ВЕСТНИК НАЦИОНАЛЬНОГО ИССЛЕДОВАТЕЛЬСКОГО ЯДЕРНОГО УНИВЕРСИТЕТА “МИФИ”, 2015, том 4, № 5, с. 440–446

КИБЕРНЕТИКА, ИНФОРМАТИКА И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 681.3.001:681.5

СИНХРОНИЗАЦИЯ ПАРАЛЛЕЛЬНЫХ ПРОЦЕССОВ НА ОСНОВЕ СЕТЕЙ ПЕТРИ С УПРАВЛЯЕМЫМИ ПЕРЕХОДАМИ

В. В. Крушный, П. В. Сагайдачная

Снежинский физико-технический институт НИЯУ МИФИ, Челябинская область

E-mail: poli_nochka@mail.ru

Поступила в редакцию 23.04.2015 г.

Рассматриваются формальные модели параллельных процессов на основе математической теории сетей Петри. Предложен другой, более простой, способ привнесения времени в формализм сети на уровне внешнего управления, исключающего нарушения изначально простой, красивой и лаконичной структуры сети Петри.

Ключевые слова: сеть Петри, управляемые переходы, параллельная обработка, вычислительные модели, обмен данными.

DOI: 10.1134/S2304487X15050077

1. ФОРМАЛИЗАЦИЯ СЕТИ ПЕТРИ И ЕЕ ОСНОВНЫЕ СВОЙСТВА

Сеть Петри – это абстрактная, математическая модель информационного потока, введенная Карлом Адамом Петри в 1962 году, широко используется в различных областях вычислительной техники для описания и анализа процессов, события которых определенным образом взаимодействуют друг с другом. Нас сети Петри будут интересовать с точки зрения синхронизации параллельных взаимодействующих процессов в автоматизированных системах с параллельной обработкой данных.

Под сетью Петри понимается четверка $N = (P, T, R, \mu_0)$, где $P = \{p_1, p_2, \dots, p_n\}$ – конечное множество позиций, $T = \{t_1, t_2, \dots, t_m\}$ – конечное множество переходов, R – бинарное отношение на $P \cup T$, соответствующее дугам двудольного графа с множеством вершин $P \cup T$, то есть $R \subseteq P \times T \cup T \times P$. Начальное маркирование μ_0 – это функция μ_0: $P \to \{0, 1, \dots\}$, которая ставит в соответствие каждой позиции определенное число меток, “0” соответствует отсутствию метки в позиции. Если позиции занумеровать, то маркирование можно представить вектором $\mu = \{\mu(p_1), \mu(p_2), \dots, \mu(p_n)\}$, где n число позиций сети, а $\mu(P_i)$ равно значению функции маркирования в i-й позиции.

Сети Петри удобно представлять в матричной форме. Обозначим через $In(p_i, t_j) = \{a_{ij}^-\}$ – матрицу входов сети размерностью $n \times m$, где

$$a_{ij}^- = \begin{cases} 1, & \text{если } a_{ij}^- | p_i R t_j; \\ 0 & \text{в противном случае,} \end{cases}$$

а через $Out(p_i, t_j) = \{a_{ij}^+\}$-матрицу выходов сети размерностью $n \times m$, где

$$a_{ij}^+ = \begin{cases} 1, & \text{если } a_{ij}^+ | t_j R p_i; \\ 0 & \text{в противном случае.} \end{cases}$$

$D = [Out(p_i, t_j) - In(p_i, t_j)]$ – называется характеристической матрицей сети Петри. Переход $t_j \in T$ в маркированной сети Петри с маркировкой μ возбужден, если $\forall\, p_i \in P\, \mu(p_i) \geqslant a_{ij}^-$.

