

Содержание

ДИСКРЕТНО-СТОХАСТИЧЕСКИЕ МОДЕЛИ (<i>P</i>-СХЕМЫ)	2
МАТЕМАТИЧЕСКАЯ СХЕМА СТОХАСТИЧЕСКОГО АВТОМАТА	3
КЛАССЫ СТОХАСТИЧЕСКИХ АВТОМАТОВ.....	7
<i>Стохастический автомат Мили</i>	7
<i>Стохастический автомат Мура</i>	8
<i>Частично-детерминированные стохастические автоматы</i>	9
Y_0 -детерминированный автомат.....	10
Модель дискретной марковской цепи	11
Финальные вероятности дискретной марковской цепи	15
ПРИМЕНЕНИЕ СТОХАСТИЧЕСКИХ АВТОМАТОВ ДЛЯ МОДЕЛИРОВАНИЯ ПРОГРАММ.....	17
<i>Тестирование программной логики</i>	17
<i>Тестирование точек завершения программы</i>	18
<i>Тестирование исключительных состояний</i>	20

Дискретно-событийные стохастические модели (*P*-схемы)

ДИСКРЕТНО-СТОХАСТИЧЕСКИЕ МОДЕЛИ
(*P*-СХЕМЫ)

Проиллюстрируем, в каком случае для интерпретации выделенных в простом агрегате $\bar{\mathcal{A}}$ автоматных отношений возникает необходимость применения дискретно-стохастических моделей класса *P*-схем. Положим что разрабатывается модель прототипа распределённой программной системы, обслуживающей запросы пользователей. Априорно оцениваемой эксплуатационной характеристикой является время ожидания отклика программной системы на поступающие от пользователей запросы (время ожидания результата). Положим, что распределённые программные модули параллельно функционируют в узлах вычислительной сети, обмениваясь между собой данными посредством сообщений, передаваемых между модулями по каналам локальной сети. Очевидно, что время отклика на запрос будет зависеть не только от времени распределённой обработки модулями поступившего запроса, но и от пропускной способности каналов передачи данных сети. Оценка времени, которое тратится на передачу сообщений по каналу, полагаясь только на технические характеристики пропускной способности используемых физических средств связи, не позволит получить адекватной оценки так как сообщения, передаваемые по каналам, подвергаются воздействию случайных помех, искажающих передаваемые по каналу данные. Контроль ошибок и повторную передачу искажённых данных осуществляют *протоколы передачи сообщений* - специальные системные программные компоненты вычислительной сети, отвечающие за приём/передачу сообщений. Если в процессе передачи сообщение искажается, то протокол повторяет приём/передачу сообщения, при этом модули обработки находятся в состоянии ожидания доставки сообщений. Если сообщение доставлено без искажения, то протокол передачи завершает работу, а ожидающие сообщение программные модули выходят из состояния ожидания и начинают обработку принятых данных. Очевидно, что с учётом возможных искажений сообщений *время* получения пользователем результатов обработки запросов будет случайным, и оценить время отклика на запрос пользователя к системе можно только

статистически. Поэтому для интерпретации автоматных отношений используются так называемые *вероятностные* (стохастические) автоматы (англ. Probabilistic automaton) – P -автоматы, которые являются конечными автоматами со стохастическим поведением.

Дискретно-стохастическая интерпретация выделенного элементарного отношения $r(p_1, p_2, \dots, p_{n_r}) \in R_{\bar{\mathcal{A}}} \{\vec{x}, \vec{v}, \vec{h}, \vec{y}, t\}$ между параметрами простого агрегата $\bar{\mathcal{A}}$ определяет его как вероятностный конечный автомат, для которого множество параметров $\{p_1, p_2, \dots, p_{n_r}\}$ разделяется на три не пересекающиеся подмножества, составляющих для автомата: входное значение \vec{x} , выходное значение \vec{y} , значение состояния \vec{z} . При этом выполняются следующие условия:

- входные значения \vec{x} , выходные значения \vec{y} , значения состояний \vec{z} автоматного отношения $r(p_1, p_2, \dots, p_{n_r})$, являются *дискретными, статическими*;
- выходные значения параметров \vec{y} и/или состояния \vec{z} являются *стохастическими*;
- инициация очередного такта срабатывания во времени автоматного отношения $r(p_1, p_2, \dots, p_{n_r})$ происходит по событию обновления входного значения \vec{x} ;
- изменяющиеся во времени значения входных параметров \vec{x} , выходных параметров - \vec{y} , и параметров состояния - \vec{z} , помечаются дискретными метками t_i модели реального времени - $M_{\mathcal{T}}(0, \tau)$, или индексами модели виртуального времени $i \in \{0, 1, 2, \dots\}$ - $M_{\mathcal{T}}(0, \infty)$, где ∞ - неопределённая протяжённость такта срабатывания автоматного отношения $r(p_1, p_2, \dots, p_{n_r})$.

