*) x(t-1;c_1,c_2-1) \quad (*) \quad (+)
\end{aligned}$$

Значения  $x(t,c_1,c_2)$  при различных значениях выборки приведены на рис. 3.8, 3.11, а финитные конфигурации – на рис. 3.9, 3.10.

0	1	0	1	0	1	0	1
000	000	000	000	001	001	001	001
0	0	1	1	0	0	1	1
0	1	1	0	0	0	1	1

0	1	0	1	0	1	0	1
100	100	100	100	101	101	101	101
0	0	1	1	0	0	1	1
0	1	1	0	0	0	0	1

0	1	0	1	0	1	0	1
010	010	010	010	011	011	011	011
0	0	1	1	0	0	1	1
0	1	1	1	0	0	1	1

0	1	0	1	0	1	0	1
110	110	110	110	111	111	111	111
0	0	1	1	0	0	1	1
0	0	0	0	0	1	1	1

Рис. 3.8. Значение  $x(t,c_1,c_2)$  при различных шаблонах соседства

$t = 0$ : возбужденных клеток 10, радиус  $r = 6$ , центр в клетке (3, 3), площадь 85.

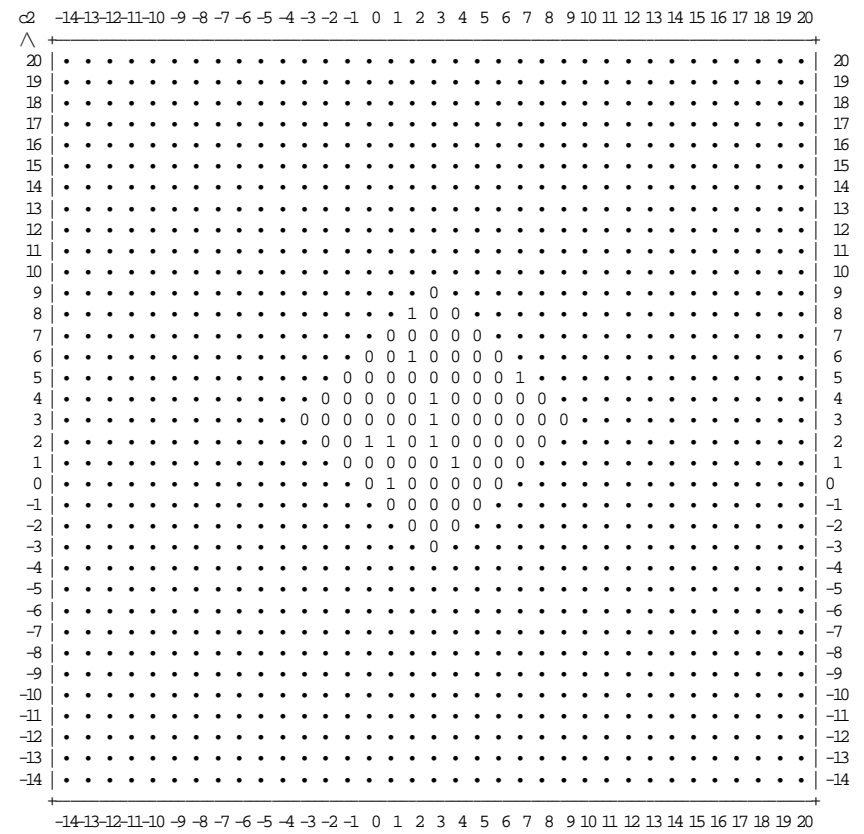


Рис. 3.9. Конфигурация клеток в момент  $t = 0$

$t = 1$ : возбужденных клеток 19, радиус  $r = 7$ , центр в клетке (3, 3), площадь 113.

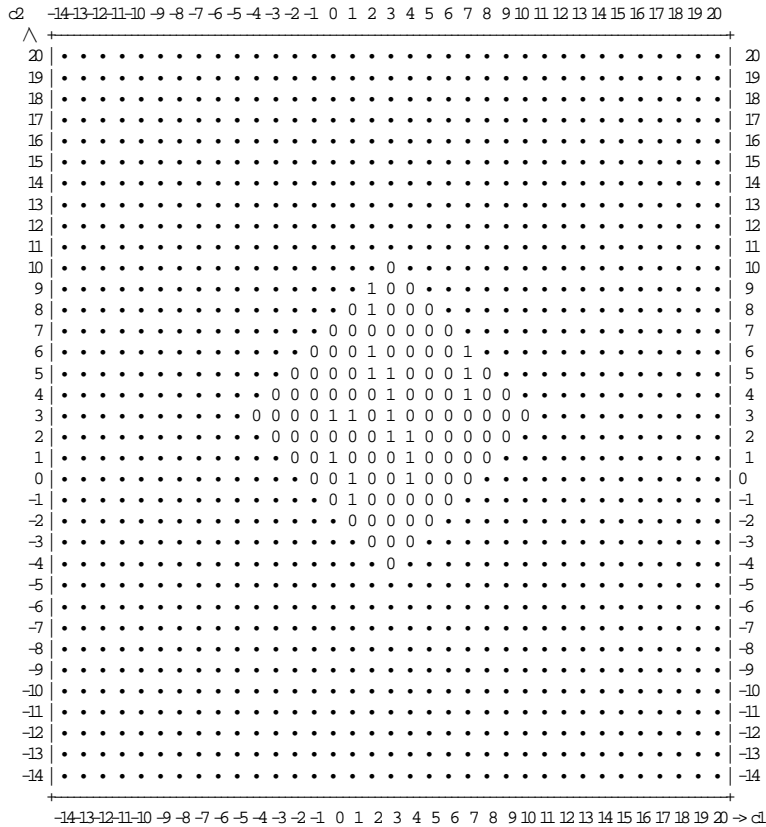


Рис. 3.10. Конфигурация клеток в момент  $t = 1$

### Пример 3.2

Дано:

1) переходная функция

Найти:

1) коэффициенты многочлена Жегалкина

2) шаблоны соседства для всех строк переходной функции.

Переходная функция представлена в табл. 3.3.

### Решение

Общий вид многочлена Жегалкина:

$$\begin{aligned}
 x(t; c_1, c_2) = & a_0 \mathring{A} a_5 x(t-1; c_1, c_2+1) \mathring{A} a_4 x(t-1; c_1, c_2-1) \mathring{A} a_{45} x(t-1; c_1, c_2-1) x(t-1; c_1, c_2+1) \mathring{A} \\
 & \mathring{A} a_{33} x(t-1; c_1+1, c_2) \mathring{A} a_{35} x(t-1; c_1+1, c_2) x(t-1; c_1, c_2+1) \mathring{A} \\
 & \mathring{A} a_{34} x(t-1; c_1+1, c_2) x(t-1; c_1, c_2-1) \mathring{A} a_{345} x(t-1; c_1+1, c_2) x(t-1; c_1, c_2-1) x(t-1; c_1, c_2+1) \mathring{A} \\
 & \mathring{A} a_2 x(t-1; c_1-1, c_2) \mathring{A} a_{25} x(t-1; c_1-1, c_2) x(t-1; c_1, c_2+1) \mathring{A} \\
 & \mathring{A} a_{24} x(t-1; c_1-1, c_2) x(t-1; c_1, c_2-1) \mathring{A} a_{245} x(t-1; c_1-1, c_2) x(t-1; c_1, c_2-1) x(t-1; c_1, c_2+1) \mathring{A} \\
 & \mathring{A} a_{23} x(t-1; c_1-1, c_2) x(t-1; c_1+1, c_2) \mathring{A} a_{235} x(t-1; c_1-1, c_2) x(t-1; c_1+1, c_2) x(t-1; c_1, c_2+1) \mathring{A} \\
 & \mathring{A} a_{234} x(t-1; c_1-1, c_2) x(t-1; c_1+1, c_2) x(t-1; c_1, c_2-1) \mathring{A} \\
 & \mathring{A} a_{2345} x(t-1; c_1-1, c_2) x(t-1; c_1+1, c_2) x(t-1; c_1, c_2-1) x(t-1; c_1, c_2+1) \mathring{A} \\
 & \mathring{A} a_{15} x(t-1; c_1, c_2) x(t-1; c_1, c_2+1) \mathring{A} a_{14} x(t-1; c_1, c_2) x(t-1; c_1, c_2-1) \mathring{A} \\
 & \mathring{A} a_{145} x(t-1; c_1, c_2) x(t-1; c_1, c_2-1) x(t-1; c_1, c_2+1) \mathring{A} a_{13} x(t-1; c_1, c_2) x(t-1; c_1+1, c_2) \mathring{A} \\
 & \mathring{A} a_{135} x(t-1; c_1, c_2) x(t-1; c_1+1, c_2) x(t-1; c_1, c_2+1) \mathring{A} \\
 & \mathring{A} a_{134} x(t-1; c_1, c_2) x(t-1; c_1+1, c_2) x(t-1; c_1, c_2-1) \mathring{A} \\
 & \mathring{A} a_{1345} x(t-1; c_1, c_2) x(t-1; c_1+1, c_2) x(t-1; c_1, c_2-1) x(t-1; c_1, c_2+1) \mathring{A} \\
 & \mathring{A} a_{12} x(t-1; c_1, c_2) x(t-1; c_1-1, c_2) \mathring{A} a_{125} x(t-1; c_1, c_2) x(t-1; c_1-1, c_2) x(t-1; c_1, c_2+1) \mathring{A} \\
 & \mathring{A} a_{124} x(t-1; c_1, c_2) x(t-1; c_1-1, c_2) x(t-1; c_1, c_2-1) \mathring{A} \\
 & \mathring{A} a_{1245} x(t-1; c_1, c_2) x(t-1; c_1-1, c_2) x(t-1; c_1, c_2-1) x(t-1; c_1, c_2+1) \mathring{A} \\
 & \mathring{A} a_{123} x(t-1; c_1, c_2) x(t-1; c_1-1, c_2) x(t-1; c_1+1, c_2) \mathring{A} \\
 & \mathring{A} a_{1235} x(t-1; c_1, c_2) x(t-1; c_1-1, c_2) x(t-1; c_1+1, c_2) x(t-1; c_1, c_2+1) \mathring{A} \\
 & \mathring{A} a_{1234} x(t-1; c_1, c_2) x(t-1; c_1-1, c_2) x(t-1; c_1+1, c_2) x(t-1; c_1, c_2-1) \mathring{A} \\
 & \mathring{A} a_{12345} x(t-1; c_1, c_2) x(t-1; c_1-1, c_2) x(t-1; c_1+1, c_2) x(t-1; c_1, c_2-1) x(t-1; c_1, c_2+1) \mathring{A} .
 \end{aligned}$$

Коэффициенты:

$a_0 = 0$	$a_2 = 0$	$a_1 = 1$	$a_{12} = 1$
$a_5 = 1$	$a_{25} = 0$	$a_{15} = 1$	$a_{125} = 0$
$a_4 = 1$	$a_{24} = 0$	$a_{14} = 1$	$a_{124} = 0$
$a_{45} = 0$	$a_{245} = 0$	$a_{145} = 0$	$a_{1245} = 0$
$a_3 = 0$	$a_{23} = 0$	$a_{13} = 1$	$a_{123} = 1$
$a_{35} = 1$	$a_{235} = 0$	$a_{135} = 1$	$a_{1235} = 1$
$a_{34} = 0$	$a_{234} = 1$	$a_{134} = 1$	$a_{1234} = 1$
$a_{345} = 0$	$a_{2345} = 1$	$a_{1345} = 0$	$a_{12345} = 0$

