Есть вещи непонятные кому-то одному. Все, вроде бы, всё понимают - а ты, ну, никак не можешь понять! Вот тут и получаются настоящие открытия. Только тут. Так-что надо ждать, когда это непонимание тебя осенит.

Максим Кантор

Содержание

ДИСКРЕТНО–ДЕТЕРМИНИРОВАННЫЕ МОДЕЛИ (*F*-СХЕМЫ) 2

ДЕТЕРМИНИРОВАННЫЕ АВТОМАТЫ 3
Устойчивые состояния автомата *5*
Синхронные и асинхронные автоматы *5*
КОНЕЧНЫЕ АВТОМАТЫ 6
Автоматы Мили и Мура *7*
Формы определения конечных автоматов *7*
Табличная форма 8
Матричная форма 9
Графовая форма 11
Примеры автоматного моделирования *13*

ДИСКРЕТНО–ДЕТЕРМИНИРОВАННЫЕ МОДЕЛИ (*F*-СХЕМЫ)

Дискретно-детерминированные модели используются в рамках подхода к проектированию программного обеспечения информационных систем известного как "*разработка, управляемая моделями*" (англ. Model-Driven Development – MDD, или Model Driven Architecture - MDA). Принцип MDD - все основные проектные решения принимаются по результатам оперирования моделями прототипа ПО, а реализация проекта в виде продукта становится формальной процедурой генерации программного кода. Использование MDD не означает, что полностью весь код разрабатываемого ПО генерируется по модели, но объём такого кода может быть большим. Иными словами, модели становятся основными артефактами[1], из которых частично (например, управляющая логика программы без вычислительной части) или полностью генерируется исполняемый код и другие артефакты (например, документация).

Дискретно-детерминированные модели интерпретируют выделенное между параметрами агрегата $\bar{\mathcal{A}}$ элементарное отношение $r(p_1, p_2, \cdots, p_{n_r}) \in R_{\bar{\mathcal{A}}}\{\vec{x}, \vec{v}, \vec{h}, \vec{y}, t\}$ как детерминированный конечный автомат, функционирующий в дискретном времени. Агрегат $\bar{\mathcal{A}}$ в целом рассматривается как структура связанных по входу/выходу элементарных детерминированных конечных автоматов, срабатывающих по событию поступления дискретных входных данных. В течении такта срабатывания агрегата $\bar{\mathcal{A}}$ осуществляется совместное преобразование поступивших дискретных значений входных экзогенных параметров $\vec{x}$ в дискретные значения экстернальных параметров $\vec{y}$ связанными элементарными детерминированными конечными автоматами. Значения интернальных параметров $\langle \vec{v}, \vec{h} \rangle$ агрегата $\bar{\mathcal{A}}$ в общем случае формируют состояния $\vec{Z}$ элементарных отношений $r(p_1, p_2, \cdots, p_{n_r})$ агрегата $\bar{\mathcal{A}}$ как детерминированных конечных автоматов.

[1] Артефакт — термин, который в программировании означает некоторый результат, полученный в процессе разработки программного обеспечения.

Для всех выделенных в агрегате $\bar{\mathcal{A}}$ элементарных отношений $r(p_1, p_2, \cdots, p_{n_r})$, в форме детерминированных конечных автоматов, выполняются следующие условия:

- все параметры агрегата $\bar{\mathcal{A}}$, включая параметр времени t, являются *дискретными*;
- все связываемые элементарным отношением $r(p_1, p_2, \cdots, p_{n_r})$ параметры $p_1, p_2, \cdots, p_{n_r}$ являются *детерминированными* и *статическими*;
- дискретное значение параметра времени $t \equiv i \in \{0,1,2, \ldots\}$ формируется моделью виртуального времени - $M_{\mathcal{T}}(0, \not\cong)$, где $\not\cong$ - неопределённая протяжённость такта срабатывания агрегата $\bar{\mathcal{A}}$, при завершении которого значение времени увеличивается на 1;
- инициация такта срабатывания агрегата $\bar{\mathcal{A}}$ происходит по событию очередного поступления входных дискретных значений в экзогенные паромеры $\vec{x}$;
- срабатывание элементарных автоматных отношений $r(p_1, p_2, \cdots, p_{n_r})$ инициируется событием обновления их входных значений;
- значения экстернальных параметров $\vec{y}_i$ агрегата $\bar{\mathcal{A}}$ формируются в результате завершения в i-ом такте срабатывания всех его элементарных автоматных отношений $r(p_1, p_2, \cdots, p_{n_r})$.

Детерминированные автоматы

Ниже структура и функционирование в дискретном времени элементарных отношений $r(p_1, p_2, \cdots, p_{n_r})$ агрегата $\bar{\mathcal{A}}$ в форме детерминированного конечного автомата A иллюстрируется графически (Рисунок 1):

Дискретно-детерминированные модели (*F*-схемы)