Математическая схема стохастического автомата

В общем виде конечный вероятностный P -автомат определяется как дискретный стохастический потактный преобразователь информации с памятью. Концепция функционирования конечного стохастического автомата во времени та

же, что и для детерминированного конечного автомата, но в каждом такте его результаты срабатывания является вероятностным. То есть, переход автомата в новое состояние и/или сформированное выходное значение являются случайными.

Для определения вероятностного P -автомата представим функцию переходов и функцию выходов единым *отображением* пар (x_i, z_r) - входного знака x_i и текущего состояния z_r , в пары (z_k, y_j) - нового состояния z_k и выходного знака y_j . Для этого введём в рассмотрение два конечных множества:

– множество $G \subset X \times Z$ – подмножество декартова произведения конечного множества входных знаков $x_i \in X, i = 1, 2, \dots, I$; и конечного множества состояний $z_r \in Z, r = 1, 2, \dots, R$; $(x_i, z_r) \in G$;

– множество всевозможных пар $(z_k, y_j) \in \Phi \subset Z \times Y$ - подмножество декартова произведения конечного множества состояний автомата $z_k \in Z, k = 1, 2, \dots, K$; и конечного множества выходных знаков $y_j \in Y, j = 1, 2, \dots, J$.

Далее будем полагать, что определена двумерная таблица, в строках которой каждой паре $(x_i, z_s) \in G$ заданы *законы распределения вероятностей* $p_{(x_i, z_s)}^{(z_k, y_j)}$:

$$B_{is} = \left\{ p_{(x_i, z_s)}^{(z_k, y_j)} \right\}_{(z_k, y_j) \in \Phi}, \forall (x_i, z_s) \in G;$$

где $p_{(x_i, z_s)}^{(z_k, y_j)} \equiv p_{is}^{kj}$ - условная вероятность того, что если в такте срабатывания на его вход поступило значение x_i в состоянии автомата z_s , то автомат перейдёт в состояние z_k и на выходе появится значение y_j . Так как множества G и Φ дискретны и конечны, то законы распределения этих вероятностей в таблице для всех пар $(x_i, z_s) \in G$ зададим строками следующего вида:

$$B = \begin{array}{|c|c|c|c|c|c|} \hline (z_k, y_j) \in \Phi & (z_1, y_1) & (z_1, y_2) & \dots & (z_k, y_{J-1}) & (z_k, y_J) \\ \hline (x_i, z_s) \in G & p_{is}^{11} & p_{is}^{12} & \dots & p_{is}^{K(J-1)} & p_{is}^{KJ} \\ \hline \end{array} \quad i=1, 2, \dots, I; s=1, 2, \dots, K.$$

Таблицу заданных для каждой пары $(x_i, z_s) \in G$ законов распределения вероятностей переходов/выходов автомата обозначим символом B . Число представленных в таблице строками законов распределения вероятностей равно числу элементов множества G , а сама таблица буде представлять собой

Дискретно-событийные стохастические модели (P -схемы)

одновременно оба отображения - $\varphi: G \rightarrow Z$ и $\psi: G \rightarrow Y$. Очевидно, что для вероятностей p_{is}^{kj} по строкам таблицы выполняется условие нормировки:

$$\sum_{k=1}^K \sum_{j=1}^J p_{is}^{kj} = 1; \forall (x_i, z_s) \in G.$$

В результате математическая схема обобщённого P -автомата формально представится четвёркой элементов вида:

$$P = \langle X, Y, Z, B \rangle.$$

Для однозначного определения стохастического автомата необходимо ещё задать распределение вероятностей его начального состояния:

$z_s \in Z$	z_1	z_2	...	z_{K-1}	z_K
$P_s(0)$	$P_1(0)$	$P_2(0)$...	$P_{K-1}(0)$	$P_K(0)$

В качестве свойств P -автомата, которые в общем случае оказывают влияние на оценку анализируемых эксплуатационных показателей прототипа информационной системы, выступают такие вероятностные характеристики как:

- $P_{(x_i, z_s)}^{z_k}$ - вероятность оказаться P -автомату в такте срабатывания в состоянии z_k , независимо от выходного значения, если срабатывание автомата определяется парой $(x_i, z_s) \in G$;
- $P_{(x_i, z_s)}^{y_j}$ - вероятность получить на выходе P -автомата в такте срабатывания значение y_j , независимо от нового состояния, если срабатывание автомата определяется парой $(x_i, z_s) \in G$;
- $P_{(x, z)}^{(z_s, y_j)}(n)$ - вероятность оказаться на n -ом такте срабатывания в состоянии $z_s \in Z$ и получить на выходе знак y_j , если на нулевом такте автомат был в состоянии z и на вход поступил знак x , $(x, z) \in G$.

Первые две характеристики вычисляются аналитически. Для пары (x_i, z_s) вероятность оказаться автомату в такте срабатывания в состоянии z_k (независимо от выходного значения) будет равна:

Дискретно-событийные стохастические модели (P -схемы)

$$\mathbf{P}_{(x_i, z_s)}^{z_k} = \sum_{j=1}^J p_{is}^{kj}, k = \overline{1, K}; \forall (x_i, z_s) \in G.$$

Суммируются все вероятности p_{is}^{kj} возможных переходов при поступлении знака x_i из состояния z_s в состояние z_k независимо от значения y_j на выходе.