Многочлен Жегалкина:

$$\begin{aligned}
 x(t; c_1, c_2) = & x(t-1; c_1, c_2+1) \dot{\wedge} x(t-1; c_1, c_2-1) \dot{\wedge} x(t-1; c_1+1, c_2) x(t-1; c_1, c_2+1) \dot{\wedge} \\
 & \dot{\wedge} x(t-1; c_1-1, c_2) x(t-1; c_1+1, c_2) x(t-1; c_1, c_2-1) \dot{\wedge} \\
 & \dot{\wedge} x(t-1; c_1-1, c_2) x(t-1; c_1+1, c_2) x(t-1; c_1, c_2-1) x(t-1; c_1, c_2+1) \dot{\wedge} \\
 & \dot{\wedge} x(t-1; c_1, c_2) \dot{\wedge} x(t-1; c_1, c_2) x(t-1; c_1, c_2+1) \dot{\wedge} x(t-1; c_1, c_2) x(t-1; c_1, c_2-1) \dot{\wedge} \\
 & \dot{\wedge} x(t-1; c_1, c_2) x(t-1; c_1+1, c_2) \dot{\wedge} x(t-1; c_1, c_2) x(t-1; c_1+1, c_2) x(t-1; c_1, c_2+1) \dot{\wedge} \\
 & \dot{\wedge} x(t-1; c_1, c_2) x(t-1; c_1+1, c_2) x(t-1; c_1, c_2-1) \dot{\wedge} x(t-1; c_1, c_2) x(t-1; c_1-1, c_2) \dot{\wedge} \\
 & \dot{\wedge} x(t-1; c_1, c_2) x(t-1; c_1-1, c_2) x(t-1; c_1+1, c_2) \dot{\wedge} \\
 & \dot{\wedge} x(t-1; c_1, c_2) x(t-1; c_1-1, c_2) x(t-1; c_1+1, c_2) x(t-1; c_1, c_2+1) \dot{\wedge} \\
 & \dot{\wedge} x(t-1; c_1, c_2) x(t-1; c_1-1, c_2) x(t-1; c_1+1, c_2) x(t-1; c_1, c_2-1).
 \end{aligned}$$

0	1	0	1	0	1	0	1
000	000	000	000	001	001	001	001
0	0	1	1	0	0	1	1
0	1	1	0	0	0	1	1

0	1	0	1	0	1	0	1
100	100	100	100	101	101	101	101
0	0	1	1	0	0	1	1
0	1	1	0	0	0	0	1

0	1	0	1	0	1	0	1
010	010	010	010	011	011	011	011
0	0	1	1	0	0	1	1
0	1	1	1	0	0	1	1

0	1	0	1	0	1	0	1
110	110	110	110	111	111	111	111
0	0	1	1	0	0	1	1
0	0	0	0	0	1	1	1

Рис. 3.11. Шаблоны соседства для всех строк переходной функции

### 3.7. Использование дискретных пространственно-временных моделей при расчете себестоимости металлопродукции

Стоимость металлопродукции в значительной степени определяется стоимостью сырья и полуфабриката, а также затратами на его обработку по всем агрегатам технологической цепочки. Основные из них:

- ♦ стоимость технологического топлива (газ природный, коксовый, доменный);

- ♦ энергетические затраты (электроэнергия, пар, вода);
- ♦ вспомогательные материалы;
- ♦ отчисления в фонд оплаты труда;
- ♦ отчисления в соцстрах и фонд занятости;
- ♦ затраты на сменное оборудование;
- ♦ отчисление в ремонтный фонд;
- ♦ амортизация;
- ♦ содержание основных средств;
- ♦ расходы на внутрикомбинатовское перемещение полуфабрикатов и продукции;

и т.д.

При расчете себестоимости необходимо учитывать ряд важных факторов [29]:

1. Длина технологической цепочки, набор и последовательность прохождения агрегатов существенно зависят от типоразмеров готовой продукции и производимых марок стали;
2. Однотипные операции могут проводиться на различных агрегатах (например, обезжиривание в башенных и горизонтальных печах, вторая холодная прокатка на двадцативалковых и реверсивном станках). С другой стороны, различные технологические операции могут выполняться на одном агрегате (например, обезжиривание и обезуглероживание – в башенных печах, вторая и третья холодные прокатки – на двадцативалковом стане);
3. Необходимо учитывать расходные коэффициенты по металлу, связанные с обрезкой кромок и концевых участков полос;
4. Чтобы анализировать действительные затраты производства, необходимо учитывать различия в затратах на технологию производства каждого типоразмера, т.е. необходимо получить реальные коэффициенты трудоемкости производства различных видов металлопродукции;
5. На каждом этапе обработки необходимо учитывать незавершенное производство предыдущих календарных периодов (объем, сортимент и текущую накопленную стоимость для каждого вида продукции).

Количество тонн полуфабриката, которое требуется для производства заданного количества продукции, вычисляется по формуле:

$$V_{\Phi} = V_T * K_P, \quad (3.54)$$

где  $V_{\Phi}$  – это фактический объем загрузки;  $V_T$  – требуемый объем готовой продукции;  $K_P$  – расходный коэффициент.

Сами же расходные коэффициенты рассчитываются как произведения коэффициентов для каждого агрегата по всем маркам продукции:

$$K_P = \prod_{i=1}^n k_{ai} \quad (3.55)$$

где  $k_{ai}$  – расходный коэффициент  $i$ -го агрегата на производство определенной марки продукции;  $n$  – количество операций.

Расчетная загрузка металла по операциям представлена на рис. 3.12.

Исходные условия	Сортамент	Требуется, тн	Расчетная загрузка
	1-й вид продукции	$V_{TP1}$	$V_{TP1}K_{P1}$
	...	...	...
	$m$ -й вид продукции	$V_{TPm}$	$V_{TPm}K_{Pm}$
...			
1-я операция	коэффициент	тонн	
	$k_{11}$		$\frac{V_{TP1}K_{P1}}{k_{11}}$
	...		...
	$k_{1m}$		$\frac{V_{TPm}K_{Pm}}{k_{1m}}$
...			
$i$ -я операция	коэффициент	тонн	
	$k_{i1}$		$\frac{V_{TP1}K_{P1}}{k_{i1}}$
	...		...
	$k_{im}$		$\frac{V_{TPm}K_{Pm}}{k_{im}}$
...			
$n$ -я операция	коэффициент	тонн	
	$k_{n1}$		$\frac{V_{TP1}K_{P1}}{k_{n1}}$
	...		...
	$k_{nm}$		$\frac{V_{TPm}K_{Pm}}{k_{nm}}$

Рис. 3.12. Определение расчетной загрузки металла по операциям

Расчет себестоимости продукции происходит итеративно, от агрегата к агрегату. Себестоимость готовой продукции равна себестоимости после обработки на последнем агрегате.

Для определения себестоимости необходимо иметь расходные коэффициенты ресурсов для всех видов продукции по каждому переделу. Они определяются по методике, описанной в [30-32]. В итоге на основе теоретических представлений и с учетом технологических особенностей осуществляется расчет затрат электроэнергии, валков, пара, газа, зарплаты и т.д. для каждого агрегата в отдельности.

Процесс производства металлопродукции является длительным многоступенчатым процессом. После каждой операции получаемые полуфабрикаты (рулоны) складываются или передаются на последующие агрегаты (рис. 3.13). Так как себестоимость рассчитывается каждый месяц, то на окончание календарного периода за каждым агрегатом скапливается определенное количество полуфабриката, которое можно считать незавершенным производством. Рулоны могут храниться на складах достаточно длительное время. Туда же поступают рулоны, обработанные в другие периоды. Они могут иметь другую текущую стоимость, например из-за изменения цены ресурсов. Таким образом, необходимо учитывать разницу в затратах и корректно перенести ее в себестоимость готовой продукции.

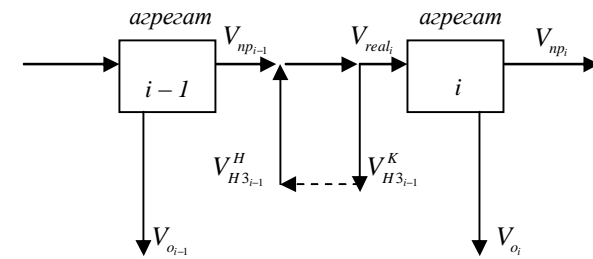


Рис. 3.13. Движение металла между агрегатами

Для этого необходимо определять новую стоимость продукции, произведенной за анализируемый календарный период с учетом незавершенного производства.

Стоимость полуфабриката:

$$C_i = \frac{S_{np_i} + S_{H3_i}}{V_{np_i} + V_{H3_i}^H} \quad (3.56)$$

где  $S_{np_i}$  – это стоимость в рублях годного полуфабриката, обработанного  $i$ -м агрегатом;  $S_{H3_i}$  – сумма в рублях незавершенного производства на начало

периода  $i$ -го агрегата;  $V_{np_i}$  – объем годного полуфабриката после  $i$ -го агрегата,  $V_{H3_i}^H$ ,  $V_{H3_i}^K$  – объем незавершенного производства после  $i$ -го агрегата (начало и конец месяца);  $V_{real_i}$  – объем полуфабриката, обработанного на  $i$ -м агрегате (за исследуемый календарный период).

При этом:

$$V_{np_{i-1}} + V_{H3_{i-1}}^H = V_{real_i} + V_{H3_{i-1}}^K ; \quad (3.57)$$

$$V_{real_i} = V_{np_i} + V_{o_i} ; \quad (3.58)$$

$$V_{np_i} = \frac{V_{np_{i-1}} + V_{H3_{i-1}}^H - V_{H3_{i-1}}^K}{k_{ai}} . \quad (3.59)$$

Объем отходов:

$$V_{o_i} = V_{real_i} (k_{ai} - 1) \quad (3.60)$$

Стоимость годного полуфабриката:

$$S_{np_i} = C_{i-1} + \frac{3_i - S_{o_i}}{V_{np_i}} , \quad (3.61)$$

где  $S_{o_i}$  – суммарная стоимость отходов;  $3_i$  – затраты на обработку полуфабрикатов на  $i$ -м агрегате.

$$3_i = \sum_{j=1}^m z_{ij} k_{z_{ij}} , \quad (3.62)$$

где  $z_{ij}$  – затраты на единицу ресурсов;  $k_{z_{ij}}$  – коэффициент трудоемкости;  $m$  – количество ресурсов.

Использование итеративных сетей позволяет учесть незавершенное производство металла от агрегата к агрегату в течение длительного времени за счет наличия временной переменной.

Величины, описанные в формулах (3.56)-(3.61), необходимо представить в терминах одномерных двунаправленных сетей.

Для каждого состояния необходимо ввести дополнительные индексы:

$i$  – номер агрегата;

$j$  – номер элемента сортамента ( $j = 1, \dots, m$ );

$t$  – календарный период.

В итоге с помощью внутренних состояний и внутренних входов и выходов можно описать основные переменные:

$$\left. \begin{aligned} x_1(i, j, t) &= V_{np}(i, j, t) \\ x_2(i, j, t) &= V_{real}(i, j, t) \\ \mathbf{x}(i, j, t) &= V_{ns}^H(i, j, t) \\ \mathbf{x}(i, j, t+1) &= V_{ns}^K(i, j, t) \\ X_1^*(i, j, t) &= S_{np}(i, j, t) \\ \mathbf{x}^*(i, j, t) &= S_{ns}(i, j, t) \end{aligned} \right\} . \quad (3.63)$$

Одномерная двунаправленная итеративная сеть, описывающая движение металла с учетом стоимости полуфабриката, представлена на рис. 3.14.