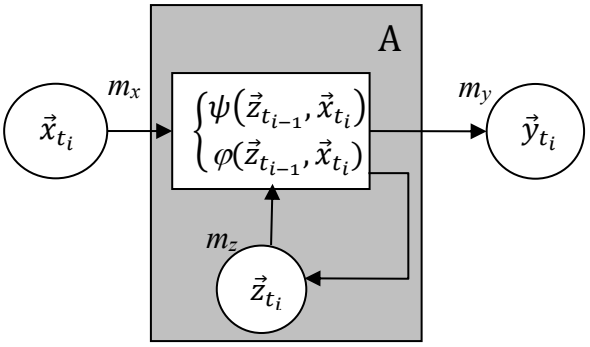

Рисунок 1

На рисунке 1 окружностями обозначены элементы памяти, предназначенные для хранения входного значения $\vec{x}_{t_i}$ (m_x - входная память), выходного значения $\vec{y}_{t_i}$ (m_y - выходная память), значения состояния автомата $\vec{z}_{t_i}$ (m_z - память состояния). В виде $\mathrm{A}(\vec{z}_{t_{i-1}},\vec{x}_{t_i}) = \begin{cases}\psi(\vec{z}_{t_{i-1}},\vec{x}_{t_i})\\ \varphi(\vec{z}_{t_{i-1}},\vec{x}_{t_i})\end{cases}$ обозначен оператор автомата A, который при срабатывании в очередном тике времени t_i выбирает из m_x текущее входное значение - $\vec{x}_{t_i}$, из m_z выбирает текущее значение состояния автомата - $\vec{z}_{t_{i-1}}$, формирует новое выходное значение - $\vec{y}_{t_i} = \psi(\vec{z}_{t_{i-1}},\vec{x}_{t_i})$, новое значение состояния - $\vec{z}_{t_i} = \varphi(\vec{z}_{t_{i-1}},\vec{x}_{t_i})$ и помещает их в память соответственно m_y и m_z.

Функционирование автомата в дискретном времени имеет специфические особенности. Прежде всего, заметим, что параметры $\vec{x}$, $\vec{z}$ и оператор преобразования $\mathrm{A}(\vec{z}_{t_{i-1}},\vec{x}_{t_i})$ от времени не зависят и, следовательно, являются статическими. Значения дискретного времени t_i используется лишь в качестве "индекса" для установления отношение строгого порядка следования входных знаков, выходных знаков и знаков состояний во времени. Последовательные событийные срабатывания автомата в дискретном времени называют *тактами*. Неопределённые по протяжённости тики модели виртуального времени - $M_T(0,\not\cong)$, ассоциируют с неопределёнными по протяжённости тактами событийного срабатывания. Параметр дискретного времени изменяет своё значение на единицу в момент завершения интервала срабатывания автомата и сохраняет это значение до следующего события срабатывания.

Устойчивые состояния автомата

На практике для моделирования используются автоматы с так называемыми устойчивыми состояниями. Состояние $\vec{z}_k$ автомата называется *устойчивым* для входного знака $\vec{x}_i \in X$, если поступление этого знака на вход автомата в состоянии $\vec{z}_k$ в такте срабатывания осуществляется переход в то же самое состояние - $\varphi(\vec{z}_k, \vec{x}_i) = \vec{z}_k$, и при этом выдаётся единственное соответствующее паре $(\vec{z}_k, \vec{x}_i)$ выходное значение $\vec{y}_j$, т.е. имеет место $\psi(\vec{z}_k, \vec{x}_i) = \vec{y}_j$. Иными словами, для входных знаков $\vec{x}_i \in X$, оставляющих автомат в том же текущем состоянии, на выходе должен формироваться один и тот же выходной знак $\vec{y}_j$. В противном случае, это интерпретируется как ошибка.

Синхронные и асинхронные автоматы

По способу формирования результатов срабатывания детерминированные автоматы делятся на *синхронные* и *асинхронные*.

Срабатывание синхронного автомата выглядит таким образом, что в течение такта срабатывания происходит однократное считывание входного значения $\vec{x}$ из памяти автомата, с учётом текущего состояния автомата $\vec{z}$ выдаётся на выход сформированное значение $\vec{y}$ и осуществляется переход автомата в новое состояние $\vec{z}$. После этого до начала следующего такта срабатывания автомат бездействует, и параметры автомата остаются неизменными.

Срабатывание асинхронного автомата основано на допущении, что в течение такта срабатывания он циклически многократно считывает из входной памяти один и тот же входной знак, изменяя своё состояние и выдавая соответствующие выходные знаки, пока не перейдёт в *устойчивое* не изменяемое состояние, в котором значение выходного знака до конца такта (до смены входного знака) остаётся неизменным. Очевидно, что асинхронный автомат для каждого входного знака должен всегда достигать устойчивого состояния, завершая тем самым такт срабатывания. В противном случае, такты срабатывания автомата будут не определёнными.

Конечные автоматы

На практике при интерпретации отношения между параметрами простого агрегата $\bar{A}$ в виде детерминированного автомата естественно полагать, что множества возможных входных и выходных значений, а также множество возможных состояний автомата являются конечными множествами. Такие детерминированные автоматы получили название *конечные автоматы* (*finite automaton*). Они собственно и составляют класс *F*-схем, используемых при моделировании для интерпретации отношений между параметрами простого агрегата.

В теории автоматов детерминированный конечный автомат (сокращённо КА) формально задаётся в виде:

$$F=<Z, X, Y, \varphi, \psi, z_0>$$

где:

X – конечное множество входных знаков (алфавит входных знаков x);

Y – конечное множество выходных знаков (алфавит выходных знаков y);

Z – конечное множество состояний (алфавит состояний z);

$z_0 \in Z$ – начальное состояние;

$\varphi(z,x)$ – функция переходов;

$\psi(z,x)$ – функция выходов.

