МАТЕМАТИЧЕСКОЕ И АРХИТЕКТУРНОЕ ОБЕСПЕЧЕНИЕ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ
(Вычислительные системы)
1985 год Выпуск 109

УДК 681.324

СЕТИ ПЕТРИ И КОРРЕКТНОСТЬ ПРОТОКОЛОВ ПЕРЕДАЧИ ДАННЫХ

О.Л. Бандман

§1. Задача проверки корректности протокола

Протоколы передачи данных определяют дисциплину взаимодействий между компонентами распределенных систем в процессе организации и выполнения обмена информацией между ними.

Поскольку эти взаимодействия имеют асинхронный характер, а система отношений между компонентами сложна и запутана, протокол трудно составить так, чтобы при реализации не возникало непредусмотренных ситуаций, т.е. чтобы описание протокола было корректно. Отсюда возникает задача создания удобного инструмента автоматизированной проверки корректности разрабатываемого протокола. Интенсивно ведутся поиски путей создания такого инструмента [1-3]. Среди многих других значительное место занимает направление поиска, связанное с использованием сетей Петри [4-6]. В пределах этого направления можно назвать несколько подходов, различающихся интерпретациями вершин сетей Петри: атрибутные сети [8], предикатные сети [7,9], временные сети [10].

В основе этих интерпретаций лежит обычная (неинтерпретированная) сеть Петри, моделирующая протекающие процессы в терминах абстрактных условий и событий. Аппарат анализа свойств таких сетей Петри достаточно хорошо разработан. Можно сказать, что уже накоплен некоторый опыт его использования [4,11]. Поэтому представляется целесообразным проанализировать имеющийся материал с тем, чтобы выяснить, насколько полезна модель обычной сети Петри для проверки корректности протоколов, какие поведенческие свойства можно и какие нельзя выявить на основе этой модели. Ответ на этот вопрос поможет разобраться также и в том, какие интерпретации необходимы для проверки корректности.

Понятие "корректность протокола" определить строго невозможно, поскольку его описание задается в виде спецификаций неформально. Спецификация задает набор объектов, всех возможных (желательных и ошибочных) состояний, в которых они могут находиться, условия переходов из состояния в состояние, исходное состояние и терминальное, которое соответствует поставленной цели. Считается, что протокол составлен корректно, если при его реализации поведение системы отвечает следующим требованиям.

I. Отсутствие тупиков, т.е. таких состояний, когда цель еще не достигнута, а ни один объект не может изменить своего состояния.

2. Полнота - все предусмотренные спецификацией условия, действия и ситуации учтены и соответствующие состояния достижимы.

3. Отсутствие бесполезных зацикливаний.

4. Недопустимость переполнений - реализация ограничений на количество переходов в отдельные состояния, обусловленные спецификацией.

5. Соответствие терминальных состояний поставленным в спецификациях целям.

Эти пять условий определяют о б щ у ю корректность, поскольку все протоколы обязательно должны удовлетворять этим условиям. В некоторых случаях бывает важно обеспечить выполнение дополнительно какого-нибудь частного условия, характеризующего правильное поведение группы объектов в системе. Например, "линия не должна передавать более чем n сообщений одновременно" или "в группе линий запросов должен содержаться один и только один запрос". Такие специфические для данного протокола условия определяют его ч а с т н у ю корректность. Они формулируются в виде инвариантов и должны быть проверяемы отдельно от условий общей корректности.

Переход к автоматизированной проверке корректности протоколов требует формального выражения условий корректности и разработки алгоритмов их распознавания в рамках выбранной модели. Говоря более конкретно, создание инструмента проверки корректности протоколов на основе сетей Петри требует решения следующих задач.

I. Разработка приемов построения сети Петри, отображающей все условия и действия, указанные в спецификации протокола.

2. Формулировка свойств сети Петри, соответствующих условиям общей корректности, а также выделение инвариантов, характеризующих частные условия.

3. Выбор методов анализа сетей Петри на заданные свойства и наличие инвариантов.

4. Реализация алгоритмов распознавания заданных свойств сетей Петри.

5. Интерпретация результатов машинного анализа.