Вероятность получить на выходе значение y_j (независимо от состояния перехода) для инициирующей срабатывание пары (x_i, z_s) будет равна:

$$\mathbf{P}_{(x_i, z_s)}^{y_j} = \sum_{k=1}^K p_{is}^{kj}, j = \overline{1, J}; \forall (x_i, z_s) \in G.$$

Суммируются все вероятности возможных переходов при поступлении знака x_i из текущего состояния z_s в те состояния z_k , в которых на выходе получается заданное значение y_j .

Последняя характеристика - $\mathbf{P}_{(x, z)}^{(z_s, y_j)}(n)$, вычисляется во времени рекуррентно для заданного начального значения пары (x, z) и заданного момента времени n .

Легко видеть, что детерминированный конечный автомат является частным случаем P -автомата. Стохастический автомат становится детерминированным, если в таблице B для входной пары (x_i, z_s) все вероятности p_{is}^{kj} закона распределения имеют значения 0, кроме одной, равной 1. Тогда представленные в B вероятностные отображения вырождаются в функции, и такой вероятностный автомат вырождается в конечный детерминированный автомат:

$$P = \langle X, Y, Z, B \rangle \rightarrow F = \langle X, Y, Z, \varphi, \psi \rangle.$$

Замечание. При определении P -автомата по умолчанию принято, что законы распределения вероятностей переходов/выходов автомата не зависят от времени (номера такта срабатывания). Однако, теоретически можно допускать, что такая зависимость имеет место, т.е. $p_{is}^{kj} = p_{(x_i, z_s)}^{(z_k, y_j)}(t_i)$. Если такая зависимость от времени допускается, то это означает, что срабатывание автомата на очередном такте будет определяться зависящими от текущего времени законами распределения вероятностей переходов/выходов автомата, отражающими предысторию функционирования и попадания автомата в это состояние. Это практически делает невозможным получение характеристик автомата аналитическими методами и требует применение методов имитационного моделирования.

Классы стохастических автоматов

Среди стохастических автоматов выделяют классы P -автоматов, которые имеют специфические особенности интерпретации вероятностного поведения автомата. Для этих автоматов полагают, что их законы распределения вероятностей переходов/выходов не зависят от времени (т.е. от предыстории попадания автомата в текущее состояние). К ним относятся следующие P -автоматы:

- P -автомат Мили;
- P -автомат Мура;
- частично-детерминированные автоматы.

Стохастический автомат Мили

Рассмотрим P -автомат, у которого каждой паре $(x_i, z_s) \in G$ соответствуют отдельные законы распределения вероятностей переходов и выходов, и они задаются в виде двух отдельных таблиц: таблицы вероятностей переходов в состояния множества Z и таблицы вероятностей выходных значений множества Y :

Таблица переходов

$z_k \in Z$	z_1	z_2	...	z_{K-1}	z_K
$(x_i, z_s) \in G$	r_{is}^1	r_{is}^2	...	r_{is}^{K-1}	r_{is}^K

$$i=1,2,\dots,I;$$

$$s=1,2,\dots,K;$$

Таблица выходов

$y_j \in Y$	y_1	y_2	...	y_{J-1}	y_J
$(x_i, z_s) \in G$	q_{is}^1	q_{is}^2	...	q_{is}^{J-1}	q_{is}^J

При этом выполняются условия:

$$\sum_{k=1}^K r_{is}^k = 1, \forall (x_i, z_s) \in G;$$

$$\sum_{j=1}^J q_{is}^j = 1, \forall (x_i, z_s) \in G;$$

где r_{is}^k – вероятность перехода P -автомата в состояние z_k , и q_{is}^j – вероятность появления на выходе P -автомата значения y_j при условии, что на вход P -автомата в начале такта поступило входное значение x_i в состоянии z_s .

Дискретно-событийные стохастические модели (P -схемы)

Если ещё положить, что выполняется условие *независимости* событий перехода P -автомата в новое состояние z_k и выдачи выходного значения y_j для всех k и j , т.е. имеет место соотношение:

$$b_{is}^{kj} = r_{is}^k \cdot q_{is}^j, \forall (x_i, z_s) \in G;$$

то такие P -автоматы составляют класс стохастических автоматов Мили. Очевидно, что если вероятности r_{is}^k и q_{is}^j имеют значения только 1 или 0, то такой стохастический автомат вырождается в детерминированный конечный автомат Мили как частный случай стохастического автомата Мили.