Конечный автомат функционирует в дискретном автоматном виртуальном времени, когда параметр времени $t = 0,1,2, \ldots$. В начальный момент $t = 0$ автомат находится в исходном состоянии z_0. Если на вход конечного автомата, находящегося в начальном состоянии z_0, начать подавать в некоторой последовательности знаки входного алфавита x_0, x_1, x_2 …, то на выходе автомата будут последовательно появляться знаки выходного алфавита $y_0, y_1, y_2 \ldots$.

Замечание. Используемые ранее в описании простого агрегата векторные обозначения входных значений - $\vec{x}$, выходных значений - $\vec{y}$ и значений состояний - $\vec{z}$, далее будем ассоциировать с соответствующими абстрактными обозначениями

входных знаков - x, выходных знаков - y и знаков состояний - z, принятыми в теории конечных автоматов, т.е.: $\vec{x} \equiv x, \vec{y} \equiv y, \vec{z} \equiv z$.

Автоматы Мили и Мура

В зависимости от особенностей выполнения функций переходов $\phi(z,x)$ и выходов $\psi(z,x)$, реализуемых автоматом, различают конечные автоматы *Мили* и конечные автоматы *Мура*.

Для автомата **Мили** эти функции в момент срабатывания выполняются *параллельно*:

$$\begin{cases} z_{t+1} = \phi(z_t, x_t), \\ y_t = \psi(z_t, x_t) \end{cases} t = 0,1,2, \ldots.$$

Для автомата **Мура** эти функции выполняются в момент срабатывания *последовательно*, сначала функция перехода, затем функция выхода:

$$\begin{cases} z_t = \phi(z_{t-1}, x_t), \\ y_t = \psi(z_t) \end{cases} t = 1,2, \ldots.$$

Принципиальное отличие функционирования автомата Мура от автомата Мили в том, что выходной знак автомата Мура – y_t, формируется в соответствии с полученным в текущем такте новым значением состояния – z_t,. Из этого следует, что в автомате Мура выходное значение инициируется новым состоянием сразу после его получения в текущем такте. В автомате Мили новое состояние, полученное в текущем такте, используется только в следующем такте срабатывания автомата и вместе с поступившим в новом такте входным знаком используется для выработки нового выходного знака.

Формы определения конечных автоматов

На практике используются различные формы реализации моделей конечных автоматов, отличающиеся способом задания функций перехода и выхода. Так как модель автомата является дискретной, то для формального задания функций перехода и выхода могут применяться: *таблицы*, *матрицы*, *графы*. Кроме того

может использоваться и алгоритмическая форма задания функций перехода и выхода.

Табличная форма

При определении (задании) КА в **табличной форме** используется сводная таблица *переходов* и *выходов*, строки которой соответствуют входным значениям автомата, а столбцы – его состояниям. При этом обычно первый столбец слева соответствует начальному состоянию z_0. На пересечении i-ой строки и j-ого столбца таблицы переходов помещается соответствующее значение функции переходов $\varphi(z_j,x_i)$, а в таблице выходов – функции выходов $\psi(z_j, x_i)$.

Автомат **Мили** определяется следующей таблицей переходов и выходов (Таблица 1):

Таблица 1

x_i	z_k			
	z_0	z_1	…	z_K
Переходы $\varphi(z_j,x_i)$				
x_1	$\varphi(z_0,x_1)$	$\varphi(z_1,x_1)$	…	$\varphi(z_K,x_1)$
x_2	$\varphi(z_0,x_2)$	$\varphi(z_1,x_2)$	…	$\varphi(z_K,x_2)$
…				
x_I	…	…	…	…
Выходы $\psi(z_j,x_i)$				
x_1	$\psi(z_0,x_1)$	$\psi(z_1,x_1)$	…	$\psi(z_K,x_1)$
x_2	$\psi(z_0,x_2)$	$\psi(z_1,x_2)$	…	$\psi(z_K,x_2)$
…				
x_I	$\psi(z_0,x_I)$	$\psi(z_1,x_I)$	…	$\psi(z_K,x_I)$

Ниже приведён пример таблицы переходов и выходов автомата Мили с тремя состояниями: z_0, z_1, z_2; двумя входными знаками: x_1, x_2; и двумя выходными знаками y_1, y_2 (Таблица 2).

Таблица 2

x_i	z_k		
	z_0	z_1	z_2
Переходы			
x_1	z_2	z_0	z_0
x_2	z_0	z_2	z_1
Выходы			
x_1	y_1	y_1	y_2

x_2	y_1	y_2	y_1

Для автомата **Мура** вид таблицы упрощается, так как выходной знак зависит от полученного нового состояния. В результате получается т.н. *отмеченная таблица переходов*, в которой над каждым состоянием z_k автомата, обозначающим столбец таблицы, стоит соответствующий этому состоянию выходное значение $\psi(z_k)$. Отмеченная таблица переходов автомата Мура имеет вид (Таблица 3):

Таблица 3

$x_i \backslash y_j$	$\psi(z_k)$			
	$\psi(z_0)$	$\psi(z_1)$	…	$\psi(z_K)$
	z_0	z_1	…	z_K
x_1	$\varphi(z_0,x_1)$	$\varphi(z_1,x_1)$	…	$\varphi(z_K,x_1)$
x_2	$\varphi(z_0,x_2)$	$\varphi(z_1,x_2)$	…	$\varphi(z_K,x_2)$
…				
x_I	$\varphi(z_0,x_I)$	$\varphi(z_1,x_I)$	…	$\varphi(z_K,x_I)$

Ниже приведён пример отмеченной таблицы переходов автомата Мура с пятью состояниями, двумя входными и тремя выходными знаками (Таблица 4).