Первая и последняя из перечисленных задач связывают проблему корректности протоколов с теорией и практикой применения сетей Петри. Эти задачи не решаются формально, по крайней мере до тех пор, пока не стандартизован формализм представления спецификаций. Решения остальных задач основаны на хорошо разработанных методах анализа свойств сетей Петри, частично автоматизированы и опробованы на реальных примерах, в том числе протоколах передачи данных [4].

§2. Сети Петри. Основные понятия

С е т ь П е т р и - двудольный граф с кратными дугами без петель, характеризуемый тремя параметрами $N = \langle \Pi, \Sigma, C \rangle$ (рис.I,а). Здесь $\Pi = \{p_1, \ldots, p_m\}$ - множество вершин, называемых позициями и обозначаемых кружками, $\Sigma = \{t_1, \ldots, t_n\}$ - множество вершин, называемых переходами и обозначаемых черточками. Подмножество позиций, из которых есть дуги в переход t_j, обозначается ${}^{\bullet}t_j$, подмножество позиций, в которых есть дуги из t_j, обозначается $t_j^{\bullet}$. Аналогично, ${}^{\bullet}p_i$ и $p_i^{\bullet}$ - подмножества входных и выходных переходов для p_i; C - матрица инцидентности сети Петри, размерами $m \times n$. Строки ее соответствуют позициям, столбцы - переходам. Элементы матрицы

$$c_{ij} = \begin{cases} v(p_i, t_j), & \text{если } p_i \in {}^{\bullet}t_j, \\ -v(t_j, p_i), & \text{если } p_i \in t_j^{\bullet}, \\ 0, & \text{если } p_i \text{ и } t_j \text{ не связаны дугой,} \end{cases}$$

где $v(p_i, t_j)$ - кратность дуги (p_i, t_j) (рис.I,б).

На множестве позиций Π задается функция, отображающая Π в множество целых неотрицательных чисел. Если $M(p_i) = k$, то говорят, что позиция p_i имеет k маркеров. Переход t_j возбужден, если в каждой его входной позиции $p_i \in {}^{\bullet}t_j$ число маркеров не меньше чем кратность дуги (p_i, t_j), т.е. если $M(p_i) \geq v(p_i, t_j)$ для всех $p_i \in {}^{\bullet}t_j$. Переход может быть возбужденным в течение произвольного конечного времени. Затем он либо срабатывает, либо возбуждение с

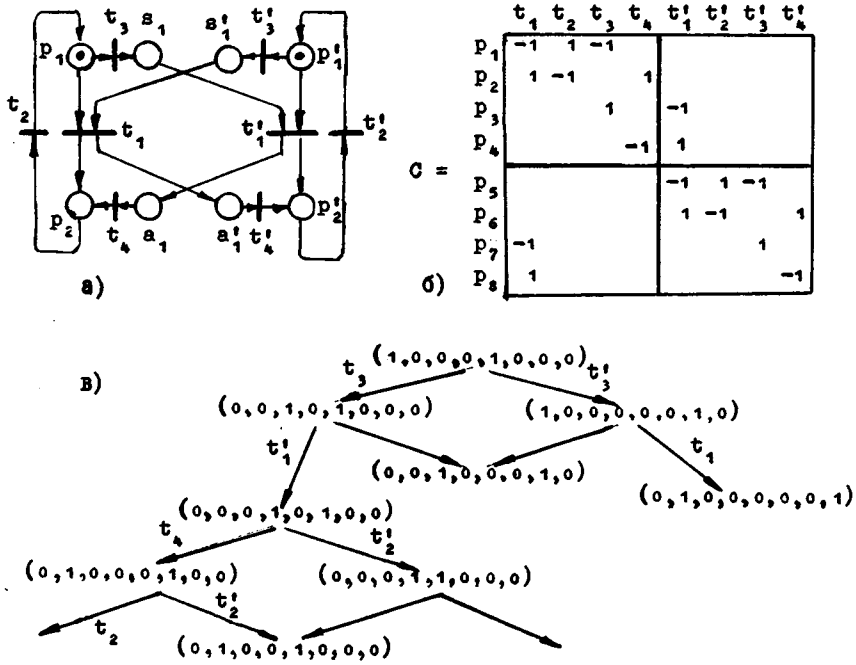


Рис. I

него снимается (при срабатывании другого перехода). Срабатывая, переход отбирает из каждой позиции $p_i \in {}^*t_j$ по $v(p_i, t_j)$ маркеров и добавляет в каждую $p_k \in {}^*t_j$ по $v(t_j, p_k)$ маркеров. Маркировкой сети Петри называется вектор $M = (M(p_1), M(p_2), \ldots, M(p_n))$.