Стохастический автомат Мура

Рассмотрим теперь P -автомат, представленный не связанными таблицами переходов/выходов вида:

$z_k \in Z$	z_1	z_2	...	z_{K-1}	z_K
$(x_i, z_s) \in G$	r_{is}^1	r_{is}^2	...	r_{is}^{K-1}	r_{is}^K

$$i=1,2,\dots,I;$$

$$s=1,2,\dots,K;$$

$y_j \in Y$	y_1	y_2	...	y_{J-1}	y_J
$z_k \in Z$	q_k^1	q_k^2	...	q_k^{J-1}	q_k^J

То есть, появление выходного значения y_j зависит лишь от состояния z_k , в котором автомат оказывается в текущем такте срабатывания, и каждому состоянию $z_k \in Z$ соответствует отдельное распределение вероятностей выходных значений $y_j \in Y$, где q_k^j - вероятность появления выходного значения y_j , если P -автомат оказался в состоянии z_k . При этом:

$$\sum_{j=1}^J q_k^j = 1, \forall k = \overline{1, K}.$$

Если положить, что для всех k и j выполняется условие *независимости* событий перехода P -автомата в новое состояние z_k и события выдачи выходного значения y_j , т.е. имеет место соотношение:

$$b_{is}^{kj} = r_{is}^k \cdot q_k^j, \forall (x_i, z_s) \in G;$$

то такой P -автомат называется стохастическим автоматом Мура. В случае если только одна из вероятностей r_{is}^k и вероятностей q_{is}^j имеет значение 1, а остальные 0, то такой вероятностный автомат вырождается в детерминированный конечный автомат Мура как частный случай стохастического автомата Мура.

Частично-детерминированные стохастические автоматы

Рассмотрим класс P -автоматов, у которых парам $(x_i, z_s) \in G$ соответствуют законы распределения вероятностей переходов/выходов, заданные отдельно на множестве Z и на множестве Y так, что их можно представить соответственно в виде двух таблиц, отдельно для состояний и выходных значений:

Таблица переходов

$z_k \in Z$	z_1	z_2	\dots	z_{K-1}	z_K
$(x_i, z_s) \in G$	r_{is}^1	r_{is}^2	\dots	r_{is}^{K-1}	r_{is}^K

Таблица выходов

$y_j \in Y$	y_1	y_2	\dots	y_{J-1}	y_J
$(x_i, z_s) \in G$	q_{is}^1	q_{is}^2	\dots	q_{is}^{J-1}	q_{is}^J

$$i=1,2,\dots,I;$$

$$s=1,2,\dots,K.$$

При этом выполняются условия:

$$\sum_{k=1}^K r_{is}^k = 1, \forall (x_i, z_s) \in G, \quad i=1,2,\dots,I;$$

$$\sum_{j=1}^J q_{is}^j = 1, \forall (x_i, z_s) \in G, \quad s=1,2,\dots,K.$$

где r_{is}^k – вероятность перехода P -автомата в состояние z_k , и q_{is}^j – вероятность появления на выходе P -автомата значения y_j при условии, что на вход P -автомата в начале такта поступило входное значение x_i в состоянии z_s . При этом для P -автомата не устанавливается условие независимости события перехода в новое состояние z_k и события выдачи выходного значения y_j , как для стохастического автомата Мили. Очевидно, класс автоматов Мили является подклассом класса таких P -автоматов.

В рамках введённого выше в рассмотрение класса P -автоматов выделим подкласс так называемых частично-детерминированных стохастических

автоматов, состоящий из Y -детерминированных и Z -детерминированных стохастических автоматов.

P -автомат называется Y -детерминированным стохастическим автоматом, если для любой пары $(x_i, z_s) \in G$ соответствующие вероятности q_{is}^j таблицы выходов имеют значение 0, и только одно 1, т.е. выходное значение является детерминированным.

P -автомат называется Z -детерминированным стохастическим P -автоматом, если для любой пары $(x_i, z_s) \in G$ соответствующие вероятности переходов в новое состояние r_{is}^k имеют значение 0 и только одно - значение 1, т.е. переход в новое состояние является детерминированным.

Y_0 -детерминированный автомат

Среди Y -детерминированных P -автоматов выделяют простейший вариант автомата, когда на его вход поступает одно единственное значение, т.е. множество входных знаков состоит из одного символа - $X = \{x\}$. В этом случае значением символа можно вообще пренебречь, так как функции перехода и выходов практически не зависят от значения входного символа, а существенным является само *событие* его поступления, инициирующее срабатывание автомата. Такой Y -детерминированный автомат называют Y_0 -детерминированным.

В общем случае для каждого состояния Y_0 -детерминированного автомата закон распределения вероятностей возможных переходов из текущего состояния в новое состояние может зависеть от предыстории его попадания в текущее состояние, т.е. может во времени меняться. Если положить для, что в любом такте срабатывания Y_0 -детерминированного автомата его вероятности переходов для всех состояний остаются детерминированными (не зависят от предыстории попадания автомата в текущее состояние), то такой Y_0 -детерминированный

Дискретно-событийные стохастические модели (P -схемы)

автомат называют марковским, а формируемая во времени последовательность сменяемых состояний называется *дискретной марковской цепью*¹.