Таблица 4

$x_i \backslash y_j$	y				
	y_1	y_1	y_3	y_2	y_3
	z_0	z_1	z_2	z_3	z_4
x_1	z_1	z_4	z_4	z_2	z_2
x_2	z_3	z_1	z_1	z_0	z_0

Например, если на вход автомата в состоянии z_1 поступает знак x_1, то в соответствии с таблицей происходит переход автомата в состояние z_4, а на выход поступает знак y_3. Если в состоянии z_1 поступает знак x_2, то автомат переходит в то же состояние z_1, а на выходе тот же знак y_1.

Матричная форма

Матричная форма задания конечного автомата описывает конечный автомат с помощью, так называемой *матрицы соединений*. Матрица соединений есть квадратная матрица $C=[c_{ij}]$, где номер строки i указывает исходное состояние z_i, а номер столбца j определяет переход в новое состояние z_j. При этом элемент матрицы c_{ij} специфицирует реакцию автомата в виде набора пар x_k/y_s, где для

каждого входного знака x_k указывается получающийся при переходе из состояния z_i в состояние z_j соответствующий выходной знак y_s.

В случае *автомата Мили* элемент c_{ij} задаёт пару "вход/выход" - x_k/y_s, где x_k соответствует входному знаку, вызывающему переход из текущего состояния z_i в следующее состояние z_j, а y_s соответствует выходному знаку, выдаваемому автоматом при приходе входного знака x_k в состоянии z_i. Если переход из состояния z_i в состояние z_j происходит под действием не единственного, а нескольких разных входных знаков, тогда элемент матрицы c_{ij} представляет собой множество пар "вход/выход" для этого перехода, соединённых знаком дизъюнкции, например, $(x_1/y_1 \vee x_2/y_1 \vee x_3/y_2)$. Для автомата Мили, представленного в Таблице 2, матрица соединений имеет вид:

$$C = \begin{array}{c} \\ z_0 \\ z_1 \\ z_2 \end{array} \begin{array}{c} \begin{array}{ccc} z_0 & z_1 & z_2 \end{array} \\ \begin{bmatrix} x_2/y_1 & - & x_1/y_1 \\ x_1/y_1 & - & x_2/y_2 \\ x_1/y_2 & x_2/y_1 & - \end{bmatrix} \end{array}$$

Из таблицы следует, например, что если в текущем состоянии z_1 на вход приходит знак x_2, то автомат переходит в состояние z_2, и на выход выдаётся знак y_2. Теперь текущим состоянием становится z_2. Если в этом состоянии на вход приходит опять знак x_2, то автомат переходит в состояние z_1, и на выход выдаётся знак y_1.

Для *автомата Мура* элемент c_{ij} представляется дизъюнкцией только входных значений x_k, вызывающих переход из z_i в z_j, а соответствующие состояниям выходы описываются отдельно *вектором* выходных значений:

$$\vec{y} = \begin{bmatrix} \psi(z_0) \\ \psi(z_1) \\ \ldots\ldots \\ \psi(z_k) \end{bmatrix},$$

i-ая компонента которого соответствует выходному знаку на выходе автомата при переходе в состояние z_i.

Для автомата Мура, представленного в Таблице 4, матрица соединений и вектор выходов будут выглядеть как:

$$C = \begin{array}{|c|c|c|c|c|c|} \hline & z_0 & z_1 & z_2 & z_3 & z_4 \\ \hline z_0 & - & x_1 & - & x_2 & - \\ \hline z_1 & - & x_2 & - & - & x_1 \\ \hline z_2 & - & x_2 & - & - & x_1 \\ \hline z_3 & x_2 & - & x_1 & - & - \\ \hline z_4 & x_2 & - & x_1 & - & - \\ \hline \end{array} \qquad \vec{y} = \begin{bmatrix} y_1 \\ y_1 \\ y_3 \\ y_2 \\ y_3 \end{bmatrix}$$

Так как для детерминированных автоматов переходы однозначны, то представление автомата в матричной форме предполагает, что в матрице соединений C в каждой строке любой входной знак не должен встречаться более одного раза.

Графовая форма

Табличный и матричный способы описания конечного автомата удобны с точки зрения формирования структуры данных, используемых для программной реализации автомата, но они не обладают внешней наглядностью для лёгкого восприятия человеком логики работы автомата. С этой точки зрения более наглядным является способ задания конечного автомата в графическом виде.

Графовая форма автомата выражается помеченным ориентированным графом, набор вершин которого, соответствует различным состояниям автомата, а соединяющие вершины направленные дуги графа обозначают переходы автомата из одного состояния в другое. Если поступившее входное значение x_k вызывает переход из состояния z_i в состояние z_j, то на графе автомата направленная дуга, соединяющая вершину z_i с вершиной z_j, помечается значением x_k. Отражение на графе значений выходов $y = \psi(z_i, x_k)$ зависит от типа автомата. Заметим, что для детерминированных автоматов из любой вершины графа не могут выходить две и более дуги, помеченные одним и тем же входным значением.

В графе автомата Мили помечаются только дуги. То есть, если значение x_k поступает на вход автомата, находящегося в состоянии z_i, то соответствующая дуга перехода из состояния z_i в состояние z_j помечается двумя метками - значением x_k

и выходным значением $y = \psi(z_i, x_k)$. Чтобы отличить одну метку от другой, они могут быть записаны через дробь: x_k/y.