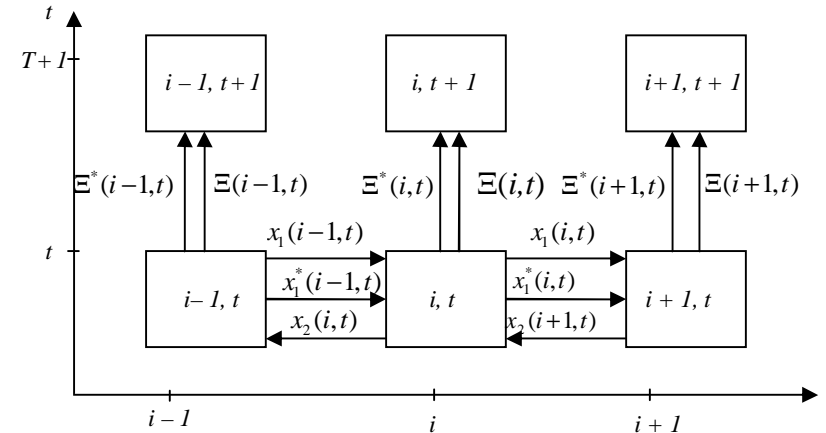


Рис. 3.14. Одномерная двунаправленная итеративная сеть движения металла с учетом стоимости полуфабриката

$X_1, X_2, X_1^*, \Xi, \Xi^*$  – векторы, учитывающие обрабатываемый сортмент.

$$\left. \begin{aligned} X_1(i-1, t) &= \{x_1(i-1, 1, t), \dots, x_1(i-1, m, t)\} \\ X_1^*(i-1, t) &= \{x_1^*(i-1, 1, t), \dots, x_1^*(i-1, m, t)\} \\ X_2(i, t) &= \{x_2(i, 1, t), \dots, x_2(i, m, t)\} \\ \Xi(i, t) &= \{\mathbf{x}(i, 1, t), \dots, \mathbf{x}(i, m, t)\} \\ \Xi^*(i, t) &= \{\mathbf{x}^*(i, 1, t), \dots, \mathbf{x}^*(i, m, t)\} \end{aligned} \right\} . \quad (3.64)$$



Управление данной сетью можно представить в виде:

$$\left. \begin{aligned} x_1(i, j, t) &= f_1[X_1(i-1, t), X_2(i+1, t), \Xi(i, t-1)] \\ x_2(i, j, t) &= f_2[X_1(i-1, t), X_2(i+1, t), \Xi(i, t-1)] \\ x(i, j, t) &= f_3[X_1(i-1, t), X_2(i+1, t), \Xi(i, t-1)] \\ x_1^*(i, j, t) &= y_1[X_1(i, t), \Xi(i, t-1), \Xi^*(i, t-1)] \\ x^*(i, j, t) &= y_2[X_1(i, t), \Xi(i, t-1), \Xi^*(i, t-1), X_1^*(i, t)] \end{aligned} \right\} \cdot \quad (3.65)$$

#### 4. Дискретные конечно-аргументные модели

Теория сигналов и систем на конечных множествах возникла в связи с задачами цифровой обработки сигналов, заданных конечным числом отсчетов. Конечное множество значений аргумента может, например, представлять множество моментов времени, в которые сняты отсчеты временного сигнала, или множества точек плоскости, в которых сняты отсчеты яркости изображения, и т.п. Эта теория в некоторых своих аспектах существенно отличается от классической, обнаруживая в то же время далеко идущие аналогии с последней. Одна из причин отличия – в том, что в случае конечного множества значений аргумента нарушаются некоторые фундаментальные для классической теории соотношения между структурой порядка и законами композиции в множестве значений аргумента. В частности, системы на конечных множествах аргументно-некаузальны.

Возникновение теории сигналов и систем на конечных множествах связано с тем, что для дискретных методов обработки сигналов, особенно при их реализации на ЭВМ, характерны ограниченная память и возможность извлечения из блока памяти отсчетов сигнала в любой последовательности. Именно в силу ограниченности памяти для таких методов является естественным представление о конечности множества значений аргумента. С возможностью извлечения из блока памяти отсчетов сигнала в любой последовательности связан фундаментальный для этой теории подход, состоящий в наделении конечного множества  $A$  значений аргумента структурой конечной группы. Основы этого подхода в отечественной литературе освещены и обоснованы в [33], где указано, что в результате такого подхода теория сигналов и систем на конечных множествах приобретает своеобразие, которое позволяет считать ее новым направлением общей теории сигналов и систем.

Групповая операция в конечном множестве значений аргумента  $A$  индуцирует сдвиг сигналов, определяемый в виде некоторой перестановки их отсчетов и не выводящий их за пределы конечного множества [34]. Это обстоятельство является главным, что отличает рассматриваемую теорию от классической дискретной теории, основанной на дискретных преобразованиях Лапласа или Фурье, в которой дискретные сигналы и системы определяются на бесконечном интервале, сдвиг понимается как параллельный перенос сигнала во времени и отсчеты сигнала в системах обработки всегда следуют друг за другом в натуральном порядке.

Конечность множества  $A$  задания сигналов и новое толкование сдвига заставляют пересмотреть многие привычные понятия и приводят к необходимости использовать новые системы базисных функций и новые виды уравнений, описывающих системы, подробно изученные в пособии [34].

При наделении конечного множества  $A$  структурой конечной абелевой группы адекватный базис в пространстве скалярных сигналов автоматически

определяется как соответствующая группа характеров. Характеры – аналоги классических экспонент – обладают свойствами ортогональности относительно естественного определяемого скалярного произведения сигналов, а также рядом других замечательных свойств, в силу чего играют в теории на конечных множествах важнейшую роль. Такие базисы являются конечными аналогами мультипликативных систем функций. Они находят применение в различных областях технической кибернетики. Их крайними частными случаями являются такие широко распространенные системы функций, как функции Уолша, основанные на двоичной арифметике (когда конечное множество содержит  $2^d$  элементов при некотором натуральном  $d$  и наделяется структурой диадической группы) и дискретные экспоненциальные функции, основанные на  $N$ -ичной арифметике (когда конечное множество содержит  $N$  элементов и наделяется структурой циклической группы). Широкое многообразие мультипликативных базисов существенно расширяет возможности теории сигналов и систем на конечных множествах. Появляется возможность производить спектральные преобразования точно, так как они выражаются конечными суммами. Спектральные преобразования систематически используются и при исследовании конечно-аргументных систем [34].

Новые виды уравнений, описывающих системы на конечных множествах, возникают как аналоги классических, с заменой традиционных понятий сверки, сдвига, дифференцирования соответствующими понятиями, отвечающими наделению конечного множества структурой конечной группы. Так определяются подробно рассматриваемые [34] сверточные и десверточные системы,  $\{m\}$ -цепи, гармонические дифференциальные системы и их частные случаи – диадические и циклические дифференциальные системы. Даже в том случае, когда параметры этих систем постоянны, они не являются стационарными в традиционном понимании, но стационарны в смысле нового понятия сдвига сигналов, определяемого групповой операцией в конечном множестве. Так трактуется аргументная стационарность этих систем. Для систем на конечных множествах большим своеобразием обладает трактовка таких фундаментальных понятий теории систем, как динамика, каузальность (причинность), рекурсивность. Своеобразие связано с тем, что групповая операция в конечном множестве, сводящаяся к перестановке его элементов, не согласована с отношениями линейного или частного порядка, которыми может быть наделено это множество. Возникают трудности и при реализации систем на конечных множествах. Классическая проблема реализации состоит в восстановлении уравнений состояния системы по ее описанию вход-выход; ее решение на разных уровнях абстрактности известно для многих классов каузальных систем, причем в случае стационарных систем описание вход-выход стандартно задается в виде сверки, а уравнения состояний – в виде дифференциальных или разностных уравнений. В случае систем на конечных группах принятие сверки на группе в качестве стандартного

описания вход-выход представляется вполне естественным; между тем, по-видимому, не существует до сих пор общепринятого мнения о том, что считать стандартной формой уравнений состояния систем на конечных множествах. В [34] в качестве основных претендентов на эту роль представлены  $K$ -дифференциальные и  $K$ -разностные системы.

#### 4.1. Одномерные циклические линейные итеративные цепи ( $N$ -цепи)

Каждая из одномерных циклических линейных итеративных цепей представляет собой цепочку клеток без памяти, но не бесконечную вправо или в обе стороны, как одномерные однонаправленные или двунаправленные линейные итеративные цепи, а замкнутую (рис. 4.1), называемую также кольцевой структурой.

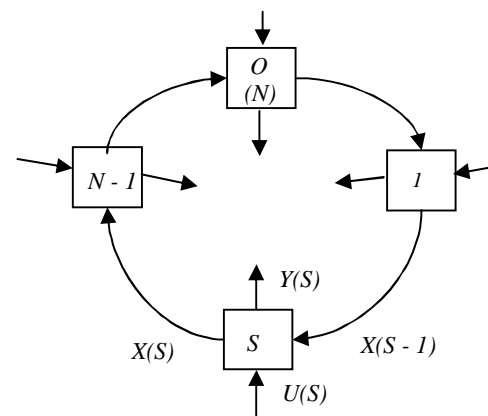


Рис. 4.1. Одномерная однонаправленная циклическая линейная итеративная сеть

Клеткам присвоены номера  $s \in Z_0(N) = \{0, 1, \dots, N-1\}$ , каждая имеет внешний вход  $u(s) \in K^l$  и выход  $y(s) \in K^q$  и внутренние (по отношению к цепи как к целому) вход  $x(s-1) \in K^n$  (вход предыдущей клетки) и выход  $x(s) \in K^n$  (выход последующей клетки): при этом фиксировано, например, направление нумерации по часовой стрелке (альтернативное направление против часовой стрелки). Так как клетка, имеющая номер 0, отождествляется с клеткой номер  $N$ , то естественно обычные арифметические операции сложения и вычитания (+ и -) заменить операциями сложения и вычитания по

модулю  $N$ , обозначаемыми через  $\overset{\bullet}{+}_N$  и  $\overset{\bullet}{-}_N$  соответственно [34]. Уравнения такой  $N$ -цепи (по терминологии [35]) запишутся в виде:

$$\begin{aligned} x(s) &= Fx(\overset{\bullet}{s-\overset{\bullet}{1}}_N) + Yu(s), \\ y(s) &= Hx(\overset{\bullet}{s-\overset{\bullet}{1}}_N) + Gu(s), \quad s \in Z_0(N). \end{aligned} \quad (4.1)$$

Такие цепи являются конечно-аргументными аналогами линейных стационарных дискретно-временных моделей (2.2) и линейных однородных одномерных однонаправленных дискретно-пространственных моделей (2.7). В операторной записи оператор  $x$  временного сдвига или оператор  $Z$  пространственного сдвига заменяются оператором  $mN$ -сдвига. Отметим существенные отличия. Хотя матрицы в уравнениях (4.1) постоянны,  $N$ -цепь не является время-стационарной или пространственно однородной в принятом смысле этих понятий: эта модель оказывается инвариантной относительно  $N$ -сдвига, или  $N$ -стационарной [33]; так трактуется ее аргументная стационарность.  $N$ -цепь аргументно-некаузальна, что сближает ее с двунаправленной цепью (2.9). В то же время можно указать конечно-аргументный аналог последней, представленный на рис. 4.2 и описываемый уравнениями:

$$\begin{aligned} x(s) &= F_1x(\overset{\bullet}{s-\overset{\bullet}{1}}_N) + F_2x(\overset{\bullet}{s+\overset{\bullet}{1}}_N) + Yu(s), \\ y(s) &= H_1x(\overset{\bullet}{s-\overset{\bullet}{1}}_N) + H_2x(\overset{\bullet}{s+\overset{\bullet}{1}}_N) + Gu(s). \end{aligned} \quad (4.2)$$

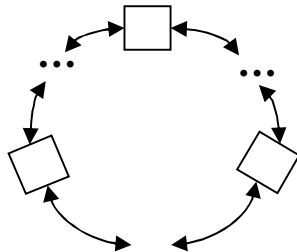


Рис. 4.2. Одномерная двунаправленная циклическая линейная итеративная цепь

## 4.2. Одномерные полициклические линейные итеративные цепи ( $m$ -цепи)

Если число  $N$  не является простым, то в соответствии с основной теоремой арифметики оно представляется, причем многими различными способами, в виде произведения (не обязательно простых и различных) сомножителей  $N = m_1 \cdot \dots \cdot m_d$ , совокупность которых обозначим через  $\{m\}$ : этим определены [34,35] операции  $\{m\}$ -сдвига  $\overset{\bullet}{+}_{\{m\}}$  и  $\overset{\bullet}{-}_{\{m\}}$  (коротко просто  $\overset{\bullet}{+}$  и  $\overset{\bullet}{-}$ ) в множестве  $Z_0(N) = Z_0(m_1, \dots, m_d)$ , которые, как и операции  $N$ -сдвига, не выходят за множество  $Z_0(N)$ : в частном случае  $N = m^d$ ; такие операции  $m$ -сдвига в  $Z_0(m_1, \dots, m_d)$  обозначаются  $\overset{\bullet}{+}_{\{m\}}$  и  $\overset{\bullet}{-}_{\{m\}}$  [33,34]. Заменяя обычный арифметический сдвиг  $\{m\}$ -сдвигом, получим уравнения  $\{m\}$ -цепи [35]:

$$\begin{aligned} x(s) &= Fx(\overset{\bullet}{s-\overset{\bullet}{1}}) + Yu(s) \\ y(s) &= Hx(s) + Gu(s). \end{aligned} \quad (4.3)$$

Клеточное представление такой структуры в случае  $N = 12 = 3 \cdot 4$  предложено на рис. 4.3.