Если в сети задана начальная маркировка M_0, при которой хотя бы один переход оказывается возбужденным, то в сети начинается движение маркеров. Говорят, что маркировка M_0 порождает последовательность срабатываний переходов $\sigma = t_{i_1}, \ldots, t_{i_k}$, что приводит к новой маркировке

$$M = M_0 + C\bar{\sigma}, \qquad (I)$$

где $\bar{\sigma} = (\delta(t_1), \delta(t_2), \ldots, \delta(t_n))$ - характеристический вектор последовательности σ, $\delta(t_i)$ - число вхождений t_i в σ.

Маркировки, получающиеся в результате срабатываний последовательностей переходов, порождаемых M_0, образуют множество достижимых из M_0 маркировок $\hat{M}(M_0)$. Граф, вершины которого обозначены маркировками $M \in \hat{M}(M_0)$, а дуги - переходами $t \in \Sigma$ таким образом, что если M' - результат срабатывания t_i при M, то вершины M и M' соединены дугой, называется графом достижимости (рис.I,в). Маркировки, которые не порождают ни одной последовательности срабатывания, называются тупиковыми.

Важными подклассами сетей Петри являют автоматные сети Петри и сети Петри со свободным выбором. В автоматных сетях каждый переход имеет одну входящую и одну выходящую дугу. По поведению автоматные сети Петри эквивалентны конечным автоматам. В сетях со свободным выбором количества входных и выходных дуг в вершинах не ограничены, однако удовлетворяется следующее условие: если два перехода имеют общую входную позицию, то других входных позиций у этих переходов нет. Сети со свободным выбором моделируют поведение параллельных граф-схем алгоритмов. Их свойства наиболее полно изложены в [I2].

§3. Построение сети Петри для заданного протокола

Процесс построения сети Петри для заданного протокола состоит из трех этапов: I) выделения взаимодействующих компонент, 2) построения сети Петри для каждой компоненты и 3) объединения компонент в единую сеть Петри.

Первый этап предусматривает неформальный анализ спецификации, в результате которого определяются компоненты взаимодействия (протокольные машины, передающая среда). Каждое действие и каждое условие, упомянутое в спецификации, приписывается своей компоненте. Для каждой компоненты составляется алгоритм функционирования. При этом следует стремиться представить этот алгоритм в максимально параллельном виде. Такое представление достигается, когда на каждом шаге допускается одновременное выполнение всех действий, для которых на одном шаге оказались удовлетворенными условия их реализации. Чем полнее использована возможность распараллеливания алгоритма функционирования компоненты, тем проще (с меньшим числом вершин) получится моделирующая его сеть Петри.

Второй этап - построение сети Петри для компонент - самый ответственный. Общие правила перехода от алгоритмического описания к сети Петри заключается в следующем.

I. Каждое действие (или неделимая последовательность дейст - вий) или переход из состояния в состояние интерпретируется как событие. Событием считается, например, выработка или прием сигнала, возврат в исходное состояние и т.д. Событию ставится в соответствие фрагмент сети Петри, изображенный на рис. 2. Входные к переходу t позиции и кратности дуг, соединяющих их с t, соответствуют условиям, которые должны быть выполнены, чтобы событие могло совершиться. Выходные позиции и кратности дуг, соединяющих с ними переход t, соответствуют условиям, возникающим в результате события. Условием считается, например, тот факт, что предшествующее событие произошло, наличие (или отсутствие) сигнала, вырабатываемого другой протокольной машиной и т.д.