При срабатывании перехода от каждой его входной позиции отнимается одна метка, а к каждой его выходной позиции прибавляется одна метка. Из множества возбужденных переходов сети Петри может сработать любой единственный переход. Переход t_j в сети Петри с маркировкой μ может быть запущен всякий раз, когда он возбужден. В результате запуска образуется новая маркировка μ', определяемая по следующему правилу:

$$\mu'(p_i) = \mu(p_i) - In(p_i, t_j) + Out(p_i, t_j) = \\ = \mu(p_i) - a_{ij}^- + a_{ij}^+.$$

Входы $In(p_i, t_j)$

	p_1	p_2	p_3	p_4	p_5
t_1	1	0	0	0	0
T_2	0	1	1	0	0
T_3	0	0	0	1	0
T_4	0	0	0	0	0

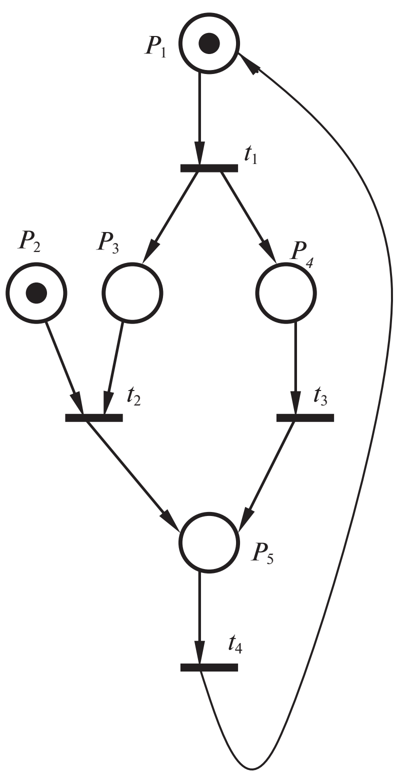

Рис. 1. Сеть Петри N_1.

Выходы $Out(p_i, t_j)$

	p_1	p_2	p_3	p_4	p_5
t_1	0	0	1	1	0
t_2	0	0	0	0	1
t_3	0	0	0	0	1
t_4	1	0	0	0	0

Вектор маркирования μ_0

	p_1	p_2	p_3	p_4	p_5
T_1	1	1	0	0	0

В сети Петри N_1 (рисунок 3.5) переход t_1 возбужден, а t_2, t_3, t_4 не возбуждены. После срабатывания t_1 получим маркировку (0, 1, 1, 1, 0). При этой маркировке возбуждены переходы t_2 и t_3, а t_1 и t_4 не возбуждены. Если сработает t_2, то получим маркировку (0, 0, 0, 1, 1), если сработает t_3, то маркировка перейдет в (0, 1, 1, 0, 1). Таким образом, на i-м шаге при срабатывании t_j одно маркирование переходит в другое,

$$\mu_i \xrightarrow{t_j} \mu_{i+1}.$$

Маркирование μ' называется *достижимым* из μ, если существует последовательность срабатываний $\sigma = t_1, t_2, ..., t_n$, приводящая из μ в μ'. Само μ можно считать достижимым из μ через пустую последовательность срабатываний, хотя возможны и такие ситуации, при которых

$$\mu \xrightarrow{\sigma} \mu, \quad \sigma \neq \varnothing.$$

Функция следующего состояния $\delta(\mu, t_j)$ для сети Петри N с маркировкой μ и переходом $t_j \in T$ определена, если $\mu(p_i) \geqslant a_{ij}^-$. В этом случае применение функции следующего состояния означает:

$$\delta(\mu, t_j) = \mu', \text{ где } \mu' = \mu - a_{ij}^- + a_{ij}^+ \quad \text{для } \forall p_i \in P.$$

Сети Петри, для которых из условия: $\forall p_i \in P$, $\forall t_j \in T$, $In(p_i, t_j) = a_{ij}^- > 0$ вытекает, что $Out(p_i, t_j) = a_{ij}^+ = 0$, полностью определяются характеристической матрицей D. Справедливость этого утверждения очевидна в том случае, если ни одна позиция p_i рассматриваемой сети Петри не является одновременно входом и выходом при одном и том же переходе t_j. В дальнейшем ограничимся рассмотрением сетей Петри без петель и кратных дуг и будем называть такие сети *ординарными*.