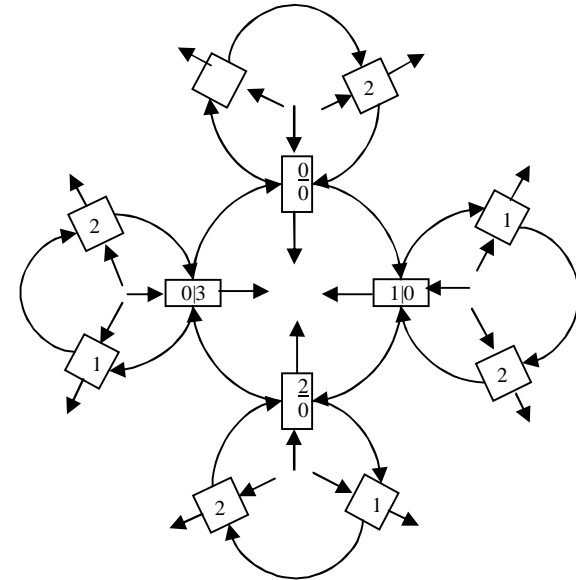


Рис. 4.3. Полициклическая  $\{4,3\}$ -цепь

Относительно  $\{m\}$ -цепей справедливы замечания, сделанные об  $N$ -цепях, являющихся их частным случаем: аргументная некаузальность близка к предложенной в [33] трактовке  $m$ -цепей как нерекурсивных. В [36] подробно обсуждаются возможности использования  $m$ -цепей как процессоров для цифровой обработки конечных массивов информации.

### 4.3. Двумерные полициклические линейные итеративные цепи (двумерные ( $m$ )-цепи)

Трактовка систем на конечных множествах как процессоров для обработки массивов информации позволяет ввести аналог  $\{m\}$ -цепей (предназначенных для обработки массивов с формально одномерным носителем) для случая массивов с двумерным носителем. Пусть двумерный массив информации задан на квадрате  $Z_0^2(m^d) = \{(s_1, s_2), s_1, s_2 \in Z_0(m^d)\}$  (это – непосредственное обобщение сигнала, определенного на конечном интервале). Соответствующий процессор можно моделировать двумерной  $m$ -цепью, которая, сочетая (2.14) и (4.1), определяется уравнениями:

$$x_1(s_1, s_2) = F_{11}x_1(s_1 - I, s_2) + F_{12}x_2(s_1, s_2 - I) + Y_1u(s_1, s_2), \quad (4.4)$$

$$x_2(s_1, s_2) = F_{21}x_1(s_1 - I, s_2) + F_{22}x_2(s_1, s_2 - I) + Y_2u(s_1, s_2).$$

Этот класс цепей включает цепи, описываемые уравнениями:

$$x(s_1, s_2) = F_1x(s_1 - I, s_2) + F_2x(s_1, s_2 - I) + F_3x(s_1 - I, s_2 - I) + Yu(s_1, s_2). \quad (4.5)$$

Для упрощения рассмотрений ограничимся случаем  $n_1 = n_2 = l = 1$ , так что  $n = 2$  (вход и составляющая выхода цепи скалярны). В этом случае уравнения (4.4) запишутся в виде:

$$x_1(s_1, s_2) = j_{11}x_1(s_1 - I, s_2) + j_{12}x_2(s_1, s_2 - I) + Y_1u(s_1, s_2), \quad (4.6)$$

$$x_2(s_1, s_2) = j_{21}x_1(s_1 - I, s_2) + j_{22}x_2(s_1, s_2 - I) + Y_2u(s_1, s_2),$$

где  $j_{11}, \dots, Y_2$  скаляры.

Полагая

$$x(s_1, s_2) = \begin{bmatrix} x_1(s_1, s_2) \\ x_2(s_1, s_2) \end{bmatrix}, \quad qx(s_1, s_2) = \begin{bmatrix} x_1(s_1 - I, s_2) \\ x_2(s_1, s_2 - I) \end{bmatrix}, \quad (4.7)$$

$$F = \begin{bmatrix} j_{11} & j_{12} \\ j_{21} & j_{22} \end{bmatrix}, \quad Y = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix},$$

запишем (4.6) в виде одного уравнения:

$$x(s_1, s_2) = Fqx(s_1, s_2) + Yu(s_1, s_2), \quad (s_1, s_2) \in Z_0^2(m^d). \quad (4.8)$$

Установим связь такой двумерной  $m$ -цепи с одномерной (4.1). Поскольку  $m$ -сдвиг не выводит за интервал  $Z_0(m^d)$  для любой точки  $(s_1, s_2) \in Z_0^2(m^d)$ ,  $s_1 + s_2 = K \in Z_0(m^d)$ . Определим «классы точек квадрата»:

$$G(k) = \{(s_1, s_2) \in Z_0^2(m^d) : s_1 + s_2 = K\}. \quad (4.9)$$

Каждый класс содержит  $m^d$  точек и

$$G(k) \cap G(l) = \emptyset, k \neq l, \quad \bigcup_{K \in Z_0(m^d)} G(k) = Z_0^2(m^d). \quad (4.10)$$

Предположим, что точки каким-либо образом упорядочены внутри класса, и обозначим:

$$C(k) = \left( x^T(s_1, s_2) \right)_{(s_1, s_2) \in G(k)}^T \in C^{2m^d}, \quad (4.11)$$

$$U(k) = \left( u^T(s_1, s_2) \right)_{(s_1, s_2) \in G(k)}^T \in C^{m^d}.$$

Если  $(s_1, s_2) \in G(k)$ , т.е.  $s_1 + s_2 = k$ , то

$$(s_1 - I) + s_2 = s_1 + (s_2 - I) = (s_1 + s_2) - I = k - I, \quad (4.12)$$

так что  $(s_1 - I, s_2), (s_1, s_2 - I) \in G(k - I)$  поэтому из уравнений (4.8) непосредственно следует, что  $X(k)$  линейно выражается через  $C(k - I)$  (а также через  $U(k)$ ). Эту связь можно описать уравнением:

$$X(k) = \underline{F}x(k - I) + \underline{Y}U(k), \quad k \in Z_0(m^d), \quad (4.13)$$

где структура матрицы  $\underline{Y}$  определяется соотношением:

$$\underline{Y} = I_{m^d} \otimes Y, \quad (4.14)$$

так что матрица  $\underline{Y}$  одна и та же для все  $k \in Z_0(m^d)$ , здесь  $\otimes$  – кронекеровское или тензорное произведение матриц. Всегда можно так упорядочить точки внутри классов и так согласовать эти упорядочивания между классами, что матрица  $\underline{F}$  даже не будет зависеть от  $k \in Z_0(m^d)$  и ее структура определяется соотношением:

$$\underline{F} = I_{m^{d-1}} \otimes (W_m \times F). \quad (4.15)$$

Здесь  $W_m$  – блочная матрица блочного порядка  $m$  (одна и та же для всех  $k \in Z_0(m^d)$ ).

#### 4.4. Одномерные полициклические линейные итеративные сети (решетчатые модели)

Каждая из решетчатых моделей представляет собой замкнутую цепочку клеток с памятью, пример которой представлен на рис. 4.4.

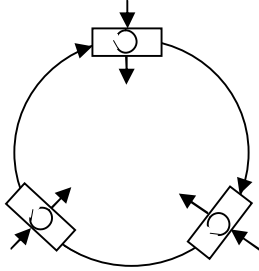


Рис. 4.4. Решетчатая система как одномерная циклическая сеть

Подобно тому, как в подглаве 3.1, сигналы такой сети зависят от номера клетки  $s \in Z_0(N)$ , так и от времени  $t \in Z_0$ . Такие модели под названием «решетчатые системы» рассмотрены в [37], где предложено описание их динамики

$$\begin{aligned} x(t+1, s) &= \sum_{s \in Z_0(N)} j(s) x(t, s-s) + \sum_{q \in Z_0(N)} Y(q) u(t, s-q) = \\ &= \sum_{s \in Z_0(N)} j(s-s) x(t, s) + \sum_{q \in Z_0(N)} Y(s-q) u(t, q) = \quad (4.16) \\ &= (F \otimes_s x)(t, s) + (Y \otimes_s u)(t, s). \end{aligned}$$

С формальной точки зрения описание (4.16) является более общим, чем например (2.14), так как его слагаемые представляют собой  $K$ -свертки [20]: некоторые важные свойства при этом сохраняются, например, линейность и  $\{m\}$ -стационарность. С точки зрения содержательной интерпретации такие модели можно трактовать (заменяя время  $t$  на другую пространственную переменную  $p$ , отличную от  $s$ ) и как бесконечные вправо цепочки из клеток, каждая из которых представляет собой  $\{m\}$ -цепь: пример такой трактовки представлен на рис. 4.5.

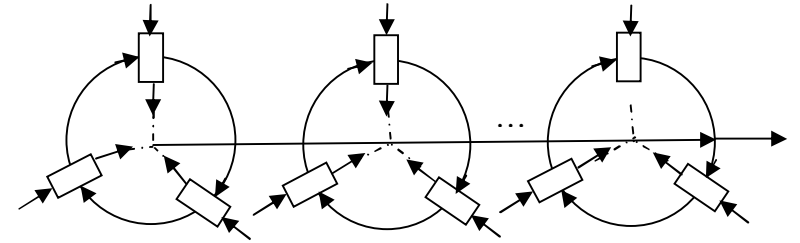


Рис. 4.5. Решетчатая система как одномерная однонаправленная цепь, составленная из циклических цепей

Различные трактовки расширяют использование таких структур при моделировании реальных объектов. Таким образом, решетчатые модели сочетают классические дискретно-временные модели и конечно-аргументные модели. В [37] рассмотрено сведение решетчатых моделей на многомерных решетках к моделям на одномерных решетках и наоборот: эти преобразования аналогичны переходу к ассоциированным моделям от многомерностных, а также приведение двумерных  $\{m\}$ -цепей к одномерным, рассмотренных в подглаве 3.3.

## 5. Сети Петри

### 5.1. Описание сетей Петри

Сети Петри (СП) являются удобным аппаратом моделирования параллельных процессов, т.е. процессов, протекающих в системе независимо один от другого. На выполнение таких процессов не накладываются какие-либо условия синхронизации. Моменты начала и завершения параллельных процессов, интервалы их реализации не являются в системе взаимообусловленными. Параллельным процессам соответствуют состояния сетей Петри, в которых разрешены сразу несколько переходов, и каждый из них может сработать. Но какой именно переход сработает, решается всякий раз случайным образом по правилам равновероятного выбора. Сети Петри широко используются при моделировании и управлении гибкими производственными системами.

В отличие от конечных автоматов, в терминах которых описываются глобальные изменения состояния системы, сеть Петри конкретизирует внимание на локальных событиях (им соответствуют переходы), на локальных условиях (им соответствуют позиции), на локальных связях между событиями и условиями. Поэтому сеть Петри более адекватно, чем конечные автоматы, моделируют поведение распределенных асинхронных систем.

Тем самым СП приспособлены для моделирования параллельных вычислительных процессов и систем [43].

Для полного описания функционирования СП задаются:

1). Множество позиций  $P = \{P_1, \dots, P_n\}$ , т.е. конечное число позиций ( $|P| = n$ );

2). Множество переходов  $T = \{t_1, \dots, t_m\}$ , т.е. конечное число переходов ( $|T| = m$ );

Позиции и переходы не пересекаются, и совпадать не могут:  
 $P \cap T = \emptyset$ ;

3). Начальная маркировка или разметка сети:

$M_0 = (M_0(P_1), \dots, (M_0(P_n))), \in Z_0^n$ , где  $n$  – вектор, компоненты целые неотрицательные числа ( $Z_0$ ).

Координаты вектора указывают количество меток или фишек в каждой позиции СП в начальный момент времени.

В результате функционирования СП в каждый момент времени  $k$  возникает текущая разметка,  $M_k = (M_k(P_1), \dots, M_k(P_n))$ ,  $M_k(P_i) \in Z_0$ , определяющая состояние сети в этот момент времени ( $M_0$  – начальное состояние);

4). Взаимосвязь позиций и переходов задаются компонентами:

$I(t_j)$  – позиции на входах перехода  $t_j$ ;

$O(t_j)$  – позиции на выходах перехода  $t_j$ ;

$I(p_i)$  – переходы на входах позиции  $p_i$ ;

$O(p_i)$  – переходы на выходах позиции  $p_i$ ;

5). Функция переходов  $j$ , определяющая  $M^j = j(C; M)$ , – возможное состояние сети при начальной маркировке  $M_0$ , следующее за состоянием  $M$ , но не обязательно непосредственно за  $M$ .