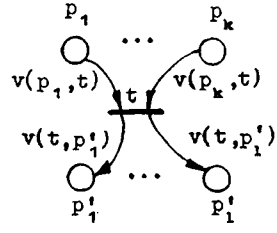


Рис.2

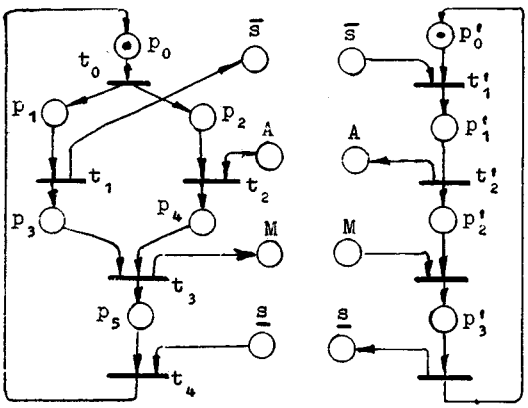
2. Позиции, соответствующие одним и тем же условиям разных фрагментов, совмещаются.

3. Выделяются подмножества позиций $П_0 \subset П$ и $П_к \subset П$, отображающих условия исходного и терминального состояний компоненты соответственно. Определяются векторы начальной $М_0$ и терминальной $М_к$ маркировок.

4. Полученная сеть Петри замыкается, т.е. добавляется переход τ, такой что $^{\bullet}\tau = П_к$, $\tau^{\bullet} = П_0$, $v(p \in П_к, \tau) = М_к(p)$, $v(\tau, p' \in П_0) = М_0(p')$, что соответствует переходу в исходное состояние.

ПРИМЕР I. Упрощенный протокол передачи данных по каналу для управления технологическим процессом в реальном времени [4]. Передача сообщения возможна, когда канал, свободен со стороны пере - дачи. Тогда посылается сигнал $\bar{s}$ о занятости канала и принимается сигнал A о готовности к приему сообщения. Если сигнал $\bar{s}$ послан и сигнал A получен, то сообщение M передается. После принятия сообщения M принимающая сторона вырабатывает сигнал $\underline{s}$ (освобождение канала) и может переходить в исходное состояние. Если сообщение послано и имеется сигнал $\underline{s}$ передающая сторона может переходить в исходное состояние. Естественным образом выделяются две компонен-

ты: передатчик и приемник (рис.I,а и I,б соответственно). Передатчик интерпретирует следующие события: выработка сигнала $\bar{s}$ (переход t_1), принятие сигнала A (переход t_2), посылка сообщения M (переход t_3), принятие сигнала $\underline{s}$ (переход t_4). Объединение фрагментов, отображающих эти события, дает сеть Петри свободного выбора (рис.3,а). Начальная маркировка $M_0(p_0) = 1$, $M_0(p_i, \bar{s}, A, M, \underline{s}) = 0, (i = 1,2,3,4,5)$. Следует отметить, что при объединении фрагментов введен вспомогательный внутренний переход t_0, позволяющий использовать условие p_0 двум событиям t_1 и t_2 одновременно или в произвольном порядке.Приемник интерпретирует следующие события: прием запроса $\bar{s}$ (переход t'_1), передача сигнала A (переход t'_2), прием сообщения M (переход t'_3) и передача сигнала $\underline{s}$ (переход t'_4). Начальная маркировка сети Петри приемника $M(p'_0)=1$, $M(p_i, \bar{s}, A, M, \underline{s}) = 0$ $(i=1,2,3)$ (рис.3,б). Полученная в результате объединения соответствующих фрагментов сеть Петри относится к классу автоматных.



а) б)

Рис. 3

Пример I иллюстрирует те случаи, когда сети Петри для протоколов могут быть построены при помощи формальных операций и, следовательно, процесс их построения может быть автоматизирован. Первый наиболее распространенный случай - это конечноавтоматное представление взаимодействующей компоненты. В этом случае граф состояний автомата преобразуется в автоматную сеть Петри простой заменой фрагментов (рис.4), причем позиция, соответствующая начальному состоянию автомата маркируется одной меткой. Второй

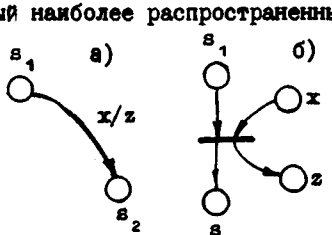


Рис. 4

случай - когда функционирование компоненты задано параллельной граф-схемой алгоритма. Переход от граф-схемы к сети Петри осуществляется формальной заменой ее вершин на фрагменты сети Петри (рис.5). При этом получается сеть Петри со свободным выбором.