Модель и определение характеристик дискретной марковской цепи

Модель марковского Y_0 -детерминированного автомата представляется для каждого состояния $z_k \in Z$ таблицей вероятностей переходов и таблицей выходов вида:

Таблица вероятностей переходов

z_k	z_1	z_2	...	z_{K-1}	z_K	$k=1,2,\dots,K;$
z_s	p_s^1	p_s^2	...	p_s^{K-1}	p_s^K	$s=1,2,\dots,K.$

Таблица выходов

z_k	z_1	z_2	...	z_{K-1}	z_K	$j=1,2,\dots,K;$
y_j	y_1	y_2	...	y_{K-1}	y_K	

$$\sum_{k=1}^K p_s^k = 1, \quad s = 1, 2, \dots, K.$$

Таблица переходов является квадратной матрицей размерности $K \times K$, которую называют матрицей переходных вероятностей или просто матрицей переходов марковской цепи:

$$P = \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1K} \\ p_{21} & p_{22} & \dots & p_{2K} \\ \dots & \dots & \dots & \dots \\ p_{K1} & p_{K2} & \dots & p_{KK} \end{bmatrix}.$$

Для полного описания Y_0 -детерминированного автомата марковской цепи необходимо ещё задать начальное распределение вероятностей нахождения автомата в нулевом такте $l = 0$ в некотором исходном состоянии:

z_k	z_1	z_2	...	z_{K-1}	z_K
$P(0)$	$P_1(0)$	$P_2(0)$...	$P_{K-1}(0)$	$P_K(0)$

$$\sum_{i=1}^K P_i(0) = 1.$$

¹ Марков Андрей Андреевич (1856-1922) русский математик, специалист по теории чисел, теории вероятностей и математическому анализу.

Характеристиками марковской цепи являются вероятности попадания марковского Y_0 -детерминированного автомата в то или иное состояние z_k на l -ом такте его срабатывания: $\mathbf{P}(l) = (P_1(l), P_2(l), \dots, P_{K-1}(l), P_K(l))$. Эти вероятности вычисляются последовательно, начиная с первого такта. Имея вероятности состояний автомата в произвольном l -том такте - $\mathbf{P}(l)$, и используя заданную матрицу вероятностей переходов P , можно вычислить вероятности состояний марковской цепи в следующем $(l + 1)$ -ом такте. Действуя таким образом, можно последовательно вычислить вероятности $P_k(l)$ попадания автомата в состояние z_k для любого l -ого такта срабатывания.

Предварительно расширим матрицу переходных вероятностей P , включив в неё заданное начальное распределение вероятностей нахождения автомата в первом такте $l = 1$ в некотором исходном состоянии. Для этого введём фиктивное состояние z_0 , в котором автомат изначально находится с вероятностью $P_0(0) = 1$. Тогда заданные вероятности начального состояния марковской цепи представляются вероятностями перехода марковской цепи в первом такте в эти состояния из фиктивного состояния z_0 и внесём их в качестве 0-ой строки в матрицу переходных вероятностей P , увеличив её размерность до $(K+1) \times (K+1)$. Теперь первая строка матрицы P , определяющая вероятности перехода после первого такта срабатывания из фиктивного состояния z_0 в реальные исходные состояния, будет иметь вид $(0, p_{01}, p_{02}, \dots, p_{0k}, \dots, p_{0K})$, где $p_{0k} = P_k(1)$, а элементами нулевого столбца будут только нули - $p_{k0} = 0$, (последующие переходы из любого реального состояния в фиктивное состояние z_0 не имеют смысла).

Определение марковского Y_0 -детерминированного автомата удобно представлять в форме размеченного ориентированного графа, вершины которого сопоставляются состояниям автомата, а ориентированные дуги — возможным переходам из одного состояния в другое. Дуги имеют веса, соответствующие

Дискретно-событийные стохастические модели (P -схемы)

вероятностям перехода p_{ij} из состояния z_i в состояние z_j , а вершины графа z_j помечаются выходными значениями y_j , индуцируемые² этими состояниями.

Имея в распоряжении размеченный граф состояний и переходов и зная начальное состояние автомата, можно найти вероятности оказаться автомату в том или ином состоянии $P_1(l), P_2(l), \dots, P_n(l)$ после любого (l -го) такта срабатывания автомата марковской цепи. Они находятся с помощью следующих рекуррентных соотношений:

$$P_i(l) = \sum_{j=1}^K P_j(l-1)p_{ji}; i = 1, 2, \dots, K; l = 1, 2, \dots$$

Это соотношение следует из того, что для дискретной марковской цепи вероятность оказаться на l -ом шаге в состоянии z_i складывается из вероятностей того, что будучи на предыдущем шаге ($l-1$) в состоянии z_j , на следующем шаге с вероятностью p_{ji} произойдёт переход автомата в состояние z_i .

Пример.