В графе автомата Мура помечаются отдельно дуги и вершины. Разметка графа такова, что если входное значение x_k, действуя на автомат в состоянии z_i, вызывает его переход в состояние z_j, то значение x_k помечает дугу, направленную из z_i в z_j, а выходным значением $y = \psi(z_j)$ помечают вершину z_j. На рисунке 2 приведены размеченные графы автоматов Мили и Мура, ранее заданные таблицами и матрицами соответственно.

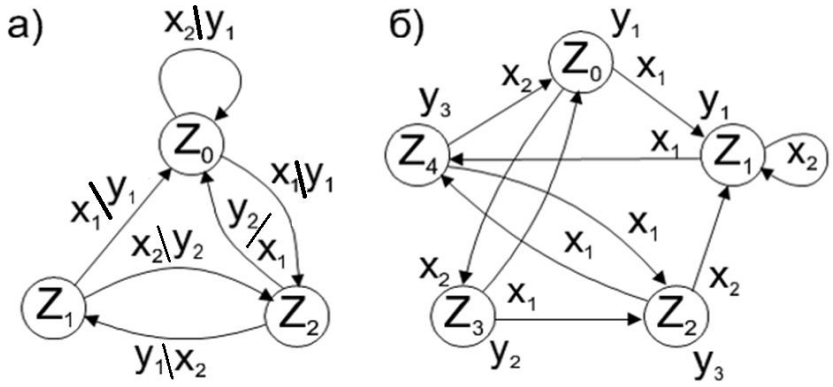

Рисунок 2 Графы автоматов Мили (а) и Мура (б)

На примере автомата Мура рассмотрим особенность представления асинхронного конечного автомата с устойчивыми состояниями, который описан в табличной форме (Таблица 5) и в виде графа (Рисунок 3).

Таблица 5

	y_j		
	y_1	y_3	y_2
x_i/z_i	z_0	z_1	z_2
x_1	z_1	z_1	z_1
x_2	z_2	z_1	z_2
x_3	z_0	z_0	z_2

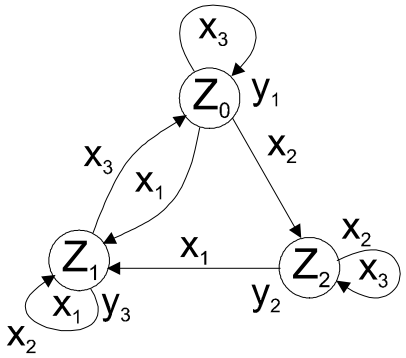

Рисунок 3 Граф асинхронного автомата Мура

Если в таблице переходов асинхронного автомата некоторое устойчивое состояние z_k стоит на пересечении строки x_i и столбца z_s, $s \neq k$, то это состояние z_k должно обязательно встретиться в этой же строке в столбце z_k. На графе видим, что если в асинхронном автомате по некоторому входному знаку осуществляется переход в некоторое устойчивое состояние, то в этом состоянии по тому же входному знаку должен происходить циклический переход в это же состояние.

Примеры автоматного моделирования

Проиллюстрируем на простых примерах применение конечных автоматов интерпретирующих отношения в форме автоматных преобразований значений входных параметров в выходные.

Пример 1

В качестве примера рассмотрим процесс управления во времени светофором, регулирующим дорожное движение на перекрёстке. Представим светофор как простой агрегат $\bar{\mathcal{A}}$ с параметрами x, y, z и одним выделенным между параметрами отношением $r(x, y, z)$, где $x \in \{x_0\}$ – экзогенный параметр агрегата, $z \in \{z_0, z_1, z_2, z_3\}$ – интернальный эндогенный параметр агрегата, $y \in \{y_0, y_1, y_2\}$ - эндогенный экстернальный параметр агрегата. Светофор, находясь в состоянии z преобразует во времени значение входного параметра x в значение выходного параметра y, т.е. - реализует отношение $r(x, y, z)$, которое выразим в форме синхронного конечного автомата, на вход которого событийно во времени поступает один и тот же входной знак $x \in X = \{x_0\}$, $x_0 \equiv$ "*Сменить_цвет*". По событию поступления входного знака x_0 светофор меняет цвет - автомат выдаёт выходной знак $y_j \in Y = \{y_0, y_1, y_2\}$. Выходные знаки автомата имеют следующие значения:

$y_0 \equiv$ *Красный*;
$y_1 \equiv$ *Жёлтый*;
$y_2 \equiv$ *Зелёный*.

Так как количество состояний светофора конечно, то история последовательного поступления входных знаков смены цвета светофора, разбивается на классы эквивалентных последовательностей, переводящих светофор в один и тот же цвет. В качестве эквивалентных

последовательностей истории поступления входных знаков смены цвета светофора, определим последовательности с одинаковой последней парой сменяемости цвета светофора. Иными словами, все промежуточные истории поступления входных знаков, для которых упорядоченные во времени пары последних изменений светофором цвета совпадают, будут эквивалентными. Так как количество цветов светофора ограничено, то история последовательного поступления знака смены цвета - x_0, разбивается на четыре класса эквивалентности с одинаковой последней парой сменяемости цвета светофора, приводящих светофор в одно и то же состояние $z_k \in Z = \{z_0, z_1, z_2, z_3\}$:

z_0 = *Жёлтый→Красный*,
z_1 = *Красный→Жёлтый*,
z_2 = *Жёлтый→Зелёный*,
z_3 = *Зелёный→Жёлтый.*