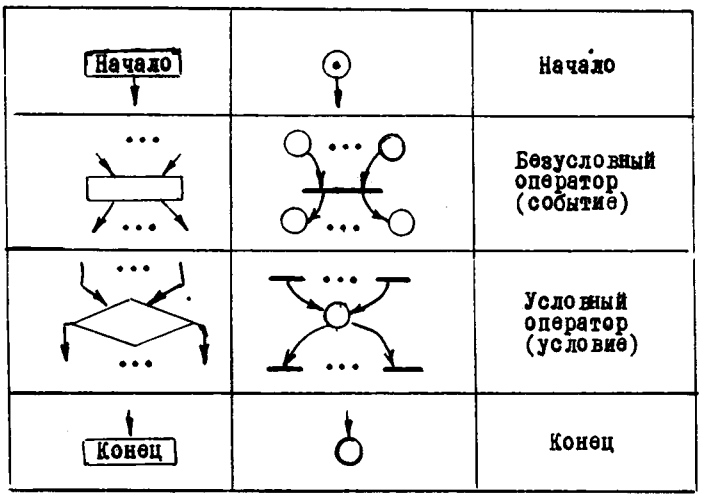


Рис. 5

ПРИМЕР 2. Функционирование передатчика в примере I может быть задано параллельной граф-схемой алгоритма, а функционирование приемника - конечным автоматом,изображенными на рис.6,а,б соответственно. Нетрудно видеть, что формальные замены, выполненные согласно рис.4 и 5,приведут к сетям Петри, полученным в примере I.

Сеть Петри, отображающая взаимодействия компонент, получается путем объединения сетей Петри всех компонент.Процедура объединения заключается в совмещении одноименных позиций (а иногда и переходов) компонентных сетей. Например, объединение сетей Петри компонент из примера I дает сеть, изображенную на рис.7.

В общем случае построение сети Петри по спецификации протокола требует большого искусства, поскольку выделить события и условия для всех предусмотренных ситуаций очень сложно. Далее приводятся примеры выработанных опытом приемов отображения типичных ситуаций.

ПРИМЕР 3. Фрагмент сети Петри,отображающий тайм-аут (в предположении,что он работает без ошибок). Во многих протоколах пре-