Функция  $j$  определяется правилами функционирования:

- а) в каждый момент времени срабатывает только разрешенный переход, т.е. такой, во всех входных позициях которого при заданной маркировке  $M$  имеются ненулевые метки;
- б) срабатывание перехода может наступить через любой конечный промежуток времени после его разрешения (т.е. переход срабатывает не обязательно сразу непосредственно после его разрешения);
- в) если в некотором состоянии разрешены несколько переходов, то всегда срабатывает только какой-то один из них; в этом правиле заключается недетерминизм функционирования сети Петри (т.е. если в какой-то момент времени несколько переходов могут сработать, то неизвестно, какой именно сработает);
- г) в результате срабатывания перехода метки во всех его входных позициях уменьшаются на единицу, а в выходных – на единицу;
- д) срабатывание перехода – неделимый акт; считается, что изменение количества во входных и выходных позициях при срабатывании осуществляется мгновенно.

При графическом изображении сети Петри вершинами графа служат позиции и переходы, образующие две доли  $P$  и  $T$ . Дуги отражают связь позиций и переходов. Дуга  $(p_i, t_j)$  направлена от позиции  $p_i$  к переходу  $t_j$ , аналогично  $(t_j, p_i)$ . Дуги обозначаются  $d$ .

Позиции обозначают окружностями, а переходы – планками. Ниже приведен пример сети Петри на рис. 5.1, 5.2.

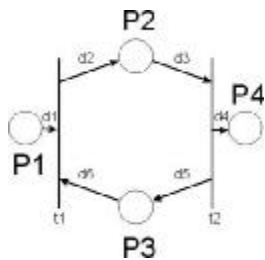


Рис. 5.1. Графическое представление сети Петри

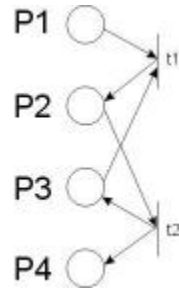


Рис. 5.2. Представление сети Петри в виде двудольного графа

Для данной сети Петри  $|P| = n = 4$ ,  $|T| = m = 2$ .

$$\begin{aligned} I(t_1) &= \{p_1, p_3\} & I(t_2) &= \{p_2\} \\ O(t_1) &= \{p_2\} & O(t_2) &= \{p_3, p_4\} \\ I(p_1) &= \emptyset & I(p_2) &= \{t_1\} & I(p_3) &= \{t_2\} & I(p_4) &= \{t_2\} \\ O(p_1) &= \{t_1\} & O(p_2) &= \{t_2\} & O(p_3) &= \{t_1\} & O(p_4) &= \emptyset \end{aligned}$$

Матрица сети Петри размера  $m \times n$ :

$$R = R^+ - R^-, \quad (5.1)$$

где

$$R^+ = [r_{ji}^+], \quad r_{ji}^+ = \begin{cases} 1, & \text{если } p_i \in O(t_j), \\ 0, & \text{в любом другом случае} \end{cases}, \quad (5.2)$$

$$R^- = [r_{ji}^-], \quad r_{ji}^- = \begin{cases} 1, & \text{если } p_i \in I(t_j), \\ 0, & \text{в любом другом случае} \end{cases}. \quad (5.3)$$

Отсюда

$$R^+ = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \quad R^- = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad R = \begin{bmatrix} -1 & 1 & -1 & 0 \\ 0 & -1 & 1 & 1 \end{bmatrix}.$$

Вершинами этого графа служат позиции и переходы, образующие две доли:  $P$  и  $T$ .

Дуги отражают связь позиций и переходов.

Дуга  $(t_j, p_i)$  направлена от позиции  $p_i$  к переходу  $t_j$  (дуга от позиции к позиции не идет, и от перехода к переходу), аналогично  $(t_j, p_j)$ .

Позиции обозначают кружками, переходы – планками.

Граф – наиболее наглядное изображение СП.

В первоначальном изображении отображается и начальная разметка (чтобы двигаться – на входе перехода все позиции должны быть заняты).

Правила функционирования могут быть частично отражены с помощью матрицы СП. Пусть  $\mathbf{m}(j)$  –  $m$ -мерный вектор из нулей с единицей на  $j$ -м месте,  $j = 1, \dots, m$  (показывает, какой переход срабатывает). Этот вектор используется как отражение правила (б). Тогда правило (а) записывается в виде:

$$M \geq \mathbf{m}(j) R^-. \quad (5.4)$$

Условие разрешения перехода  $t_j$  (используется  $R^-$ , т.к. речь о входных позициях, а  $R^-$  – оговаривает все входные позиции).

Правило (г) записывается в виде:

$$M_j = M_0 + \mathbf{m}(j) R = M_0 - \underline{\mathbf{m}(j) R^-} + \underline{\mathbf{m}(j) R^+}. \quad (5.5)$$

Для любого момента времени связь разметок выражается соотношением:

$$M_k = M_{k-1} + \mathbf{m}(j) R \quad (5.6)$$

Разметка  $M_k$  оказывается непосредственно достижимой из разметки  $M_{k-1}$  (но не обязательно непосредственно следует). Последовательно применяя формулу (5.6), получим:

$$M_k = M_0 + \mathbf{m} R, \quad (5.7)$$

где

$$\mathbf{m} = \mathbf{m}(j_1) + \mathbf{m}(j_2) + \dots + \mathbf{m}(j_k) \quad (5.8)$$

есть  $m$ -мерный вектор, координатами которого могут быть уже любые неотрицательные целые числа ( $n$  – количество шагов).

Разметка  $M_k$  – достижима из разметки  $M_0$ .

Функционирование:

$$M_1 = M_0 + \mathbf{m}(j) R.$$

Рассмотрим маркировки:  $M_0' = [0 \ 0 \ 0 \ 0]$  и  $M_0'' = [1 \ 1 \ 1 \ 0]$ .

Необходимо определить разрешенность данных маркировок:

$$\mathbf{m}(1) R^- = [1 \ 0] \cdot \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} = [1 \ 0 \ 1 \ 0];$$

т.к.  $M_0' = [0 \ 0 \ 0 \ 0]$ , то условие  $M_0' \geq \mathbf{m}(1) R^-$  не выполняется;

$$\mathbf{m}(2) = [0 \ 1] \cdot \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} = [0 \ 1 \ 0 \ 0] \leq M_0'' \text{ не выполняется, следовательно}$$

но, такая сеть функционировать не будет и маркировка  $M_0'$  не разрешена.

В то же время  $M_0'' = |1 \ 1 \ 1 \ 0\rangle > |1 \ 0 \ 1 \ 0\rangle = m(1)R^-$  – переход разрешен; маркировка разрешена. Функционирование сети с данной маркировкой на один шаг:

$$M_i^{(1)} = M_0 + m(1)R = [1 \ 1 \ 1 \ 0] + [1 \ 0] \cdot \begin{bmatrix} -1 & 0 & -1 & 0 \\ 0 & -1 & 1 & 1 \end{bmatrix} = [0 \ 2 \ 0 \ 0];$$

$$M_i^{(2)} = M_0 + m(2)R = [1 \ 1 \ 1 \ 0] + [0 \ 1] \cdot \begin{bmatrix} -1 & 1 & -1 & 0 \\ 0 & -1 & 1 & 1 \end{bmatrix} = [1 \ 0 \ 2 \ 1],$$

следовательно, сработать может любой переход. Дерево маркировок представлено на рис. 5.3.

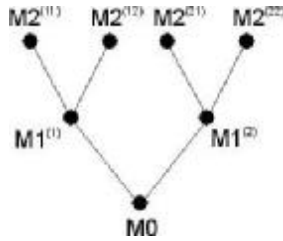


Рис. 5.3. Дерево маркировок после первого шага

Следующий этап – исследование достижимости некоторой маркировки  $M''$ . Задача достижимости сводится к определению неотрицательного целочисленного вектора

$$m = m(j_1) + m(j_2) + m(j_3) + \dots + m(j_k),$$

где  $k$  – количество шагов.

Необходимо решить уравнение:

$$mR = M'' - M. \quad (5.9)$$

Пусть  $M_0 = [0 \ 1 \ 0 \ 0]$ ,  $M'' = [1 \ 1 \ 1 \ 0]$ ;

$$[m_1, m_2] \cdot \begin{bmatrix} -1 & 1 & -1 & 0 \\ 0 & -1 & 1 & 1 \end{bmatrix} = [1 \ 1 \ 1 \ 0] - [0 \ 1 \ 0 \ 0] = [1 \ 0 \ 1 \ 0];$$

$$\begin{cases} -m_1 = 1, \\ m_1 - m_2 = 0, \\ -m_1 + m_2 = 1, \\ m_2 = 0. \end{cases}$$

Имеем систему четырех уравнений с двумя неизвестными;  $m$  не может равняться  $-1$  или  $0,5$ ; следовательно, маркировка  $M''$  не достижима.

Если  $M'' = [0 \ 0 \ 1 \ 1]$ , то

$$[m_1, m_2] \cdot \begin{bmatrix} -1 & 1 & -1 & 0 \\ 0 & -1 & 1 & 1 \end{bmatrix} = [0 \ 0 \ 1 \ 1] - [0 \ 1 \ 0 \ 0] = [0 \ -1 \ 1 \ 1],$$

$$\begin{cases} -m_1 = 1, \\ m_1 - m_2 = -1, \\ -m_1 + m_2 = 1, \\ m_2 = 1. \end{cases}$$

Следовательно,  $m_2 = 1$ ,  $m_1 = 0$ ,  $m = m(2) = [0 \ 1]$ . Это означает, что разрешен второй переход и метка  $M''$  достижима.

### Проблема сохранения

Сеть Петри с разметкой  $M$  называется сохраняющей относительно везового вектора  $W = (w_1, \dots, w_n) \in Z_0^n$  (число координат по числу позиций у этого вектора), если для любой достижимой разметки  $\tilde{M}$  выполняется соотношение

$$\sum_{i=1}^n w_i M(p_i) = \sum_{i=1}^n w_i \tilde{M}(p_i) \quad (5.10)$$

(для всех последующих разметок весовая функция не должна изменяться).

Краткая запись:

$$MW = \tilde{M}W, \quad (5.11)$$

т.к.  $\tilde{M} = M + \mu R$ , то умножив скалярно все члены этого выражения на  $W$ , получим:

$$\tilde{M}W = MW + \mu RW. \quad (5.12)$$

Но по условию (5.10) следует:

$$\mu RW = \emptyset. \quad (5.13)$$

Т.к. это должно выполняться при любом  $\mu$ , то проверка сохранности и отыскания сохраняющих весов полностью решается уравнением

$$RW = \emptyset. \quad (5.14)$$

В случае  $W = (1, 1, 1, \dots, 1)$  (все веса единичны), СП называется строго сохраняющей. Она характеризуется тем, что для любого перехода  $t_j$  число выходящих и входных позиций одинаково:

$$|O(t_j)| = |I(t_j)| \text{ для всех } t_j. \quad (5.15)$$



С помощью матричного представления СП рассматриваются и другие проблемы (проблема живости (живая СП), проблемы тупиков и ограниченности, и т.д.) [38-43].

## 5.2. Использование сетей Петри при расчете себестоимости продуктов металлургического производства

При расчете себестоимости продуктов металлургического производства представляет интерес проблема выбора соответствующего математического аппарата, позволяющего упростить процесс расчета и представить его более наглядными (в том числе и графическими) средствами. Процесс расчета себестоимости может быть представлен в виде функционирования сети Петри. В такое представление входят следующие элементы.

1. Непустое конечное множество позиций  $P = \{P_1, P_2, \dots, P_n\}$ . Каждый элемент этого множества соответствует одному материалу, занятому в металлургическом производстве. Для каждого материала в любой текущий дискретный момент времени  $k = 0, 1, 2, \dots$  себестоимость либо известна (задана изначально или рассчитана в один из предыдущих моментов времени), либо не известна (расчет для этого материала еще не производился). Материал может быть либо сырьем (его себестоимость, т.е. в данном случае покупная цена, известна заранее), либо продуктом (включает в себя несколько материалов вида «сырье» и может включать один или несколько материалов вида «продукт»); себестоимость продукта рассчитывается в один из моментов времени функционирования сети Петри). Себестоимость продуктов металлургического производства складывается из себестоимости сырья, расходов по переделу и вспомогательных материалов, таких как сжатый воздух, пар, техническая вода, электроэнергия и т.п. Поэтому нахождение искомой себестоимости  $c_i$  каждого из продуктов сводится к вычислению суммы произведений [44]:

$$c_i = \sum_{j=1}^n c_j a_{ij}, \quad i = 1, 2, \dots, m, \quad (5.16)$$

где  $c_j$ , и  $n$  – себестоимость  $j$ -го материала (продукта либо сырья) и количество материалов, используемых при производстве  $i$ -го продукта;  $a_{ij}$  – коэффициент, характеризующий долю объема  $j$ -го материала в общем объеме производства  $i$ -го продукта;  $m$  – общее количество продуктов.