Среди ординарных сетей Петри обычно выделяются четыре класса сетей: автоматные сети Петри, маркированные графы, сети Петри со свободным выбором и правильные сети Петри.

В *автоматных сетях* каждый переход может иметь только один вход и один выход, то есть для $\forall t_j \in T\ In(t_j, p_i) = 1$ и $Out(t_j, p_i) = 1$. Некоторые свойства автоматных сетей Петри очевидны. Так автоматные сети – строго сохраняющие. Это означает, что число меток в такой сети никогда не изменяется. Отсюда следует, что дерево достижимости автоматных сетей является всегда конечным и, следовательно, вопросы анализа таких сетей разрешимы. Фактически автоматные сети Петри эквивалентны автоматам, как они определены в теории автоматов и теории формальных языков. Таким образом, эти модели имеют ограниченный интерес из-за ограниченных возможностей моделирования конечных автоматов.

В *маркированных графах* (*MG*) каждая позиция является входом для точно одного перехода и выходом точно для одного перехода, то есть каждая позиция имеет один вход и один выход. В этом случае для $\forall p_i \in P\ In(p_i, t_j) = 1$ и $Out(p_i, t_j) = 1$. Маркированные графы двойственны автоматным сетям Петри с точки зрения моделирования. С помощью *MG* – сети удобно моделировать параллельность и синхронизацию и в том виде, в котором они представлены, не могут моделировать конфликты и принятие решений, зависящих от данных.

Сети Петри свободного выбора или *FC*-сети (*free choice*) – это такие графы, в которых каждая позиция либо имеет один выход, либо является единственным входом в каждый свой переход. Это позволяет свободно осуществлять выбор запускаемого перехода, присутствие меток в других позициях не влияет на выбор запускаемого перехода. Именно для сетей свободного выбора найдены условия живости и безопасности в терминах структурных сетей Петри путем, включающим перебор всех достижимых маркировок. *FC*-сети моделируют довольно широкий класс таких параллельных формализмов, как параллельные граф-схемы алгоритмов.

В *правильных* сетях требуется, чтобы каждый переход имел не более одной входной позиции, которая совместно используется с другим переходом и поэтому служит для ограничения возможностей возникновения конфликтов.

В теории сетей Петри предложено также несколько расширений, ориентированных на увеличение моделирующих возможностей данного аппарата. Одним из расширений являются *обобщенные сети Петри* $N_g = (P, T, F_p, F_t, \mu_0)$, где

$$F_p\colon P \times T \to \omega_p = \{0, 1, 2, \ldots\};$$

$$F_t\colon T \times P \to \omega_t = \{0, 1, 2, \ldots\}.$$

Граф обобщенной сети Петри имеет форму мультиграфа, в котором возможно несколько дуг между позициями и переходами, или же форму взвешенного графа с использованием весов дуг ω_p и ω_t. Такое расширение не увеличивает моделирующих свойств и обобщенные сети принципиально полностью эквивалентны обычным сетям Петри.

Раскрашенные сети Петри также по своим возможностям эквивалентны обычным сетям. Введение в данных сетях еще одного множества – раскрасок (цветов) меток позволяет уменьшить размерность графа при моделировании сложных систем.

Приоритетные сети и *сети с проверкой на нуль* позволяют учитывать в модели приоритетность событий. В сетях с проверкой на нуль для этой цели вводится дополнительное множество Φ_I прямых инцидентных дуг запрета, причем $In(p_i, t_j) \cap \Phi_I = \varnothing$. На графе сети Петри дуги, идущие от позиций к переходам, отображающие множество Φ_I, называются дугами запрета. Для срабатывания некоторого перехода t_j сети Петри с проверкой на нуль требуется кроме наличия меток во всех входных позициях $In(p_i, t_j)$ также и отсутствие меток во входных позициях запрета $\Phi_I(t_j)$ данного перехода.

В приоритетных сетях Петри вводится специальная функция приоритетности, задающая соотношение приоритетов срабатывания для двух подмножеств переходов T_λ и T_0, где T_λ, $T_0 \in T$; $T \cup T_0 = T$; $T_\lambda \cap T_0 = \varnothing$.