Функции перехода и выходов конечного автомата зададим графом, в котором вершины соответствуют состояниям, а ребро, помеченное значением x_0, определяет соответствующее изменение состояния автомата. При этом заметим, что выходной знак y_j зависит только от текущего состояния автомата. Поэтому граф автомата будет иметь следующий вид (Рисунок 4):

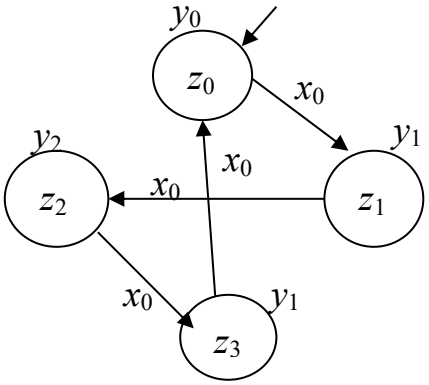

Рисунок 4 Модель отношения "поведение светофора"

Начиная с начального состояния z_0, светофор по событию k-ого поступления входного знака x_0 - "*Сменить_цвет"*, переходит из текущего состояния $z_l \in \{z_0, z_1, z_2, z_3\}$ в следующее - $z_m \in \{z_0, z_1, z_2, z_3\}$, и выдаёт новый выходной знак $y_j \in \{y_0, y_1, y_2\}$ — меняет цвет.

Полученный синхронный конечный автомат интерпретирует отношение $r(x, y, z)$ как автомат Мура, так как формирование выходного знака y_j определяется только его текущим состоянием z_k. Так как автомат синхронный, то он может функционировать как в реальном, так и в виртуальном времени, срабатывая по событию регулярного или спорадического поступления знака x_0 – смены цвета светофора.

Пример 2

Построим автоматную модель отношения, связывающего параметры простого агрегата, моделирующего поведение "умного родителя", реагирующего на оценки, получаемые учеником в школе. Очевидно, автомат будет синхронным и работать только в виртуальном времени, так как получение оценок учеником происходит сугубо спорадически.

Пусть результаты учёбы оцениваются либо оценкой "*хорошо*"- 5, либо "*плохо*"- 2. Реакция родителя связана с принятием различных мер стимулирования успешной учёбы.

Полученные учеником оценки являются входными значениями автомата: $x \in X = \{2, 5\}$. Положим, что история получения оценок вызывает у родителя следующие реакции: $Y = \{y_0, y_1, y_2, y_3, y_4, y_5\}$:

y_0: «*негодовать*»;

y_1: «*ругать*»;

y_2: «*успокаивать*»;

y_3: «*подбадривать*»;

y_4: «*радоваться*»;

y_5: «*ликовать*».

Эквивалентные предыстории получения оценок, формирующие одинаковые состояния автомата, свяжем с конечным набором различных комбинаций последних подряд идущих оценок, формирующих состояние родителя. В результате определим четыре абстрактных состояния родителя: $Z=\{z_0, z_1, z_2, z_3\}$:

z_0 – <∅> - оценок нет, или <…,2,5>;
z_1 – <2> - первая оценка - двойка, или <…,5,2>;
z_2 – <…,2,2> - подряд две двойки;
z_3 – <5> - первая оценка - пять, или <…,5,5>- подряд две пятёрки.

Формирование состояния родителя и его реакции на текущую историю получения оценок зададим с помощью графа конечного автомата (Рисунок 5):

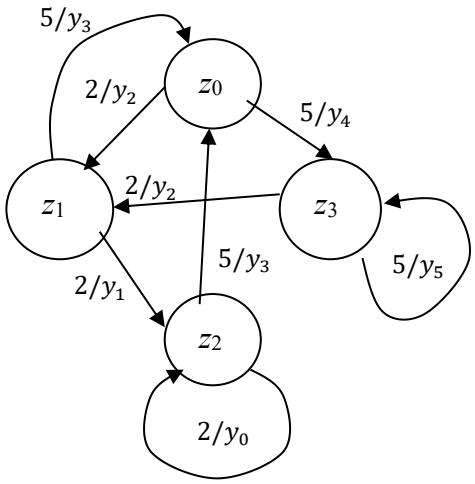

Рисунок 5 Модель "поведение родителя"

Состояние z_0 - история оценок пуста. Родитель, находясь в начальном состоянии z_0, под воздействием входной оценки переходит в состояние, соответствующее предыстории получения оценок. Например, когда в самом начале в состоянии z_0 возникает первая оценка 5, то автомат из состояния z_0 переходит в состояние z_3 и выдаётся выходное значение y_4: "*радоваться*". Если автомат оказался в состоянии z_0 из состояния z_1, т.е. оценка 5 следует за предыдущей оценкой 2, то выдаётся выходное значение y_3: "*подбадривать*". Если автомат из состояния z_0 переходит в состояние z_3 (вторая подряд оценка 5) , то выдаётся выходное значение y_4: «*радоваться*». Если из состояния z_3 опять происходит переход в состояние z_3 (вторая подряд оценка 5), то выдаётся выходное значение y_5: «*ликовать*». Если же несколько раз подряд идут оценки 2, то это быстро переводит родителя в состояние z_3, когда на очередную оценку 2 выдаётся реакция родителя y_0: "*негодовать*".