Пусть марковский Y_0 -детерминированный автомат задан следующей расширенной матрицей переходов:

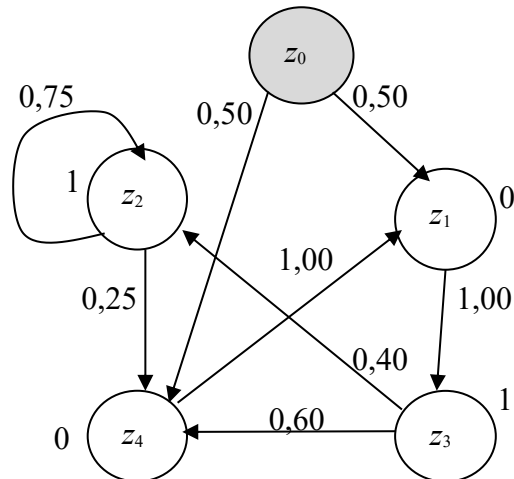
$$P = \begin{bmatrix} 0 & 0.50 & 0 & 0 & 0.50 \\ 0 & 0 & 0 & 1.00 & 0 \\ 0 & 0 & 0.75 & 0 & 0.25 \\ 0 & 0 & 0.40 & 0 & 0.60 \\ 0 & 1.00 & 0 & 0 & 0 \end{bmatrix};$$

и таблицей выходов, с пустым выходом $y = \emptyset$ для фиктивного состояния z_0 :

z	z_0	z_1	z_2	z_3	z_4
y	\emptyset	0	1	1	0

Построим для этого автомата граф переходов/выходов (рисунок 1):

² Индуцировать — воспроизводить.

Дискретно-событийные стохастические модели (P -схемы)Рисунок 1 Граф марковского Y_0 -детерминированного автомата

Найдём вероятности оказаться автомату на 1-ом шаге срабатывания в том или ином состоянии, если в начальный момент ($l=0$) автомат с вероятностью 1 находится в фиктивном состоянии z_0 , из которого он с вероятностью $p_{01} = 0.5$ может перейти в состояние z_1 или с вероятностью $p_{04} = 0.5$ - в состояние z_4 :

$$P_1(1)=0.5, P_2(1)=0, P_3(1)=0, P_4(1)=0.5, \quad l=1;$$

$$P_1(2)=0.5, P_2(2)=0, P_3(2)=0.5, P_4(2)=0, \quad l=2;$$

$$P_1(3)=0, P_2(3)=0.2, P_3(3)=0.5, P_4(3)=0.3, \quad l=3;$$

$$P_1(4)=0.3, P_2(4)=0.35, P_3(4)=0, P_4(4)=0.35, \quad l=4;$$

$$P_1(5)=0.35, P_2(5)=0.2625, P_3(5)=0.3, P_4(5)=0.0875, \quad l=5;$$

...

В 0-ом шаге автомат находится в состоянии z_0 с вероятностью $P_0(0)=1$ и с вероятностью $p_{01} = 0.5$ переходит из состояния z_0 в состояние z_1 либо с вероятностью $p_{04} = 0.5$ - в состоянии z_4 . Поэтому $P_1(1)=P_0(0) \cdot p_{01}=1 \cdot 0.5=0.5$. Аналогично - $P_4(1)=P_0(0) \cdot p_{04}=1 \cdot 0.5=0.5$. Вероятность оказаться на 1-ом шаге в любом другом состоянии равна 0, так как переходы в них из фиктивного состояния z_0 отсутствуют.

Найдём теперь вероятности оказаться автомату в том или ином состоянии на 2-ом шаге срабатывания. Вероятность $P_1(2)$ оказаться автомату на 2-ом шаге в состоянии z_1 зависит только от вероятности $P_4(1)=0.5$ нахождения автомата на 1-ом шаге в состоянии z_4 , и от вероятности $p_{41}=1$ перехода на 2-ом шаге с из состояния z_4 в состояние z_1 . Тогда вероятность $P_1(2)=0.5 \cdot 1=0.5$.

Вероятность оказаться автомату на 2-ом шаге в состоянии z_2 равна нулю: $P_2(2)=0$, так как согласно графу отсутствуют переходы в состояние z_2 из состояний z_1 или z_4 , а вероятности оказаться после первого шага в состояниях z_2 или z_3 , из которых возможен переход в состояние z_2 , равны нулю: $P_2(1)=0, P_3(1)=0$.

Дискретно-событийные стохастические модели (*P*-схемы)

Определим теперь вероятность оказаться автомату на 3-ем шаге в состоянии z_3 , т.е. - $P_3(3)$.

Так как переход в это состояние возможен из состояния z_1 с вероятностью $p_{13}=1$, а вероятность оказаться автомату на 2-ом шаге в состоянии z_1 равна $P_1(2)=0.5$, следовательно $P_3(3)=P_1(2) \cdot p_{13}=0.5$.

Действуя аналогично, можно последовательно получить вероятности оказаться автомату в том или ином состоянии для любого шага срабатывания автомата.

