

Раскрашенная сеть Петри CPN- это кортеж

$CPN = \{\Theta, \Sigma, P, T, A, N, C, G, E, I\}$.

(2.13)

Рассмотрим элементы определения (2.13).

1. $\Theta = \{\theta = 0, 1, 2, \dots\}$ – множество дискретных моментов времени (шагов), в которые происходит функционирование сети. Номер шага есть переменная целого типа:

var teta: Integer;

2. Σ – конечное множество непустых типов, называемое цветами. Типы имеют общее название color и задаются аналогично тому, как это принято в языках программирования.

Например:

type colorI = Integer; - целый тип;

*type colorP = product U * I;* - прямое произведение типов.

При описании переменных, используемых в сети, указывают их принадлежность к типу, например:

Var x:colorU; vari:colorI; varj:colorP.

В последнем примере переменная j состоит из пары элементов $j=(jU, jI)$, где элемент jU принадлежит типу colorU, а jI – типу colorI.

3. P – конечное множество позиций. Это множество может быть представлено перечисляемым типом:

Type P=(p1,p2,...,pn);

а для работы с ним вводится переменная

var p : P.

С каждой позицией может быть связана определенная маркировка, которая учитывает наличие различных типов ресурсов. Маркировка позиции $p \in P$ представляет собой мультимножество, например, следующего вида:

$m(p)=(I^x, 2^i, I^j)$.

Здесь x, i, j - переменные указанных в п. 2 цветовых типов, определяющие различные виды ресурсов, а цифры , стоящие перед кавычками – количество фишек соответствующего типа в позиции $p \in P$.

Совокупность маркировок всех позиций называется маркировкой сети.

$M = \{m(p_1), m(p_2), \dots, m(p_n)\}$.

В языке Паскаль мультимножество, определяющее маркировку позиции, может быть представлено, например типом - *множество*, состоящим из записей вида

type m — record

i: Integer; x:colorU end;

Операции над мультимножествами можно определить соответствующими процедурами. Аналогично определяется мультимножество, задающее маркировку сети.

4. T - конечное множество переходов, которое представить перечисляемым типом

type T - (t1,t2,...,tm);

и соответствующей переменной:

var t:T.

Это множество не отличается от множества переходов для обыкновенных сетей Петри, однако правила их срабатывания могут быть более сложными. Эти правила рассмотрены ниже.

5. A - конечное множество дуг, связывающих между собой позиции и переходы. В отличие от обыкновенных сетей Петри, где дуги задаются матрицами инцидентности F^p и F^t , а на языке CPN ML указываются все дуги с помощью выражения вида $p \text{ to } t$ и $t \text{ to } p$. где $p \in P$, $t \in T$. Такая нотация введена для того, чтобы можно было пару элементов p_i, t_j соединить несколькими дугами с различными свойствами.

В этих выражениях первый элемент указывает начало дуги, второй – конец дуги. Множество A имеет следующую структуру:

$$A = \{a_1, a_2, \dots, a_p\},$$

где дуги $a_k \in A$ определяются одним из видов выражений

$$a_k = p_i \text{ to } t_j \text{ или } a_k = t_j \text{ to } p_i, p_i \in P, t_j \in T,$$

Поскольку дуги имеют два вида: от позиции к переходу ($p \text{ to } t$) и от перехода к позиции ($t \text{ to } p$), то множество A можно представить в виде суммы непересекающихся множеств:

$$A = A^p \cup A^t, A^p \cap A^t = \emptyset,$$

Где множество A^p содержит элементы вида $p \text{ to } t$, а множество A^t – элементы вида $t \text{ to } p$.