Заметим, что ученика, получившего подряд оценку 2, ожидает разная реакция родителя в зависимости от его состояния - от предыстории результатов учёбы ученика. Автоматная модель родителя помнит предыдущие результаты учёбы и строит своё поведение с учётом предыстории его успехов и неудач. Например, если в текущей предыстории в конце идут подряд две двойки - <…,2,2>, то независимо от того, какая предыстория этому предшествовала, родитель негодует, а если текущая история заканчивается последовательностью оценок <…,5,2>, то родитель будет ученика успокаивать. Каждая предыстория определяет текущее состояние автомата, при этом все предыстории разбиваются на конечное множество классов эквивалентности (предыстории одного класса приводят автомат в одно и то же состояние). Например, любая предыстория <…,2,5> эквивалентна начальному состоянию автомата.

Построенный конечный автомат является синхронным автоматом Мили, работающим в виртуальном времени.

Пример 3

Построим автоматную модель отношение $r(x, y, z)$, интерпретирующего функционирование простейшего телефонного аппарата с памятью для сохранения набранного номера абонента. Для простоты положим, что количество номеров абонентов равно 3, номер состоит из одной цифры и набирается нажатием кнопки с этой цифрой.

Для вызова абонента предварительно снимается телефонная трубка. При снятой трубке аппарат способен инициировать звонок абоненту двумя способами: нажатием кнопки с номером абонента или нажатием специальной кнопки *K* - вызов абонента по номеру, сохранённому в памяти.

При любом положении трубки память можно очистить нажатием кнопки *C* – очистка памяти.

При любом положении трубки и пустой памяти номер абонента, соответствующий нажатой кнопке, сохраняется в памяти. Если память не пуста, то нажатие кнопки абонента игнорируется.

Множество *входных знаков X* автомата определим как набор следующих входных знаков $X = \{x0, x1, x2, \dots, x6\}$; где:

$x0 \equiv T_0$ – "*повешена_трубка*";
$x1 \equiv T_1$ – "*снята_трубка*";
$x2 \equiv K$ – "*нажата_кнопка_вызова_номера_из_памяти*";
$x3 \equiv C$ – "*нажата_кнопка_очистки_памяти*";
$x4 \equiv L_1$ – "*нажата_кнопка_вызова_номера*_1";
$x5 \equiv L_2$ – "*нажата_кнопка_вызова_номера*_2";
$x6 \equiv L_3$ – "*нажата_кнопка_вызова_номера*_3".

Будем полагать, что часы "показывают" виртуальное временя, изменяющееся на 1 по событию поступления входного знака, т.е. имеем модель виртуального времени - $M_T(0, \not\equiv)$, показание часов $t \equiv i = 0,1,2,3, \dots$ совпадает с номером такта срабатывания автомата. В каждом такте выполняется ровно одно срабатывание автомата, т.е. режим работы - *синхронный*.

Состояние автомата определяется историей смены положения телефонной трубки (снята, повешена) и занятости памяти номером n. Классы эквивалентности разделяют все истории использования во времени телефонного аппарата по последней комбинации пары: *текущее положение трубки*, *номер в памяти*. Поэтому состояния будем специфицировать парой признаков (m, α), где:

- $m \in \{0$ - "память_пуста", n - "в_памяти_номер_n"$\}$ - содержимое памяти,
- $\alpha \in \{0$ - "трубка_повешена", 1 - "трубка_снята"$\}$ - положение трубки.

Тогда множество Z состояний автомата будет формироваться комбинацией пары (m, α): $z \in Z = \{(0,0), (0,1), (1,0), (1,1), (2,0), (2,1), (3,0), (3,1)\}$. В начальный момент времени (в нулевом такте срабатывания) состояние автомата будет (0,0) – ("память_пуста","трубка_повешена").

Выходные знаки будут при срабатывании автомата представляться *сериями импульсов*, выдаваемых телефоном в канал связи в количестве, равном значению набранного номера n. Тогда множество выходных знаков $y \in Y = \{0,1,2,3\}$, где $y = 0$, когда импульсы в канал не выдаются, и $y = n > 0$ – количество импульсов, соответствующее телефонному номеру, выданных последовательно в канал связи. Заметим, что формируемый автоматом выходной знак y, будет определяться его текущим состоянием (m, α) и поступившим на вход нажатием либо кнопки L_n, либо кнопки K. Из этого следует, что автоматная модель интерпретируемого отношения $r(x, y, z)$ является автоматом Мили.

Определим *функцию перехода* автомата Мили – $z_{i+1} = \varphi(z_i, x_i)$, где z_i – состояние автомата в i-ом такте срабатывания. Предположим, что в некотором такте срабатывания $i = 0,1,2,3, \ldots$ модель находится в состоянии $z_i \equiv (m, \alpha)_i$ и на вход поступает инициирующий срабатывание автомата некий входной знак $x_i \in \{T_0, T_1, K, L_n, C\}$. Для понимания того, в каком состоянии автомат окажется в $(i+1)$-ом такте, рассмотрим следующие возможные комбинации входных знаков x_i и текущих состояний аппарата z_i:

1. $x0 \equiv T_0$ – "*повешена_трубка*".
   Если автомат был в состоянии $(m, 1)$ - "*трубка_снята*", то для любого содержимого памяти он перейдёт в состояние - $(m, 0)$ – "*трубка_весит*", $m = \overline{0,3}$.
   Если автомат был в состоянии $(m, 0)$ - "*трубка_весит*", то для любого содержимого памяти он переходит в то же состояние - $(m, 0)$ – "*трубка_весит*", $m = \overline{0,3}$.