При построении моделей очень важным является учет временных характеристик моделируемых событий. Два расширения сетей Петри – *временные сети и сети Мерлина* – позволяют отразить в модели временные параметры системы.

Задание временной сети *Ns* включает 6 множеств:

$Ns = (P, T, R, \mu_0, u, v)$, где $u = (\tau_1, \ldots, \tau_i, \ldots)$ – возрастающая последовательность действительных чисел, называемая временной базой:

$v\colon P \times u \to u$ – функция временных задержек.

Фактор времени учитывается в *Ns* путем введения пассивного состояния метки в позиции. При поступлении метки в позицию p_i она остается в пассивном состоянии (не может участвовать в возбуждении переходов) на время $v(p_i, \tau_s) - \tau_s$ и только после этого переходит в активное состояние.

Сеть Мерлина задается как $N_M = (P, T, R, \mu_0, \lambda^{\min}, \lambda^{\max})$, где $\lambda^{\min} = \{\tau_i^{\min}\}$ – множество времени минимальной задержки для переходов $t_i \in T$; $\lambda^{\max} = \{\tau_i^{\max}\}$ – множество времен максимальной задержки для переходов $t_i \in T$; $\tau_i^{\min}, \tau_i^{\max} \geqslant 0$; $\tau_i^{\min} \leqslant \tau_i^{\max}$. Срабатывание любого перехода t_i сети Мерлина может наступить через время не менее $\tau_i^{\min}$ после его возбуждения и не более $\tau_i^{\max}$ после указанного события.

Временные сети и сети Мерлина не позволяют моделировать операции над данными, выполняемые в системе, не отражают зависимостей процессов обработки от типа или признаков задачи (сообщения). Учет указанных факторов возможен в наиболее мощном расширении сетей Петри – классе так называемых оценочных, или *E*-сетей.

E-сеть задается совокупностью следующих множеств:

$$N_E = (P, P_Y, P_R, T, \mu_0),$$

где P – конечное множество позиций, $P \neq \varnothing$; P_Y – множество периферийных позиций, $P_Y \subset P$; P_R – множество решающих позиций, $P_R \subset P$; T – конечное множество описаний переходов t_i, $T \neq \varnothing$, $t_i = \{s, \tau(t_i), \rho\}$ (здесь s – тип перехода; $\tau(t_i)$ – время перехода; ρ – процедура перехода).

Кроме учета фактора времени отличиями этого класса сетей Петри являются усложнение логики работы перехода, выделение нескольких базовых типов переходов, введение в модель различных операций над метками.

Таким образом, в аппарате сетей Петри можно выделить три класса, обладающих моделирующими свойствами, необходимыми для представления микропроцессорных устройств с параллельной обработкой или их отдельных подсистем. Временные сети могут использоваться в тех случаях, когда возможно предположение о постоянном времени обработки для каждого события в модели. Сети Мерлина позволяют рассматривать случай равномерного распределения времени обслуживания в некотором заданном интервале. Наиболее же общий случай функционирования микропроцессорных устройств может моделироваться классом E-сетей. Введем другой, более простой, способ привнесения времени в формализм сети на уровне внешнего управления, исключающего нарушение изначально простой, красивой и лаконичной структуры сети Петри.

2. КОНЦЕПЦИЯ УПРАВЛЯЕМОГО ПЕРЕХОДА

Обычным образом в определенной сети Петри событие происходит при выполнении всех входных для него условий. Если же при какой-то разметке могут сработать несколько переходов, то решение о том, какие из них и в какой последовательности в действительности срабатывают, должно приниматься вне формализма сети.

Таково первое затруднение, встречающееся на пути практических приложений теории сетей Петри. Второе затруднение связано с тем фактом, что описываемые этими сетями процессы по существу асинхронны. Изменение разметки сети происходит в результате завершения какого-либо процесса, то есть в совершенно конкретный момент времени, однако время в сетевой модели никак не фигурирует. Это обстоятельство, изначально присущее сетям Петри, для определенного класса моделируемых процессов не имеет значения. Однако существуют объекты, модель которых без явного учета реального времени не может быть адекватной. Таковы, например, распределенные системы, управляющие процессами синхронизации и взаимодействия информационных потоков в сложных программных комплексах реального времени с многопроцессорной структурой вычислительных средств.