Финальные вероятности дискретной марковской цепи

Из примера видно, что вероятность нахождения автомата в некотором заданном состоянии с каждым шагом меняется. При этом интерес представляет, что будет происходить с вероятностями состояний при $l \rightarrow \infty$? Можно, например, предположить, что со временем значения вероятностей стабилизируются, перестанут меняться и автомат перейдет в стационарное состояние. То есть, вероятности $P_1(l), P_2(l), \dots, P_n(l), l=0,1,2,3, \dots$; будут стремиться к неким пределам при $l \rightarrow \infty$:

$$\lim_{l \rightarrow \infty} P_i(l) = C_i, i = \overline{1, K}.$$

Если пределы C_i существуют и *не зависят от начального состояния* системы, то они называются *финальными вероятностями* состояний. Финальные вероятности состояний – это уже не переменные величины (функции времени), а константы. Наличие финальных вероятностей является важной характеристикой стохастического автомата. Они характеризуют возможность через достаточно продолжительный период времени застать автомат на произвольном шаге в том или ином состоянии с вероятностью, уже не зависящей от значения текущего шага. Заметим, что финальная вероятность состояния z_i – это, по существу, *средняя доля времени* пребывания автомата в этом состоянии после длительного периода его функционирования, принятого за единицу.

Очевидно, что если финальные вероятности существуют, то их можно найти из системы уравнений:

$$C_i = \sum_{j=1}^K C_j p_{ji}, i = 1, 2, \dots, K.$$

Заметим, что эти уравнения не имеют свободного члена, следовательно являются однородными³. Система однородных уравнений всегда совместна, но не является замкнутой. Для получения замкнутой системы уравнений, следует воспользоваться нормировочным условием, заменив им в системе любое из уравнений:

$$\sum_{i=1}^K C_i = 1.$$

Пример.

Найдём финальные вероятности для рассмотренного выше марковского Y_0 -детерминированного автомата (рисунок 1). Соответствующая ему система уравнений для нахождения финальных вероятностей примет вид:

$$\begin{cases} C_1 = C_4; \\ C_2 = 0.75C_2 + 0.40C_3; \\ C_3 = C_1; \\ C_4 = 0.25C_2 + 0.60C_3; \end{cases}$$

Заменим второе уравнение (как одно из "громоздких") нормировочным условием $C_1 + C_2 + C_3 + C_4 = 1$. В результате система уравнений примет вид:

$$\begin{cases} C_1 = C_4; \\ C_1 + C_2 + C_3 + C_4 = 1; \\ C_3 = C_1; \\ C_4 = 0.25C_2 + 0.60C_3; \end{cases}$$

В качестве решения получим:

$$C_1=5/23=0.2174, C_2=8/23=0.3478, C_3=5/23=0.2174, C_4=5/23=0.2174.$$

Полученные финальные вероятности показывают, что в стационарном режиме автомат в среднем 21.74% времени будет находиться в каждом из состояний z_1, z_3 или z_4 , а в состоянии z_2 –

³ Система линейных уравнений называется *однородной*, если во всех её уравнениях свободные члены равны нулю. Однородная система уравнений всегда совместна и имеет тривиальное решение – все неизвестные равны нулю.

34.78 %. Кроме того, знание финальных вероятностей позволяет оценить вероятность появления на выходе автомата заданных возможных значений. Например, вероятность появления на выходе автомата в стационарном режиме в любой момент времени единицы будет равна $C_2 + C_3 = 0.5652$, а нуля – 0.4348

Применение стохастических автоматов для прототипирования программ

Прототипирование и программной логики

При проектировании и разработке сложной программы алгоритмическая структура её функции `main()` как правило формируется из иерархического множества функциональных модулей $\{M_1, M_2, \dots, M_N\}$, последовательно вызываемых операторами условного перехода $if(P(\dots))$ в зависимости от значения вычисленного оператором предиката $P(\dots)$ на наборе параметров (переменных), принадлежащих множеству глобальных переменных программы - $\{p_1, p_2, \dots, p_K\}$. Если положить, что глобальные переменные $\{p_1, p_2, \dots, p_K\}$ в программе определены, то прототип функции `main()` можно "собрать" из "прототипов" модулей \bar{M}_i с соответствующей сигнатурой, имитирующих работу функциональных модулей $M_i \in \{M_1, M_2, \dots, M_N\}$. Выполнение прототипа программы будет представлять собой последовательные переходы в соответствии с вычисленными операторами условного перехода $if(P(\dots))$ значениями предикатов $P(\dots)$ от одного прототипа программного модуля \bar{M}_i к другому \bar{M}_j . Прототип имитирует работу реального программного модуля путём генерации в качестве результата выполнения "практически допустимых" случайных значений своих выходных переменных на множестве $\{p_1, p_2, \dots, p_K\}$. Переход программы из завершившегося программного модуля \bar{M}_i в последующий программный модуль \bar{M}_j регулируется операторами условного перехода $if(P(\dots))$, вычисляющим реальные предикаты $P(\dots)$ на основе текущих значений глобальных переменных программы $\{p_1, p_2, \dots, p_K\}$. В итоге выполнение прототипа программы выглядит как случайная траектория последовательного прохождения вызываемых программных модулей.