Как и в случае обыкновенных сетей Петри, элементы множеств P, T, A не пересекаются:

$$P \cap T = \emptyset, P \cap A = \emptyset, T \cap A = \emptyset.$$

6. $N(a)$ – узловая функция, которая для каждой дуги $a \in A$ указывает ее исходный и конечный узел. Формально соответствует выражение $s \text{ to } d$. Здесь s – имя узла, от которого начинается дуга, d – имя узла, где она заканчивается. Рассмотрим один из способов задания этой функции. Каждая дуга может быть представлена в виде записи. Дуга типа $p_i \text{ to } t_j$ задаются записью

Type AP= record
(2.14)

p:P; t:T end;

Дуги типа $t_j \text{ to } p_i$ задаются аналогично:

Type AT=record
(2.15)

t:T p:P; end;

Множества дуг A^p и A^t могут быть представлены перечисляемыми типами:

Type AP=(ap1,ap2,...,apk);
(2.16)

Type AT=(at1,at2,...,atk);

Для работы с дугами вводится переменные соответствующих типов:

Var ap:AP; var at: AT;

7. $C(p)$ – цветовая функция, определяющая множество типовцветов, разрешенных для каждой позиции. Например, запись

$$C(p) = \begin{cases} \text{colorU} & \text{if } p \in \{p_1, p_2, p_4\}, \\ \text{colorI} & \text{otherwise} \end{cases}$$

означает, что цвета типа U разрешены для позиций p_1, p_2, p_4 , а цвета типа I разрешены для всех остальных позиций из P .

8. $G(t)$ – блокировочная (спусковая) функция, описывающая дополнительные условия, которые должны быть выполнены для срабатывания перехода $t \in T$. Эта функция представляет собой предикат, составленный из переменных принадлежащих типов цветов Σ . Например, функции

$$G(t) = \begin{cases} x = q & \text{if } t = t_1, \\ \text{true} & \text{otherwise} \end{cases}$$

Принимает истинное значение для перехода t_i , если переменная $x = q$, а во всех остальных случаях она ложь. Таким образом, для того, чтобы было разрешено срабатывание перехода t_i , требуется выполнение дополнительного условия $x = q$ на входной дуге к t_i . Если эта функция не определена, то по умолчанию предполагается истинной.

9. $E(a)$ – функция, задающая выражения на дугах $a \in A$. Эта функция для каждой дуги a определяет мультимножество, состоящее из элементов, описанных в множестве цветов Σ . Мы будем говорить, что это мультимножество помечает дугу. Введение в CPN функции $E(a)$ является развитием понятия кратности дуг в формализме обыкновенных СП.

Рассмотрим примеры задания функции $E(a)$.

$$E(a_1) = 2'e$$

- дуга a_1 помечается двумя фишками типа e ;

$$E(a_2) = \text{case } x \text{ of } p \Rightarrow 2'e \mid q \Rightarrow 1'e$$

- дуга a_2 помечается двумя фишками типа e , если переменная x – p одной фишкой типа e , если $x = q$.

$$E(a_3) = \text{if } x = p \text{ then } 1'e \text{ else empty}$$

- дуга a_3 помечается одной фишкой типа e , если $x = p$, иначе не помечается.

$$E(a_4) = \text{if } x = q \text{ then } 1'(q?i + 1) \text{ else empty}$$

- дуга a_4 помечается одной фишкой типа P (см. пример в п.2), если $x = q$ и не помечается в противном случае.

Отсутствие выражения для какой-либо дуги означает, что дуга не помечена.

9. $I(p)$ – функция инициализации сети CPN. Эта функция по аналогии с обыкновенными СП задает начальную маркировку (разметку) сети M_0 , т.е. для каждой позиции $p \in P$ указывает цветное мультимножество, обозначаемое $m_0(p)$.

Например, $m_0(p_1) = 3'e$; $m_0(p_2) = 2'(p, 0)$; $m_0(p_3) = \emptyset$ – начальные маркировки позиций; $M_0 = \{3'e; 2'(p, 0); \emptyset\}$ – начальная маркировка сети.

2.2.3. Функционирование CPN

Рассмотрим работу раскрашенных сетей Петри. Как и в случае обыкновенных СП. функционирование сети состоит в срабатывании переходов и происходящих вследствие этого изменениях маркировок.