2. $x1 \equiv T_1$ – "*снята_трубка*".
   Если автомат был в состоянии $(m, 0)$ - "*трубка_весит*", то для любого содержимого памяти он переходит в состояние $(m, 1)$ – "*трубка_снята*", $m = \overline{0,3}$.
   Если автомат был в состоянии $(m, 1)$ - "*трубка_снята*", то для любого содержимого памяти он переходит в то же состояние - $(m, 1)$ – "*трубка_снята*", $m = \overline{0,3}$.

3. $x2 \equiv K$ – "*нажата_кнопка_вызова_номера_из_памяти*".
   Нажатие кнопки памяти в любом состоянии (m, α) не влияет на содержимое памяти и положение трубки, следовательно автомат перейдёт в то же состояние (m, α), $m = \overline{0,3}$.

4. $x3 \equiv C$ – *нажата кнопка очистки памяти*.
   В любом состоянии (m, α) память очищается и автомат переходит в состояние $(0, \alpha)$.

5. $x4 \equiv L_1$ – "*нажата_кнопка_вызова_номера_*1".
   Если состояние автомата было $(0, \alpha)$, то, независимо от положения трубки, автомат перейдёт в состояние $(1, \alpha)$ – введённый номер 1 записывается в пустую память.
   Если же состояние автомата было (m, α) – память не пустая и хранит ранее введённый номер $m \neq 0$, то состояние остаётся прежним (номер в памяти не заменяется).

6. $x5 \equiv L_2$ – "*нажата_кнопка_вызова_номера_*2".
   Если состояние автомата было $(0, \alpha)$, то, независимо от положения трубки, автомат перейдёт в состояние $(2, \alpha)$ – введённый номер 2 записывается в пустую память.
   Если же состояние автомата было (m, α) – память не пустая и хранит ранее введённый номер $m \neq 0$, то состояние остаётся прежним (номер в памяти не заменяется).

7. $x6 \equiv L_3$ – "*нажата_кнопка_вызова_номера_*3".
   Если состояние автомата было $(0, \alpha)$, то, независимо от положения трубки, автомат перейдёт в состояние $(3, \alpha)$ – введённый номер 3 записывается в пустую память.

Если же состояние автомата было (m, α) – память не пустая и хранит ранее введённый номер $m \neq 0$, то состояние остаётся прежним (номер в памяти не заменяется).

В итоге функция перехода состояний $z_{i+1} = \phi(z_i, x_i)$ в форме аналитического отображения пар $\langle z, x\rangle_i$ будет иметь вид:

$$\phi: \begin{cases} \langle (m,0), T_0\rangle_i \to (m,0)_{i+1}, \forall m = \overline{0,3}; \\ \langle (m,1), T_0\rangle_i \to (m,0)_{i+1}, \forall m = \overline{0,3}; \\ \langle (m,0), T_1\rangle_i \to (m,1)_{i+1}, \forall m = \overline{0,3}; \\ \langle (m,1), T_1\rangle_i \to (m,1)_{i+1}, \forall m = \overline{0,3}; \\ \langle (m,\alpha), K\rangle_i \to (m,\alpha)_{i+1}, \forall m = \overline{0,3}; \\ \langle (m,\alpha), C\rangle_i \to (0,\alpha)_{i+1}, \forall m = \overline{0,3}; \\ \langle (0,\alpha), L_n\rangle_i \to (n,\alpha)_{i+1}, n = 1,2,3; \\ \langle (m,\alpha), L_n\rangle_i \to (m,\alpha)_{i+1}, \forall m > 0, n = 1,2,3. \end{cases}.$$

Определим теперь для автомата Мили *функцию выхода* $y_i = \psi(z_i, x_i)$ – количество импульсов, посылаемых в телефонную линию, перечисляя возможные комбинации знаков x_i на входе и состояний аппарата z_i, и сопоставляя им знаки y_i на выходе аппарата:

1. $x_i \in \{T_0, T_1, C\}$. Данные входные знаки в любом состоянии автомата z_i не инициируют посылки в линию серии импульсов, т.е. - $y_i = 0$.
2. $x_i = K$. В этом случае в состоянии $(m, 1)$ – $m > 0$ и трубка снята, и наличие в памяти номера m, в телефонную сеть будет послана серия импульсов в количестве хранимого номера - m. Следовательно, $y_i = m$. А если текущее состояние есть $(m, 0)$ или $(0, \alpha)$, то посылки в линию импульсов не будет - - $y_i = 0$.
3. $x_i = L_n$ – *нажат номер* n. В состоянии $(m, 1)$ либо $(0,1)$ в телефонную сеть будет послана серия из n импульсов, соответствующих нажатому номеру - $y_i = n$, а в состоянии $(m, 0)$ – не будет: $y_i = 0$.

В итоге функция выхода $y_i = \psi(z_i, x_i)$ в форме отображения будет иметь вид:

$$\psi: \begin{cases} \langle (m,\alpha), T_0\rangle_i \to 0; \\ \langle (m,\alpha), T_1\rangle_i \to 0; \\ \langle (m,\alpha), C\rangle_i \to 0; \\ \langle (m,1), K\rangle_i \to m, m = 1,2,3; \\ \langle (m,0), K\rangle_i \to 0; \\ \langle (0,\alpha), K\rangle_i \to 0; \\ \langle (m,1), L_n\rangle_i \to n, \forall m = \overline{0,3}, n = 1,2,3. \end{cases}$$

Построенная аналитическая автоматная модель относится к классу автоматов Мили.

Рекомендация. По полученной модели автомата Мили в аналитической форме постройте её аналог в табличной, матричной и графовой форме.