2. Непустое конечное множество переходов  $T = \{t_1, t_2, \dots, t_m\}$ . Каждый элемент этого множества соответствует моменту расчета себестоимости одного из элементов вида «продукт» из множества позиций  $P$ , поэтому одно-

значно закреплен за этим элементом. Множества позиций и переходов не пересекаются.

3. Вектор текущей маркировки, возникающей в процессе функционирования сети Петри в каждый момент времени  $k$  и определяющей состояние производимого расчета в этот момент времени:

$$m_k = [m_k(P_1)m_k(P_2)\dots m_k(P_n)]^T, \quad (5.17)$$

где

$$m_k(P_i) = \begin{cases} 0, & \text{если себестоимость материала } P_i, \\ & i = 1, \dots, n, \text{ ещё не рассчитана;} \\ I, & \text{если себестоимость материала } P_i, \\ & i = 1, \dots, n, \text{ рассчитана или известна первоначально.} \end{cases}$$

Рассматриваемая сеть Петри является безопасной, поскольку ни один из элементов вектора  $m_k$  текущей маркировки не может превысить единицу и допускает реализацию в виде триггера.

4. Вектор начальной маркировки в момент времени  $k = 0$ :

$$m_0 = [m_0(P_1)m_0(P_2)\dots m_0(P_n)]^T, \quad (5.18)$$

где

$$m_0(P_i) = \begin{cases} 0, & \text{если материал } P_i, i = 1, \dots, n, \\ & \text{является продуктом;} \\ 1, & \text{если материал } P_i, i = 1, \dots, n, \\ & \text{является сырьем.} \end{cases}$$

Структура расчета себестоимости может быть представлена в виде двудольного ориентированного графа. На рис. 5.4 показан фрагмент структуры расчета себестоимости продукции Новолипецкого металлургического комбината (НЛМК) на примере аглопроизводства.

Для удобства восприятия из материалов вида «сырье» взяты только те, которые используются при производстве агломерата (позиция  $P_{26}$ ). Материалы вида «сырье», используемые при производстве промежуточных продуктов  $P_1, \dots, P_9$ , опущены. Вершинами графа являются позиции и переходы, т.е. материалы и моменты расчета себестоимости, которые образуют две доли:  $P$  и  $T$ . Дуги отражают связь позиций и переходов. Позиции изображены окружностями, переходы – отрезками. Взаимосвязь позиций и переходов задается множествами:  $1(t_j)$  – для позиций на входах перехода  $t_j$ ;  $0(t_j)$  – для позиций на выходах перехода  $t_j$ .

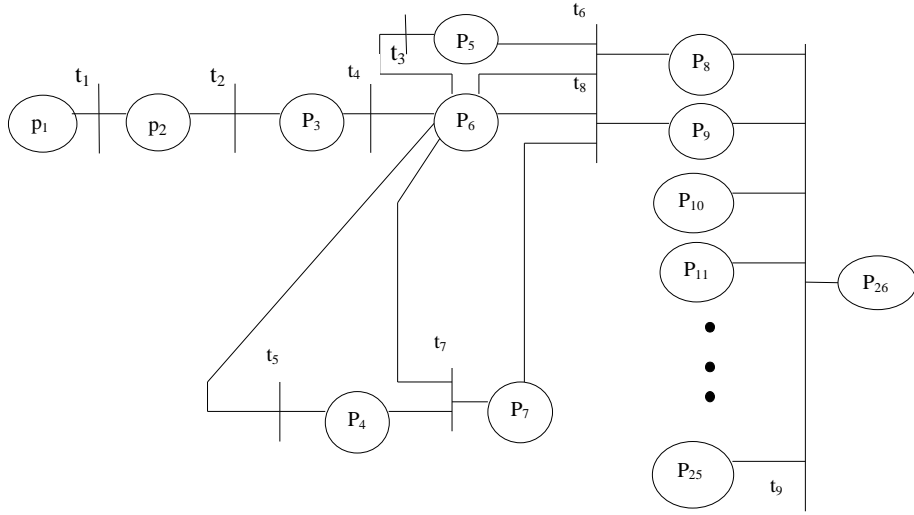


Рис. 5.4. Фрагмент структуры расчета себестоимости продукции ОАО «НЛМК» в виде графа на примере аглопроизводства

Для описания функционирования сети Петри применяется ее матричное представление. В матрицах  $R^+ = [r_{ij}^+]$ ,  $R^- = [r_{ij}^-]$  со строками  $n$  и столбцами от строки соответствуют позициям (материалам), а столбцы – переходам (моментам расчета):

$$r_{ij}^+ = \begin{cases} 1, & \text{если } P_i - \text{элемент множества } 0(t_j); \\ 0 & \text{в противном случае.} \end{cases} \quad (5.19)$$

$$r_{ij}^- = \begin{cases} 1, & \text{если } P_i - \text{элемент множества } 1(t_j); \\ 0 & \text{в противном случае.} \end{cases}$$

Функционирование данной сети Петри осуществляется в соответствии с основными правилами.

**Правило 1.** В каждый момент времени срабатывает только разрешенный переход  $t_j$ , т.е. такой, для которого все элементы вектора текущей маркировки  $m_k$ , соответствующие позициям из множества  $1(t_j)$ , равны единице. Таким образом, в текущий момент времени можно определить себестоимость только тех продуктов, для которых уже известна себестоимость всех используемых при их производстве материалов.

**Правило 2.** Если в некоторый момент времени разрешены несколько переходов, то срабатывает только какой-то один из них. Иначе, говоря, если есть информация для расчета себестоимости сразу нескольких продуктов, то на конечный результат расчета не влияет результат выбора какого-либо одного из них.

**Правило 3.** В результате срабатывания  $j$ -го перехода все равные нулю элементы вектора текущей маркировки  $m_k$ , соответствующие позициям из множества  $0(t_j)$ , становятся равными единице, что отвечает появлению информации об окончании расчета себестоимости для продукта для этого перехода.

Пусть  $B_k(j)$  – вектор размерности  $m$  из нулей с единицей на  $j$ -м месте ( $j = 1, 2, \dots, m$ ). Этот вектор используется как отражение правила 2. Тогда правило 1 запишется в следующем виде: если

$$m_k = R^- B_k(j) \quad (5.20)$$

(сравнение векторов производится поэлементно), то переход  $t_j$  разрешен, т.е. в текущий момент времени  $k$  можно рассчитать себестоимость продукта, за которым закреплен переход  $t_j$ .

Правило 3 для любого момента времени  $k > 0$  выглядит следующим образом:

$$m_k = m_{k-1} + R^+ B_k(j), \quad (5.21)$$

где  $B_k(j)$  удовлетворяет соотношению (5.20).

Последовательно применяя соотношение (5.21), получим

$$m_k = m_0 + R^+ B, \quad (5.22)$$

где

$$B = B_1(j) + B_2(j) + \dots + B_k(j). \quad (5.23)$$

Очевидно, что

$$\sum_{i=1}^k b_i = k, \quad (5.24)$$

где  $b_i$  –  $i$ -я координата вектора  $B$ .

Таким образом, текущая маркировка  $m_k$  отражает информацию о том, себестоимость каких продуктов производства уже рассчитана и используется в правиле 1, определяющем для текущего момента времени  $k$ , все ли составляющие себестоимости для продукта, которому соответствует переход  $t_j$ , уже известны.

Для рассматриваемой проблемы характерны такие особенности: каждый переход срабатывает только один раз (действительно, нет необходимости дважды просчитывать себестоимость какого-либо продукта производства); условием окончания расчета является информация о том, что все переходы сработали; это равносильно  $m_k = [11\dots 1]^T$  – единичный вектор размерности  $n$ ,  $B = [11\dots 1]^T$  – единичный вектор размерности  $m$ .

Важное применение в описанных расчетах находит проблема достижимости в сетях Петри. Пусть заданы структура сети Петри (в матричном графическом представлении); начальная маркировка  $m_0$  конечная маркировка  $m = [11\dots 1]^T$ , т.е. единичный вектор размерности  $n$ . Необходимо определить, достижима ли конечная маркировка  $m$ ; иначе: возможно ли рассчитать себестоимость всех продуктов производства при заданной структуре исходных данных и известной себестоимости материалов вида «сырье»?

Эта задача сводится к проверке равенства

$$R^+ B = m - m_0, \quad (5.25)$$

в котором  $B$  и  $m$  единичные векторы размерности  $m$  и  $n$  соответственно.

Такая проверка является частным случаем более общей задачи исследования целочисленных систем линейных алгебраических уравнений вида  $Ax = b$  на наличие целочисленных неотрицательных решений из множества  $\{0, 1\}$  [45, 46].

Все вышеизложенное справедливо для несколько идеализированной структуры внутризаводских производственных связей, допускающей представление в виде сети Петри с отсутствующими тупиковыми разметками [46]. Однако металлургическое производство характеризуется наличием обратных связей, возникающих в случае, когда часть производимого продукта в дальнейшем используется для производства этого же самого продукта. С учетом этого структура, приведенная на рис. 5.4, примет более сложный вид (рис. 5.5).

Обратные связи могут быть как непосредственными, когда часть продукта используется на собственные нужды, например электроэнергия в цехе сетей и подстанций НЛМК (продукт  $P_4$  на рис. 5.5), так и опосредованными, например цепочка из продуктов  $3 \rightarrow 2 \rightarrow 1 \rightarrow 3$  на рис. 5.5. Такая структура сети Петри уже содержит тупиковые разметки, и поэтому равенство (5.25) не выполнимо. Однако тупиковые разметки могут быть устранены с помощью следующих простых преобразований.

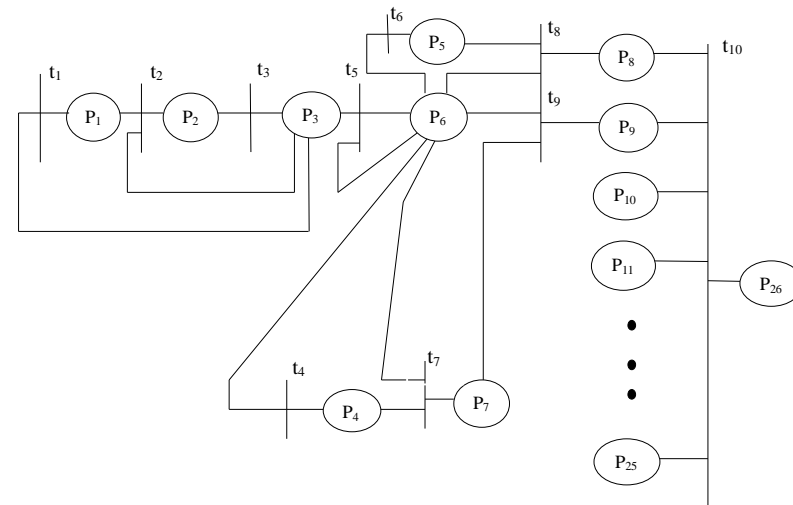


Рис. 5.5. Фрагмент структуры расчета себестоимости продукции ОАО «НЛМК» на примере аглопроизводства (с учетом обратных связей)

Для расчета себестоимости продукта, входящего в непосредственную обратную связь, необходимо решить линейное уравнение вида  $y = x + ay$ , равносильное уравнению  $y = x/(1 - a)$ , где  $y$  – искомая себестоимость продукта;  $x$  – себестоимость используемых в его производстве материалов;  $a$  – коэффициент, имеющий тот же смысл, что и  $a_{ij}$  в уравнении (5.16).

Для расчета себестоимости продуктов, входящих в опосредованную обратную связь, необходимо решить систему из  $n$  уравнений с  $n$  неизвестными, где  $n$  – количество продуктов, входящих в эту обратную связь. Для каждой обратной связи ввиду ее индивидуальности такая система уравнений будет иметь свой уникальный вид. В силу экономического смысла производимых вычислений определители систем будут ненулевыми и каждая из систем будет иметь единственное решение.

## Заключение

Развитие теории дискретно-аргументного моделирования обуславливается возрастающими потребностями решаемых практических задач, необходимостью создания базы для разработки автоматизированных систем автоматизации и управления. В монографии делается попытка постановки и решения новых сложных задач, не поддающихся достаточно полному исследованию в рамках известных ранее моделей.