Рассмотренные в первом разделе варианты расширения сетей Петри с попытками введения времени в формализм сети основаны опять же на структурных изменениях сети, к тому же громоздки и малоэффективны.

Поэтому естественным образом возникает концепция управляемого прохода. Управляемость перехода сразу влечет за собой синхронизацию функционирования сети, что, в общем случае, и требуется.

2.1. Функционирование управляемой сети Петри

Функционирование сети – это процесс изменений ее маркировок. Изменение разметки происходит в результате срабатывания перехода, которое приводит к изъятию меток (по кратности соответствующей дуги) из каждой входной для этого перехода позиции и к добавлению меток в каждую выходную его позицию. Таким образом, изменение разметки сети в результате срабатывания перехода $t_i \in T$ определяется векторами $In(t_i)$ и $Out(t_i)$ по очевидной формуле: $\mu' = \mu - In(t_i) + Out(t_i)$.

Теперь дополним математическое описание сети. А именно, введем вектор $V \in B^m$, где $B = \{1, 0\}$, то есть сопоставим каждому переходу либо 1, либо 0. Потребуем также, чтобы в каждый момент времени ровно одна компонента вектора V была единичной, а остальные нулевыми.

Используем также целочисленную матрицу Φ размерности $[n \times m]$, каждый столбец которой представляет собой n-вектор $F_i = -In(t_i) + Out(t_i)$. Тогда функционирование сети будет описано уравнением, которое отражает динамику поведения сети в целом:

$$\mu(k+1) = \mu(k) + \Phi^* V(k), \quad \mu(0) = \mu_0,$$

где $\mu(k)$ – разметка сети на k-м шаге ее функционирования; $V(k)$ – вектор, определяющий, какие именно переходы срабатывают на k-м шаге.

Рассмотрим более детально смысл, вкладываемый в понятие шага функционирования и вектора V.

Определение 2.1. Шагом функционирования сети будем называть изменение ее разметки в результате срабатывания ровно одного перехода.

Определение 2.2. Тактом функционирования сети будем называть изменение ее разметки в результате срабатывания всех возможных переходов.

Необходимо отметить, что на возможность срабатывания перехода $t_j \in T$ в данном случае дополнительно к условию $\forall p_i \in P\ \mu(p_i) \geqslant In(p_i, t_j)$ могут быть наложены и другие ограничения.

Такой подход (как при “шаге”, так и при “такте”) исключает конфликтные ситуации, когда некоторый переход, срабатывая, изымает метку из общей для двух переходов позиции и тем самым препятствует срабатыванию второго перехода. Точнее сказать, при “такте” описываются все возможные ситуации (в том числе и конфликтные), а при “шаге” реально конфликт возможен, но он должен быть разрешен до формирования вектора $V(k)$. На первый взгляд, исключаются также и параллельные процессы, когда могут срабатывать одновременно два перехода и более. Однако здесь важно понять, что в принятом формализме время пока нигде не участвует; интервал времени между состояниями $\mu(k)$ и $\mu(k+1)$ может быть любым, в том числе и нулевым, что и даст одновременность срабатывания переходов.

Уравнение $\mu(k+1) = \mu(k) + \Phi \times V(k)$, $\mu(0) = \mu_0$ описывает, собственно говоря, движение сети по графу разметок, причем одновременное срабаты-

$ich = 0$,
если $V_i(k) > 0$, то $\mu(isv + i) = \mu(k) + \Psi^* V_i(k)$ и $ich = ich + 1$,
где i меняется от 1 до m (проверяется активность переходов и набор разрешений на их запуск) при фиксированном k,
$isv = isv + ich$

Рис. 2. Базовые правила формирования дерева достижимости.