Мы будем говорить, что переход $t \in T$ может сработать, если выполняется условие

$$\forall p \in P: (m(p) \geq \sum_{a \in AP} E(a)) \wedge (G(t) = \text{true}).$$

В первой скобке выражения (2.17) происходит поэлементное сравнение мультимножества, являющегося маркировкой позиции p с мультимножеством, которым помечены дуги, ведущие от p к t . Суммирование мультимножеств $E(a)$ связано с тем, что, как отмечалось в п. 5, два узла могут быть соединены несколькими дугами, и суммирование производится по всем таким дугам.

Во второй скобке записано условие отсутствия блокировки перехода t (см. п. 8 определения).

Если условие срабатывания перехода t выполняется, то он может сработать, при этом говорят, что выполнится шаг работы CPN. Отсчет шагов производится в дискретном времени $\theta = 0, 1, 2, \dots$. Поэтому маркировку отдельных позиций $m(p)$ и сети в целом M мы будем также привязывать ко времени и записывать соответственно в виде $m(p, \theta)$ и $M(\theta)$.

Изменение маркировки узла $p \in T$ при осуществлении шага работы CPN рассчитывается по формуле

$$m(p, \theta + 1) = m(p, \theta) - \sum_{a \in AP} E(a) + \sum_{a \in AT} E(a). \quad (2.18)$$

В выражении (2.18) все слагаемые являются мультимножествами, и вычисления производятся в соответствии с правилами сложения и вычитания мультимножеств. Знаки суммирования мультимножеств используются в формуле по причине, которая уже пояснена выше: каждая пара узлов может быть связана несколькими дугами, а каждая дуга $a \in A$ помечена отдельным выражением $E\{a\}$.

Если на некотором шаге θ выполняются условия срабатывания нескольких переходов, то по аналогии с обыкновенными СП, в зависимости от задачи исследования CPN, мы можем разрешить сработать всем переходам (при построении дерева возможных маркировок) либо дать такую возможность только одному из них на основе некоторых дополнительных условий (например, приоритетов).

Пример. Приведем описание простой CPN, схема которой и начальная маркировка приведены на рисунке 2.5 а.

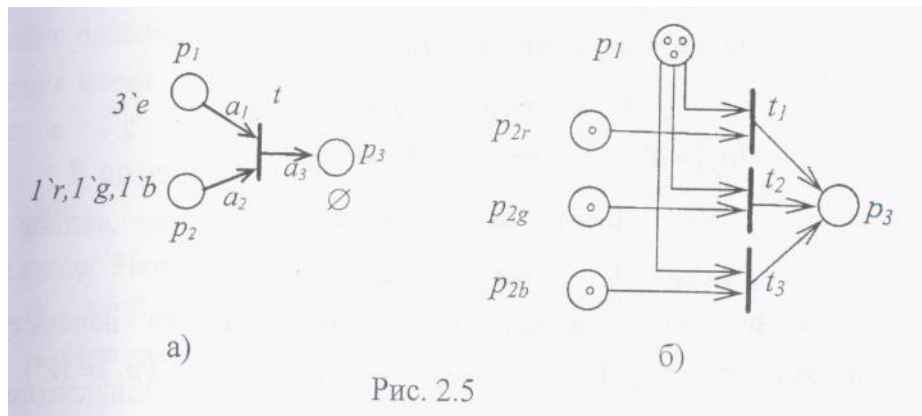


Рис. 2.5

•Множества цветов:

$type\ color\ C = (r, g, b); colorE = (e).$

•Переменные:

$Var\ x: color\ C; Var\ y: colorE.$

•Множество позиций:

$typeP = (p_1, p_2, p_3).$

•Множество переходов:

$typeT = (t).$

•Множество дуг $A = A^p \cup A^t$, $A^p = \{a_1, a_2\}$, $A^t = \{a_3\}$,

$a_1 = p_1 \text{ to } t$, $a_2 = p_2 \text{ to } t$, $a_3 = t \text{ to } p_3$

$type\ AP = (a_1, a_2); type\ AT = (a_3);$

$var\ a_1, a_2: ap; var\ a_3: at;$

•Цветовая функция

$colorC\ if\ p \in \{p_2, p_3\}.$

$C(p) = colorE\ if\ p \in \{p_1\}.$

•Блокировочная функция всегда истинна:

$G\{t\} = true.$

• Функция $E(a)$, задающая выражения на дугах:

$E(a_1) = 1'e;$

$E(a_2) = if\ x = r\ then\ 1'e\ else$
 $if\ x = g\ then\ 1'g\ else\ 1'b;$

$E\{a_3\} = if\ (x = r\ and\ y = e)\ then\ 1'r$
 $else\ if\ [x = g\ and\ y = e]\ then\ 1'g$
 $else\ if\ (x = b\ and\ y = e)\ then\ 1'b.$

• Функция инициализации $I(p)$, задающая начальную маркировку $m_0(p_1) = (3'e);$
 $m_0(p_2) = (1'r, 1'g, 1'b); m_0(p_3) = (\emptyset).$

Ниже $\theta=0$	приведено $\theta=1$	полное $\theta=2$	дерево $\theta=3$	маркировок $\theta=3$
$M_0 = \{3'e; 1'r; 1'g; 1'b; \emptyset\}$				
	$t:x = r, y = e$	$\{2'e; 1'g; 1'b; 1'r\}$		
		$t:x = g, y = e$	$\{1'e; 1'b; 1'g; 1'r\}$	
			$t:x = b, y = e$	$\{\emptyset; \emptyset; 1'r; 1'g; 1'b\}$
		$t:x = b, y = e$	$\{1'e; 1'g; 1'r; 1'b\}$	
			$t:x = g, y = e$	$\{\emptyset; \emptyset; 1'r; 1'b; 1'g\}$
	$t:x = g, y = e$	$\{2'e; 1'r; 1'b; 1'g\}$		
		$t:x = r, y = e$	$\{1'e; 1'b; 1'g; 1'r\}$	
			$t:x = b, y = e$	$\{\emptyset; \emptyset; 1'g; 1'r; 1'b\}$
		$t:x = b, y = e$	$\{1'e; 1'r; 1'g; 1'b\}$	
			$t:x = r, y = e$	$\{\emptyset; \emptyset; 1'g; 1'b; 1'r\}$
	$t:x = b, y = e$	$\{2'e; 1'r; 1'g; 1'b\}$		
		$t:x = r, y = e$	$\{1'e; 1'g; 1'b; 1'r\}$	
			$t:x = g, y = e$	$\{\emptyset; \emptyset; 1'b; 1'r; 1'g\}$
		$t:x = g, y = e$	$\{1'e; 1'r; 1'b; 1'g\}$	
			$t:x = r, y = e$	$\{\emptyset; \emptyset; 1'b; 1'g; 1'r\}$

Поясним описание сети. Для позиции p_1 разрешен цвет e , для позиций p_2 , p_2 разрешены цвета r, g, b . Переход t может сработать, если в позиции p , находится хотя бы одна фишка цвета e и в позиции p_2 находится хотя бы одна фишка цветов r, g, b .

В позицию p_3 при срабатывании перехода пересылается та фишка, которая была извлечена из p_2 при срабатывании перехода. Начальная маркировка - 3 фишки цвета e в позиции p_1 , по одной фишке всех цветов в позиции p_2 .

При работе сети переменная x принимает поочередно (в произвольном порядке) значения r, g, b , а переменная y - значение e . Работа сети продолжается до тех пор, пока все фишки из p_2 не будут пересланы в p_3 . При этом позиции p_1 и p_2 оказываются пустыми. Дерево маркировок описывает все возможные варианты последовательности пересылки фишек.

Эквивалентная рассмотренной CPN обыкновенная сеть Петри приведена на рисунке 2.5 б.

Построив дерево маркировок этой сети, нетрудно убедиться, что оно структурно совпадает с рассмотренным выше деревом CPN. Однако раскрашенная сеть содержит больше

информации, поскольку ресурсы в позиции p_3 не обезличиваются и можно проследить порядок их поступления.