Сложность и нетрадиционность новых моделей, определяемая их эволюцией во времени и распределенностью в пространстве, а также ориентация на применение ЭВМ при их реализации привели к возрастанию интереса к различным классам дискретно-аргументных моделей. В частности, предложенные подходы должны обеспечить решение задач выбора оптимальных технологических режимов, получения максимально приближающейся к заданному качеству продукции путем изменения технологических условий обработки полупродукта на последующих переделах на основе информации о предшествующей технологии. Она призвана предоставить более полную информацию о реальных затратах по всему сортаменту.

Исходным этапом работы является представление технологии в виде конечного автомата с выбором алфавита факторов и выходных величин. С помощью вероятностных автоматов и итеративных цепей можно реализовать многомерный случай анализа технологии и получить возможность рассматривать любое сочетание технологических факторов из разных переделов.

Предложенные в монографии подходы могут применяться к различным типам и структурам анализируемого технологического процесса и использоваться практически для любых процессов, математическое моделирование которых затруднено, но присутствует статистическая информация о процессе. После нахождения оптимальной технологии можно осуществлять контроль ее исполнения, и в случае выхода одного из факторов за рекомендуемые пределы осуществлять прогноз характеристик качества для данной технологии, а также анализировать дальнейшее продолжение технологии с попыткой улучшить уровень выходных свойств.

## Библиографический список

1. **Блюмин, С. Л.** Линейные клеточные машины: подход пространства состояний (обзор) [Текст] / С.Л. Блюмин, Р.Г. Фараджев // Автоматика и телемеханика. – 1982. – № 2. – С. 125-162.
2. **Бутковский, А. Г.** Структурная теория распределенных систем [Текст] / А.Г. Бутковский. – М.: Наука, 1977. – 320 с.
3. **Хопкрофт, Д.** Введение в теорию автоматов, языков и вычислений. – 2-е изд.; пер. с англ. [Текст] / Д. Хопкрофт, Р. Мотвин, Д. Ульман. – М.: Вильямс, 2002. – 528 с.
4. **Карпов, Ю. Г.** Теория автоматов [Текст] / Ю.Г. Карпов.– СПб.: Питер, 2002. – 224 с.
5. **Мелихов, А. Н.** Ориентированные графы и конечные автоматы [Текст] / А.Н. Мелихов. – М.: Наука, 1971. – 416 с.
6. **Коршунов, Ю. М.** Математические основы кибернетики [Текст] / Ю.М. Коршунов. – М.: Энергоиздат, 1987. – 376 с.
7. **Сигорский, В. П.** Математический аппарат инженера [Текст] / В.П. Сигорский. – Киев: Техника, 1975. – 768 с.
8. **Бухараев, Р. Г.** Вероятностные автоматы [Текст] / Р.Г. Бухараев. – Казань: КГУ, 1970. – 288 с.
9. **Бухараев, Р. Г.** Основы теории вероятностных автоматов [Текст] / Р.Г. Бухараев. – М.: Наука, 1985. – 288 с.
10. **Поспелов, Д. А.** Вероятностные автоматы [Текст] / Д.А. Поспелов. – М.: Наука, 1970. – 178 с.
11. **Кузнецов, Л. А.** Введение в САПР производства проката [Текст] / Л.А. Кузнецов. – М.: Металлургия, 1991. – 112 с.
12. **Кузнецов, Л. А.** Автоматизированная система выбора оптимальной технологии производства проката [Текст] / Л.А. Кузнецов, А.М. Корнеев // Известия вузов, Черная металлургия. – 1994. – № 5. – С. 45-48.
13. **Блюмин, С. Л.** Дискретно-аргументное моделирование систем обработки информации и управления [Текст]: учеб. пособие / С.Л. Блюмин, А.М. Корнеев. – Липецк: ЛипПИ, 1993. – 90 с.
14. **Гилл, А.** Линейные последовательные машины [Текст] / А. Гилл. – М.: Наука, 1974. – 288 с.
15. **Фараджев, Р. Г.** Линейные последовательные машины [Текст] / Р.Г. Фараджев. – М.: Наука, 1975. – 248 с.

16. **Блюмин, С. Л.** Преобразование сигналов и анализ систем [Текст]: учеб. пособие / С.Л. Блюмин, Л.А. Кузнецов, Ю.П. Качановский. – Воронеж: ВорПИ-ЛипПИ, 1990. – 78 с.
17. **Воеводин, А. Ф.** Численные методы расчета одномерных систем [Текст] / А. Ф. Воеводин, С. М. Шугрин. – Новосибирск: Наука, 1981. – 208 с.
18. **Барышев, В. Г.** К управлению системами с многомерным параметром [Текст] / В.Г. Барышев, С.Л. Блюмин, Л.А. Кузнецов // Автоматика и телемеханика. – 1977. – № 4. – С. 37-42.
19. **Блюмин, С. Л.** О дискретизации дифференциально-разностных систем [Текст] / С.Л. Блюмин, Л.А. Кузнецов // Автоматика и телемеханика. – 1978. – № 11. – С. 188-190.
20. **Блюмин, С. Л.** Соотношения типа Кэли-Гамильтона в теории дискретно-аргументных систем [Текст] / С.Л. Блюмин // Автоматика и телемеханика. – 1981. – № 9. – С. 133-142.
21. **Кузнецов, Л. А.** Применение УВМ для оптимизации тонколистовой прокатки [Текст] / Л.А. Кузнецов. – М.: Металлургия, 1988. – 304 с.
22. **Даджон, Д.** Цифровая обработка многомерных сигналов [Текст] / Д. Даджон, Р. Мерсеро. – М.: Мир, 1998. – 376 с.
23. **Нейман, Дж.** Теория самовоспроизводящихся автоматов [Текст] / Дж. Нейман. – М.: Мир, 1974. – 382 с.
24. **Блюмин, С. Л.** О конструировании линейными клеточными автоматами [Текст] / С.Л. Блюмин // Автоматика и телемеханика. – 1981. – № 11. – С. 131-138.
25. **Мур, Э. Ф.** Математические модели самовоспроизведения [Текст] / Э.Ф. Мур // Математические проблемы в биологии / Э.Ф. Мур. – М.: Мир, 1966. – С. 36-62.
26. **Блюмин, С. Л.** Двумерные m-цепи [Текст] / С. Л. Блюмин // Автоматика и телемеханика. – 1979. – № 10. – С. 58-64.
27. **Алберт, А.** Регрессия, псевдоинверсия и рекуррентное оценивание [Текст] / А. Алберт. – М.: Наука, 1977. – 285 с.
28. **Варден, Б. Л.** Алгебра [Текст] / Б.Л. Варден. – М.: Наука, 1976. – 353 с.
29. **Кузнецов, Л. А.** Расчет себестоимости трансформаторной стали с учетом незавершенного производства [Текст] / Л.А. Кузнецов [и др.] // Современные системы управления предприятием: сб. трудов междунар. науч.-техн. конф. – Липецк, 2001. – С. 119-121.
30. **Кузнецов, Л. А.** Система расчета затрат на производство проката [Текст] / Л.А. Кузнецов, В.А. Бреус, А.М. Корнеев // Сталь. – 1995. – № 3. – С. 63-64.
31. **Кузнецов, Л. А.** Система расчета затрат на производство проката [Текст] / Л.А. Кузнецов [и др.] // Известия вузов. Черная металлургия. – 1998. – № 9. – С. 72-76.
32. **Кузнецов, Л. А.** Расчет коэффициентов трудоемкости для определения себестоимости производства трансформаторной стали [Текст] / Л.А. Кузнецов [и др.] // Вестник ЛГТУ-ЛЭГИ – 2001. – № 1. – С. 134-139.
33. **Трахтман, А. М.** Основы теории дискретных сигналов на конечных интервалах [Текст] / А.М. Трахтман, В.А. Трахтман. – М.: Советское радио, 1975. – 208 с.
34. **Блюмин, С. Л.** Конечные преобразования Фурье и анализ конечных систем [Текст]: учеб. пособие / С.Л. Блюмин. – Воронеж: ВорПИ-ЛипПИ, 1991. – 80 с.
35. **Блюмин, С. Л.** О некоторых математических аспектах цифровой обработки сигналов [Текст] / С.Л. Блюмин, А.М. Шмырин // Радиотехника. – 1975. – № 8. – С. 24-27.
36. **Блюмин, С. Л.** Моделирование класса систем, состоящих из распределенной и сосредоточенной подсистем [Текст] / С.Л. Блюмин, Л.А. Кузнецов // Управление распределенными системами с подвижным воздействием. – Куйбышев: КПТИ, 1983. – С. 37.
37. **Вунш, Г.** Теория систем [Текст] / Г. Вунш – М.: Советское радио, 1978. – 288 с.
38. Технология системного моделирования [Текст] / под ред. С.В. Емельянова. – М.: Машиностроение, 1988. – 580 с.
39. **Питерсон, Дж.** Теория сетей Петри и моделирование систем [Текст] / Дж. Питерсон. – М.: Мир, 1984. – 264 с.
40. **Слепцов, А. И.** Автоматизация проектирования управляющих систем гибких автоматизированных производств [Текст] / А.И. Слепцов, А.А. Юрасов. – Киев: Техника, 1986. – 110 с.
41. Управление гибкими производственными системами: модели и алгоритмы [Текст] / под ред. С. В. Емельянова. – М.: Машиностроение, 1986. – 368 с.
42. **Лескин, А. А.** Сети Петри в моделировании и управлении [Текст] / А.А. Лескин, П.А. Мальцев, А.М. Спиридонов. – Л.: Наука, 1989. – 133 с.
43. **Васильев, В. В.** Сети Петри: параллельные алгоритмы и модели мультипроцессорных систем [Текст] / В.В. Васильев, В.В. Кузьмук. – Киев: Наукова Думка, 1991. – 213 с.

44. **Блюмин, С. Л.** Использование Сетей Петри при расчете себестоимости продуктов металлургического производства [Текст] / С.Л. Блюмин, Ю.В. Машковцев // Известия вузов. Черная металлургия. – 1995. – № 10. – С. 74-77.

45. **Грегори, Р.** Безошибочные вычисления [Текст] / Р. Грегори, Е. Кришнамурти. – М.: Высшая математика, 1988. – 208 с.

46. **Иванов Н. Н.** / Н.Н. Иванов// Автоматика и телемеханика. – 1991. – № 7. – С. 125-130.

НАУЧНОЕ ИЗДАНИЕ

Блюмин Семен Львович  
*доктор физико-математических наук, профессор*

Корнеев Андрей Мстиславович  
*кандидат технических наук, доцент*

УДК 62.529+669:162.24  
ББК 32.965.6

Б71 Блюмин С.Л., Корнеев А.М. Дискретное моделирование систем автоматизации и управления: монография. – Липецк: ЛЭГИ, 2005. – 124 с.

ISBN 5-900037-47-9

Представлены подходы к построению дискретных моделей, базирующихся на клеточной интерпретации; рассмотрены дискретные пространственные, дискретные пространственно-временные, конечно-аргументные модели. Технологический процесс представлен в виде клеток, каждая из которых соответствует отдельному технологическому агрегату или переделу. Использование конечных и вероятностных автоматов, итеративных цепей и сетей позволяет описать любой сложный технологический процесс и осуществлять управление технологией.