вание двух и более переходов возможно тогда, когда в графе существует простая цепь, дуги которой соответствуют последовательности срабатываний этих переходов.

Определение 2.3. Вектор $V(k) \in B^m$ будем называть вектором запуска переходов, где $B = \{1, 0\}$, и на каждом шаге функционирования ровно одна компонента $V(k)$ единична.

Ясно, что в сети могут возникнуть ситуации (например, тупиковые), когда ни один переход не может сработать. Поэтому необходимо ввести в рассмотрение нулевой вектор V, но при этом смысл шага функционирования не меняется, то есть для нулевого вектора V уравнение функционирования "вырождается": $\mu(k + 1) = \mu(k)$.

Далее, не все возможные значения вектора V при данной разметке являются допустимыми. Поэтому необходим еще механизм "фильтрации", запрещающий запуск некоторых переходов в зависимости от текущей разметки. Вектор V будет результатом работы этого механизма.

Определение 2.4. Вектор $U \in B^m$ будем называть вектором управлений, где $B = \{1, 0\}$, смысл которого заключается в указании того перехода, который необходимо запустить на k-м шаге функционирования.

Формализуем связь между векторами U и V. Условие срабатывания перехода имеет вид: $\mu(k) \geqslant In(t_j)$, где неравенство между векторами означает, что оно выполняется попарно для всех компонент. Иначе говоря, все входные позиции перехода должны иметь разметку не меньшую, чем кратность соответствующей дуги.

Аналитически связь между векторами U и V на k-м шаге функционирования для разметки $\mu(k)$ имеет вид:

$$V(k) = \Psi(\mu(k))^* U(k),$$

где Ψ – матрица размерности $[m \times m]$ определяется следующим образом:

$$\Psi_{ij} = 0, \quad \text{если } i \neq j;$$
$$\Psi_{ij} = \begin{cases} 1, & \text{если } \mu(k) \geq In(t_i); \\ 0 & \text{в противном случае.} \end{cases}$$

Вектор запуска $V(k)$ и уравнение изменения текущей разметки совместно полностью описывают функционирование сети в форме многошагового управляемого динамического процесса, а обобщенные подобным образом сети Петри, в которых введен бинарный вектор разрешений и запрещений на срабатывание всех переходов, имеющий смысл управления, будем называть управляемыми сетями или U-сетями.

Рассмотренная выше картина описывает функционирование сети лишь по одному из указанных путей (указание осуществляется бинарным вектором U) и не отражает полного дерева достижимости сети, то есть не описывает всех возможных направлений (дуг) от текущей разметки. С целью описания полного дерева сети для каждой разметки рассматривается такт функционирования, который состоит из перебора и запуска всех возможных (разрешенных) при данной разметке переходов. Заметим, что в этом случае смысл управления теряется, поскольку описываются все возможные срабатывания переходов. Рассмотрим процедуру построения полного дерева достижимости сети.

Формально такт функционирования сети для разметки $\mu(k)$ можно задать набором векторов управления $U_i \ldots U_m$, по которым формируется набор векторов разрешений и запрещений запуска переходов $V_1 \ldots V_m$.

Введем в описание понятие "номер вершины", который однозначно ставится в соответствие каждой вершине дерева сети. Так для начальной разметки сети μ_0 (корневой вершины) номер вершины равен 1 и записывается в виде $\mu(1) = \mu_0$. Первоначально дерево состоит из одной вершины $\mu(1)$. Заведем новую переменную isv, которая будет "счетчиком" вершин в полном дереве сети. Ясно, что начальное значение isv равно 1, поскольку дерево содержит одну вершину $\mu(1)$. Теперь такт функционирования можно записать в виде набора линейных правил для получения новых вершин дерева сети от текущей вершины с номером k (то есть для $\mu(k)$, причем первоначально $k = 1$) и соответствующим такой разметке набором векторов запуска $V_1(k) \ldots V_m(k)$:

Теперь, если менять последовательно номер вершины $k = k + 1$ при условии, что $k < isv$, то изложенные правила полностью задают рекурсив-

$ich = 0$,

если $V_i(k) > 0$, то
$\mu(isv + i) = \mu(k) - \ln{}^*V_i(k)$,
$\tau(isv + i) = \tau(k) + q_i{}^*V_i(k)$,
$EstTime(isv + i) = EstTime(k)$,
$ich = ich + 1$,

где $i = 1 \ldots m$ при фиксированном k,
$isv = isv + ich$,

где $k = 1 \ldots isv$

Рис. 3. Базовый набор правил поглощения фишек.