Теоретически прототип программы является вероятностным конечным автоматом, текущее состояние которого ассоциируется с вызванным функциональным модулем $\bar{M}_i \in \{\bar{M}_1, \bar{M}_2, \dots, \bar{M}_N\}$. При этом входные знаки ассоциируются со значениями глобальных переменных $\{p_1, p_2, \dots, p_K\}$, полученных текущим модулем \bar{M}_i до вычисления предиката $P(\dots)$ текущего оператора условного перехода $if(P(\dots))$, а выходные знаки ассоциируются с значениями набора глобальных переменных $\{p_1, p_2, \dots, p_K\}$, полученных в результате выполнения очередного условно вызванного модуля \bar{M}_j . Очередной такт срабатывания вероятностного конечного автомата начинается по событию завершения текущего ранее вызванного модуля.

Целью прототипирования является априорное выявление на прототипе программы потенциальных ошибок в логике управления функциональными модулями программы. Выполнение такого прототипа программы на случайно формируемом наборе значений глобальных переменных $\{p_1, p_2, \dots, p_K\}$ позволяет, например, априори выявлять потенциальные ошибки в программной реализации предикатов, когда программа формально должна, но никогда не попадает в те или иные модули, а также выявление потенциальной возможности возникновения исключительных ситуаций в реальных модулях M_i , заменивших в программе соответствующие им прототипы \bar{M}_i . В качестве ошибок могут быть выявлены деление на ноль, исчезновение или переполнение порядка, выход за пределы массива и т.п.

Тестирование точек завершения программы

Одной из задач прототипирования разрабатываемой программы может быть анализ точек её завершения. Если время выполнения программы конечно, то точки завершения выполнения прототипа на множестве случайно формируемых значений глобальных переменных программы должны быть достижимы. Теоретически это означает, что реализация прототипа программы в виде вероятностного конечного P -автомата должна представить её выполнение в виде *поглощающей* дискретной марковской цепи из последовательности выполненных

Дискретно-событийные стохастические модели (P -схемы)

программных модулей, играющих роль состояний P -автомата. Предикаты управления переходами от текущего модуля к последующему модулю вычисляются на случайных значениях выходных данных, генерируемых на выходе каждого модуля независимо от предшествующей траектории выполнения прототипа программы и попадания в текущий модуль. Поэтому зависимость вероятностей перехода от предыстории отсутствует. Прототип программы в виде поглощающей марковской цепи характеризуются тем, что они содержат модули - состояния z_i , для которых теоретически вероятность перехода $p_{ii} = 1$ (т.е. с вероятностью равной 1 осуществляется переход в то же самое состояние). Практически это означает, что прототип через конечный период времени должен попасть в модуль, в котором осуществляется завершение программы.

Профилирование программы

Поглощающие дискретные марковские цепи используются также для прототипирования программ с целью анализа временных характеристик их выполнения. Состояния цепи, при моделировании, отождествляются с модулями программы, а матрица переходных вероятностей определяет тенденции переходов между модулями, отражающие семантику программы, но не учитывающие влияние реальных данных на развитие вычислительного процесса (это конечно существенное допущение). Представление прототипа программы поглощающей цепью позволяет аналитически оценивать долю обращений к тем или иным модулям при выполнении программы до момента её завершения. Если ввести в модель оценку среднего времени выполнения модулей, то в итоге можно априори оценивать среднее время выполнения программы и вклад в него каждого модуля. Меняя структуру и логику выполнения программы, а как следствие и матрицу переходных вероятностей, можно пытаться оптимизировать количество и трудоёмкость модулей программы для ускорения выполнения критических участков программы. Подобный метод называется *профилированием кода*.

Профилирование кода обычно проводится для программ, которые значительную долю времени тратят на обращения к различным вычислительным

Дискретно-событийные стохастические модели (*P*-схемы)

ресурсам системы: файлам, памяти, периферийным устройствам, каналам передачи данных. Если представить такую программу поглощающей марковской цепью, состояния которой отражают обращение к ресурсам системы, а переходные вероятности выражают интенсивность обращения к таким ресурсам, то можно исследовать наличие «узких мест» и их влияние на время выполнения вычислительного процесса и принятие решений по модификации структуры программы.

Тестирование исключительных состояний

При разработке и анализе прототипа программной системы реального времени важно убедиться, что ни при каких обстоятельствах в программной системе не возникают заикливания или фатальные завершения. Одновременно с этим важно убедиться, что все модули потенциально достигаемы при выполнении прототипа. Такой взгляд на выполнение программы ассоциирует её с так называемой *неприводимой* марковской цепью. Дискретная марковская цепь называется неприводимой, если любое состояние z_i может быть достигнуто из любого другого состояния z_j за конечное число переходов. В этом случае все состояния цепи называются *сообщающимися*. Процесс, порождаемый неприводимой цепью, начавшись в некотором состоянии, никогда не завершается, а последовательно переходит из одного состояния в другое, попадая в различные состояния с разной частотой. Основная характеристика неприводимой цепи – доля времени (финальные вероятности) пребывания процесса в состояниях $z_i, i = 1, \dots, K$. Анализ этих вероятностей позволяет априори выявлять в прототипе подозрительно избыточно или недостаточно загруженные модули и изменять структуру или логику работы программы.