Техническое редактирование

Н.С. Правильниковой,  
И.Ф. Ковешниковой

Подписано в печать 21.11.2005. Формат 60x84/16.  
Бумага 55-60 г/м<sup>2</sup>. Гарнитура «Таймс». Усл. печ. л. 7,75.  
Тираж 200 экз. Заказ № 904. Цена свободная.  
Липецкий эколого-гуманитарный институт.  
398600, г. Липецк, ул. Интернациональная, 5а

Таблица переходов и выходов конечного автомата

$v_1, \dots, v_m$	$x'_1, \dots, x'_n$	$x_1, \dots, x_n$	$y_1, \dots, y_l$ (для КА Мили)	$y'_1, \dots, y'_l$ (для КА Мура)
$a_{11}, \dots, a_{i1}, \dots, a_{m1}$ ( $\sigma_1$ )	$b'_{11}, \dots, b'_{k1}, \dots, b'_{n1}$ ( $\xi'_1$ )	.	.	.
	...			
.....	$b'_{1j_1}, \dots, b'_{kj_k}, \dots, b'_{nj_n}$ ( $\xi'_\beta$ )	.	.	.
	.....			
$a_{1j_1}, \dots, a_{ij_i}, \dots, a_{mj_m}$ ( $\sigma_\alpha$ )	$b'_{1j_1}, \dots, b'_{kj_k}, \dots, b'_{nj_n}$ .	$c_{1j_1}, \dots, c_{pj_p}, \dots, c_{lj_l}$ ( $\tau_\gamma$ )	$c_{1j_1}, \dots, c_{pj_p}, \dots, c_{lj_l}$ ( $\tau'_\gamma$ )	.
	( $\xi'_\beta$ )			
.....	.....	.	.	.
	$a_{1j_1}, \dots, a_{ij_i}, \dots, a_{mj_m}$ ( $\sigma_\lambda$ )			
$a_{1j_1}, \dots, a_{ij_i}, \dots, a_{mj_m}$ ( $\sigma_\lambda$ )	( $\xi'_1$ )	.	.	.
	.....			
$a_{1j_1}, \dots, a_{ij_i}, \dots, a_{mj_m}$ ( $\sigma_\lambda$ )	( $\xi'_\beta$ )	.	.	.
	( $\xi'_B$ )			

Таблица переходов автоматов Мили и Мура

$x'_1, \dots, x'_n$ $v_1, \dots, v_m$	$b'_{1l}, \dots, b'_{kl}, \dots, b'_{nl}$ $(X'_1)$	$\dots$	$b'_{1j_1}, \dots, b'_{kj_k}, \dots, b'_{nj_n}$ $(X'_B)$	$\dots$	$b'_{1J_1}, \dots, b'_{kJ_k}, \dots, b'_{nJ_n}$ $(X'_B)$
$a_{1l}, \dots, a_{il}, \dots, a_{ml}$ $(\sigma_1)$					
$\dots$					
$a_{1j_1}, \dots, a_{ij_i}, \dots, a_{mj_m}$ $(\sigma_\alpha)$					
$\dots$					
$a_{1J_1}, \dots, a_{iJ_i}, \dots, a_{mJ_m}$ $(\sigma_\Lambda)$	$\cdot$ $\cdot$ $\cdot$ $b'_{1j_1}, \dots, b'_{kj_k}, \dots, b'_{nj_n}$ $\cdot$ $\cdot$ $\cdot$				



Таблица 1.3

Таблица выходов автомата Мили

$x'_1, \dots, x'_n$ $v_1, \dots, v_m$	$b'_{11}, \dots, b'_{k1}, \dots, b'_{n1}$ ( $\xi'_1$ )	...	$b'_{1j_1}, \dots, b'_{kj_k}, \dots, b'_{nj_n}$ ( $\xi'_\beta$ )	...	$b'_{1j_1}, \dots, b'_{kj_k}, \dots, b'_{nj_n}$ ( $\xi'_B$ )
$A_{11}, \dots, a_{i1}, \dots, a_{m1}$ ( $\sigma_1$ )					
...					
$a_{1j_1}, \dots, a_{ij_i}, \dots, a_{mj_m}$ ( $\sigma_\alpha$ )					
...					
$a_{1j_1}, \dots, a_{ij_i}, \dots, a_{mj_m}$ ( $\sigma_A$ )	$c_{1j_1}, \dots, c_{pj_p}, \dots, c_{ij_i}$				

Таблица выходов с одним входом для  $k$ -го передела

$v_{k1}, \dots, v_{kL_k}$	$x_{k-1,1}, \dots, x_{k-1, M_{k-1}}$		$y_{k1}, \dots, y_{kP_k}$	$z_{k1}, \dots, z_{kG_k}$
$a_{k11} \dots a_{k1k} \dots a_{kL_k 1}$ $(s_{1(k)})$ .....	$b_{(k-1)11}, \dots, b_{(k-1)m_{k-1}1}, \dots, b_{(k-1)M_{k-1}1}$ $(x_{1(k-1)})$ $b_{(k-1)1j_{k-1}}, \dots, b_{(k-1)m_{k-1}j_{mk-1}}, \dots, b_{(k-1)M_{k-1}j_{Mk-1}}$ $(x_{b(k-1)})$ $b_{(k-1)1J_{k-1}}, \dots, b_{(k-1)m_{k-1}J_{mk-1}}, \dots, b_{(k-1)M_{k-1}J_{Mk-1}}$ $(x_{B(k-1)})$			
.....		...	$c_{k1j_1} \dots c_{kR_k j_{R_k}} \dots c_{kR_k j_{R_k}}$ $(t_{g(k)})$	$d_{k1j_1} \dots d_{kgk j_{gk}} \dots$ $\dots d_{kG_k j_{G_k}}$ $(j_{1(k)})$
$a_{k1j_1} \dots a_{k1j_k} \dots a_{kL_k j_k}$ $s_{a(k)}$	$(x_{1(k-1)})$ $(x_{b(k-1)})$ $(x_{B(k-1)})$			
.....				
$a_{kL_k 1} \dots a_{kL_k j_k} \dots a_{kL_k L_k}$ $s_{A(k)}$	$(x_{1(k-1)})$ $(x_{b(k-1)})$ $(x_{B(k-1)})$			

Вероятности получения свойств для различных режимов

$V_{11} \dots V_{1L_1}$	$V_{21} \dots V_{2L_2}$	..	$V_{(k+1)1} \dots V_{(k+1)M_{k+1}}$	$x_{11} \dots x_{1M_1}$	$x_{k1} \dots x_{kM_k}$	$P_{1_{(k+1)}} \dots P_{g_{(k+1)}} \dots P_{\Gamma_{(k+1)}}$
$S_{1(1)}$	$S_{1(2)}$		$S_{A_{(k+1)}}$	$X_{1(1)}$	$X_{1(k)}$	$P_{t_{1(k+1)}} \dots P_{t_{g(k+1)}} \dots P_{t_{\Gamma(k+1)}}$
.	.		.	.	.	.
.	.		.	.	.	.
.	.		.	.	.	.
.	.		.	.	.	.
.	.		.	.	.	.
.	.		.	.	.	.
.	.		.	.	.	.
.	.		.	.	.	.
.	.		.	.	.	.
.	.		.	.	.	.
.	.		.	.	.	.
.	.		.	.	.	.
.	.		.	.	.	.
.	.		.	.	.	.
$S_{A(1)}$	$S_{A(2)}$		$S_{A_{(k+1)}}$	$X_{B(1)}$	$X_{B(k)}$	$P_{(C_{(k+1)1}^1, \dots, C_{(k+1)R_{k+1}^1}, \dots, C_{(k+1)j_{(k+1)}^1, \dots, C_{(k+1)R_{k+1}^j, \dots, C_{(k+1)R_{k+1}^J})}$ $\dots P_{(C_{(k+1)1}^j, \dots, C_{(k+1)R_{k+1}^j, \dots, C_{(k+1)R_{k+1}^J})}$

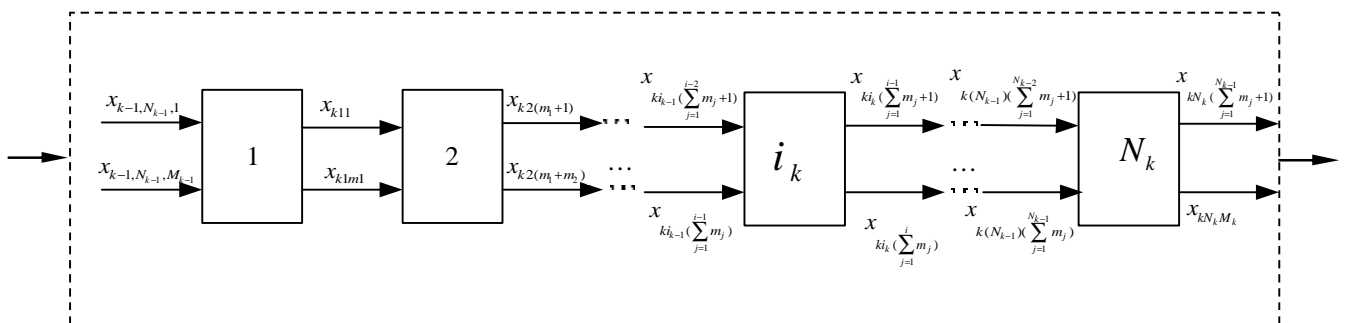
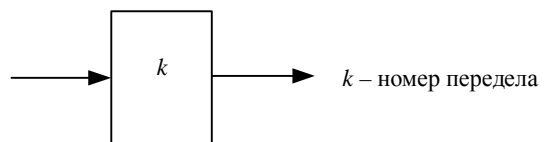


Рис. 2.16. Представление внутренней итеративной цепи

Переходы и выходы одномерной однонаправленной нелинейной итеративной сети для клетки  $s$  в момент времени  $t$

$u_m [t,s]$	$x_l [t,s-1], \dots, x_n [t,s-1]$	$\zeta_l [t-1,s], \dots, \zeta_l [t-1,s]$	$\zeta_l [t,s], \dots, \zeta_l [t,s]$	$x_l [t,s], \dots, x_n [t,s]$	$y_l [t,s], \dots, y_k [t,s]$
$(a_{11}, \dots, a_{ib}, \dots, a_{mi})$ $S_1$ $\mathbf{M}$	$C_l$ $\mathbf{M}$ $C_b$ $\mathbf{M}$ $C_B$	$Y_1, \dots, Y_n, \dots, Y_N$ $\mathbf{M}$ $Y_1, \dots, Y_n, \dots, Y_N$ $\mathbf{M}$ $Y_1, \dots, Y_n, \dots, Y_N$	$c_{1j_1}, \dots, c_{pj_p}, \dots, c_{mj_m}$ $(Y_n)$	$b_{1j_1}, \dots, b_{jr}, \dots, b_{nj_n}$ $(C_b)$	$d_{1j_1}, \dots, d_{qj_q}, \dots, d_{kj_k}$ $(t_g)$
$(a_{j1}, \dots, a_{ij\bar{j}}, \dots, a_{mjm})$ $S_a$	$C_l$ $\mathbf{M}$ $C_b$ $\mathbf{M}$ $C_B$	$Y_1, \dots, Y_n, \dots, Y_N$ $\mathbf{M}$ $Y_1, \dots, Y_n, \dots, Y_N$ $\mathbf{M}$ $Y_1, \dots, Y_n, \dots, Y_N$	$c_{1j_1}, \dots, c_{pj_p}, \dots, c_{mj_m}$ $(Y_n)$	$b_{1j_1}, \dots, b_{jr}, \dots, b_{nj_n}$ $(C_b)$	$d_{1j_1}, \dots, d_{qj_q}, \dots, d_{kj_k}$ $(t_g)$
$(a_{j1}, \dots, a_{ij\bar{j}}, \dots, a_{mjm})$ $S_A$	$C_l$ $\mathbf{M}$ $C_b$ $\mathbf{M}$ $C_B$	$Y_1, \dots, Y_n, \dots, Y_N$ $\mathbf{M}$ $Y_1, \dots, Y_n, \dots, Y_N$ $\mathbf{M}$ $Y_1, \dots, Y_n, \dots, Y_N$	$c_{1j_1}, \dots, c_{pj_p}, \dots, c_{mj_m}$ $(Y_n)$	$b_{1j_1}, \dots, b_{jr}, \dots, b_{nj_n}$ $(C_b)$	$d_{1j_1}, \dots, d_{qj_q}, \dots, d_{kj_k}$ $(t_g)$

Таблица 3.2

Функция переходов автономного ККА

$x(t-1; c_1; c_2)$	$x(t-1; c_1-1; c_2)$	$x(t-1; c_1+1; c_2)$	$x(t-1; c_1; c_2-1)$	$x(t-1; c_1; c_2+1)$	$x(t; c_1; c_2)$
0	0	0	0	0	0
0	0	0	0	1	-
0	0	0	1	0	-
0	0	0	1	1	-
0	0	1	0	0	-
0	0	1	0	1	-
0	0	1	1	0	-
0	0	1	1	1	-
0	1	0	0	0	-
0	1	0	0	1	-
0	1	0	1	0	-
0	1	0	1	1	-
0	1	1	0	0	-
0	1	1	0	1	-
0	1	1	1	0	-
0	1	1	1	1	-
1	0	0	0	0	-
1	0	0	0	1	-
1	0	0	1	0	-
1	0	0	1	1	-
1	0	1	0	0	-
1	0	1	0	1	-
1	0	1	1	0	-
1	0	1	1	1	-
1	1	0	0	0	-
1	1	0	0	1	-
1	1	0	1	0	-
1	1	0	1	1	-
1	1	1	0	0	-
1	1	1	0	1	-
1	1	1	1	0	-
1	1	1	1	1	-
1	1	1	1	0	-
1	1	1	1	1	-
1	1	1	1	1	-