$\mu(isv + i) = \mu(k) + Out{}^*\, V_j(k)$,
$\tau(isv + 1) = \tau(k) - \tau_j(k)$,
$EstTime(isv + 1) = EstTime(k) + \tau_j(k)$,
$isv = isv + 1$

Рис. 4. Базовый набор правил возвращения фишек.

ное нахождение новых вершин по их номерам и тем самым позволяют создавать полное дерево сети, которое в большинстве случаев неограниченно. Выходом из описанной выше рекурсии являются ограничения на общее число вершин дерева ($isv <$ const), на число рассматриваемых вершин (k = 1, 2, ..., const) и другие условия, отражающие внутренние параметры сети – например, $V_i(1) = 0$ для любого i из $[1 \ldots m]$ или $k > isv$.

Поскольку данная процедура формирования дерева достижимости сети основана на использовании векторов запуска, то полученное таким образом дерево сети назовем полным деревом достижимости U-сети. Важно отметить, что рассмотренный аппарат совместно с алгоритмом для классических сетей Петри позволяет получать дерево достижимости в виде матрицы следования достижимых маркировок.

2.2. Временная область функционирования U-сетей. Определение τ-класса сетей

Введем явно время в математическое описание управляемой сети. Будем полагать, что функционирование U-сети происходит в дискретном временном пространстве, где длительность одного такта дискретного временного пространства равна условной временной единице.

Теперь выясним, как соотносится функционирование рассмотренной U-сети и сети, в которой дополнительно введены ненулевые длительности q_i срабатывания переходов t_i. Такая сеть задается как:

$$N_q = (P, T, R, W, \mu_0, Q),$$

где дополнительно введен временной вектор Q размерности $[m]$, состоящий из компонент q_i. В дальнейшем будем называть такую сеть временной сетью или τ-сетью. Ясно, что в τ-сети могут возникнуть разметки, не имевшие места в графе исходной асинхронной U-сети, благодаря поглощению меток. Такие разметки назовем промежуточными. На графе таким разметкам будут соответствовать промежуточные вершины или p-вершины. Наличие промежуточных вершин не привносит новых по сравнению с исходной U-сетью свойств, и анализ такой сети на наличие, например, тупиков можно проводить известными методами. Таким образом, τ-сеть не обладает новыми по сравнению с исходной U-сетью свойствами, и анализ такой сети на наличие, например, тупиков можно проводить известными методами.

Основное отличие модели τ-сети от асинхронной U-сети заключается в том, что наряду с уравнением функционирования сети, в модели τ-сети должны участвовать вектор $\tau(k)$ и число $EstTime(k)$. Каждая компонента $\tau_i(k)$ есть время, оставшееся работать переходу с номером i для рассматриваемой вершины графа с номером k (то есть $\mu(k)$), а число $EstTime(k)$ – текущее (расчетное) время для этой вершины. Теперь уравнение функционирования сети распадается на два автономных, но связанных временными параметрами, уравнения:

$$\mu(k+1) = \mu(k) - In{}^*V(k),$$

$$\mu(l+1) = \mu(l) - Out{}^*V(l).$$

Первое из них описывает изменение разметки в результате запуска перехода – переход “поглощает” фишки, второе – изменение разметки при завершении срабатывания перехода – переход “возвращает” фишки.

Действие сети при “поглощении” фишек может быть формализовано набором правил, определяющих начало работы переходов на каждом шаге функционирования. Второй набор правил соответствует тому, что работающий переход должен вернуть фишки в сеть. Базовый набор правил возвращения фишек определен в зоне действия следующих двух условий: $V_i(k) = 0$ для любого $I = 1 \ldots m$ (то есть нельзя запустить ни один переход) и существует переход t_j, который еще находится в рабочем состоянии (то есть в векторе $\tau(k)$ существует компонента $\tau_j(k) > 0$).

В базовом наборе правил возвращения фишек, во-первых, не учтено, что если “запущенных” переходов несколько (в векторе $\tau(k)$ существует несколько компонент $\tau_i(k) > 0$), то выбирается переход, у которого время, оставшееся до конца работы, – минимально ($\tau_j(k) = \min(\tau_i(k))$, где $\tau_i(k)$ – ненулевая компонента вектора $\tau(k)$ и $i = 1...m$).

Во-вторых, не учтена возможность одновременного окончания работы сразу нескольких переходов. Оценку завершения срабатывания перехода t_j можно проводить по рассчитанным параметрам *EstTime* и вектора τ таким образом:

если $\tau_j(isv + 1) > 0$ – переход продолжает работу и не меняет состояние, а при $\tau(isv + 1) = 0$ и $\tau_j(k) > 0$ – переход заканчивает работу и состояние разметки сети должно быть изменено. Если несколько переходов (t_{j1}, t_{j2}, ...) закончили свою работу одновременно, то уравнение изменения разметки сети при возвращении фишек преобразуется к виду:

$$\mu(isv + 1) = \mu(k) + Out^*(V_{j1}(k) + V_{j2}(k) + \ldots).$$

Рассмотренные поправки учитываются при конкретной реализации наборов правил управления сетью. Ясно, что правила поглощения фишек и их возвращения в сеть тесно взаимосвязаны между собой, поскольку запуск перехода при помощи вектора $V_i(k)$, формируемого управляющей программой, неизбежно влечет за собой завершение срабатывания перехода через время q_i. Вектор $V_j(k)$ из правил возвращения фишек, таким образом, не является управляющим и относится к внутренним характеристикам сетевой модели.

Концепция перехода с временным векторным управлением носит универсальный характер, поскольку способ формирования вектора управления U здесь не учтен. Варианты в определении вектора U позволяют реализовать управление по различным законам. В частности, если каждому переходу поставить в соответствие некоторый приоритет π_i, то все конфликтные ситуации будут разрешаться в пользу перехода с высшим приоритетом. Подобным же образом можно моделировать сеть, для каждого из переходов t_i которой задана вероятность p_i, и выявлять наиболее вероятное поведение сети при заданной начальной разметке. Возможны и другие варианты в зависимости от характера описываемых U-сетями объектов и решаемых для этих объектов задач.

СПИСОК ЛИТЕРАТУРЫ

1. *Крушный В.В.* Сети Петри при управлении параллельными взаимодействующими процессами. Deutschland, Saarbrucken: LAP LAMBERT Academic Publishing, 2012. 217 с.
2. *Котов В.Е.* О связи алгоритмических и архитектурных аспектов параллельных вычислений / В кн. Вычислительные процессы и системы. М.: Наука, 1986. С. 54–80.
3. *Питерсон Дж.* Теория сетей Петри и моделирование систем. М.: Мир, 1984. 264 с.
4. Применение микропроцессорных средств в системах передачи информации / Б.Я. Саветов, О.И. Кутузов, Ю.А. Головин, Ю.В. Аветов. М.: Высшая школа, 1987. 256 с.

Synchronization of Parallel Processes on the Basis of Petri's Networks with Operated Transitions

V. V. Krushny and P. V. Sagaydachnaya

Snezhinsk Physical Technical Institute, National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), ul. Komsomol'skaya 8, Snezhinsk, Chelyabinskaya oblast, 456776 Russia

e-mail: poli_nochka@mail.ru

Abstract—Formal models of parallel processes have been considered on the basis of the mathematical theory of Petri's nets. Another simpler method has been proposed for introducing the time into the net formalism at the level of the external control. This method excludes the violation of the originally simple developed and laconic structure of the Petri's net.

Keywords: Petri's net, operated transitions, parallel processing, computing models, data exchange