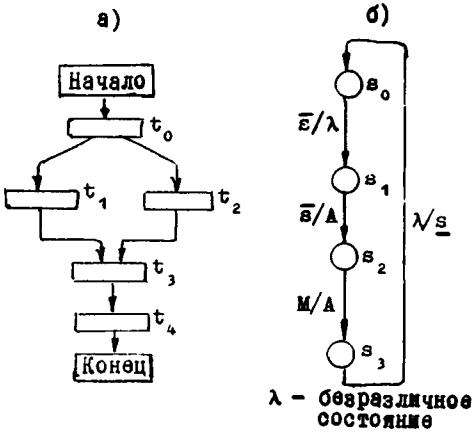


Рис. 6

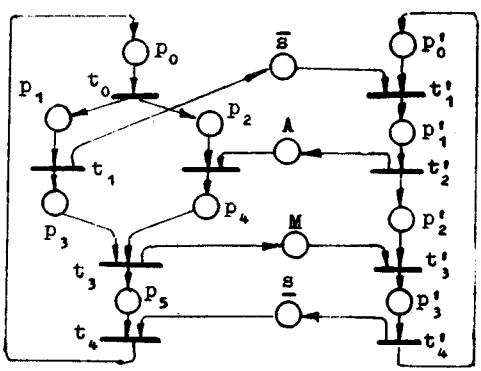


Рис. 7

дусматривается возможность потери сообщения при передаче или потере подтверждения о его получении и предписывается посылка нового сообщения, если в течение заданного времени T (тайм-аут) подтверждения нет. Считается, что взаимодействуют три компоненты: передатчик, передающая среда (линия) и приемник. В сети Петри передатчика выделим переходы t_1 (сообщение послано) и t_2 (подтверждение принято) (рис. 8). В сети Петри приемника выделим переходы t_1' (сообщение принято) и t_2' (подтверждение получено). В линии выделим условия p_1 (сообщение в линии) и p_2 (подтверждение в линии). Для отображения работы таймера в сеть Петри линии вводится позиция p_3 (информация потеряна) и два перехода t_1'' и t_2'', отображающих события потери сообщения и подтверждения соответственно. В сеть Петри передатчика вводится позиция p_4 (таймер взведен) и t_3 (таймер сработал). Таймер взводится ($M(p_4)=1$) всякий раз, как посылается сообщение (переход t_1), и срабатывает, если информация потеряна ($M(p_3)=1, M(p_4) = 1$), возвращая передатчик в исходное состояние для повторной передачи. Если сообщение принято, то должен произойти сброс таймера

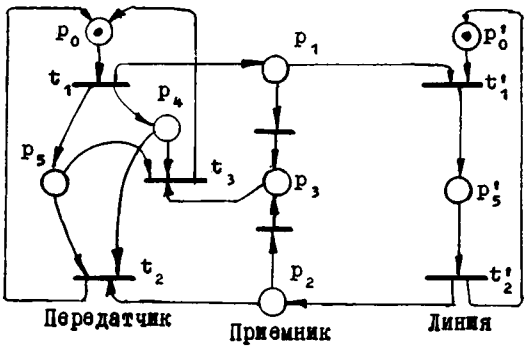


Рис. 8

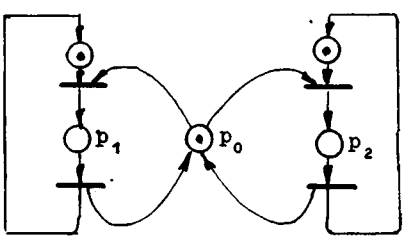


Рис. 9

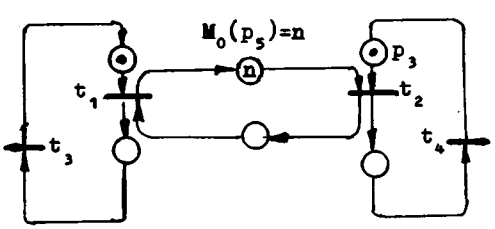


Рис. I0

(дуга (p_4,t_2)).Отсутствие указания на сброс в протоколе X.2I и соответственно этому отсутствие дуги сброса было одной из ошибок, обнаруженных при проверке корректности описания этого протокола [I3].

ПРИМЕР 4. Фрагмент сети Петри, отображающий взаимное исключение. Если какие-либо два условия в разных компонентах (или в параллельных ветвях одной компоненты) не должны выполняться одновременно, то в сеть Петри добавляется фрагмент взаимного исключения (арбитр). Например, две протокольные машины имеют общую память. Два события: машина M_1 производит чтение (условие p_1), и машина M_2 производит

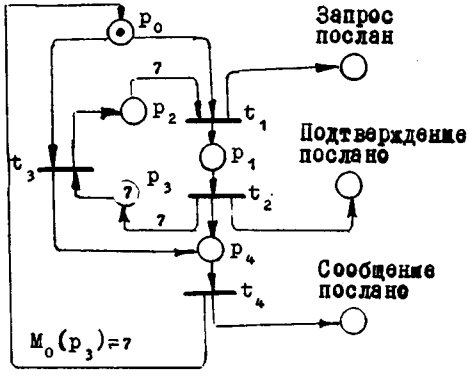


Рис. II

запись (условие p_2) должны быть развязаны во времени. В сеть Петри добавляется маркированная позиция p_0 ($M(p_0) = 1$), отображающая нужную развязку (рис.9).

ПРИМЕР 5. Фрагмент сети Петри, отображающий канал с буфером. Если в протоколе указывается, что в канале может находиться одновременно не более n сообщений, то такой канал отображается сетью, изображенной на рис.10, причем $M_0(p_1) = n$. Маркировка позиции p_1 ($M(p_1)$) отображает число свободных, а маркировка p_2 ($M(p_2)$) – число занятых ячеек. Если $M(p_1)=0$ (буфер занят), то t_1 не срабатывает (передача в канале невозможна). Число $M(p_1)+M(p_2)$ всегда равно n и является инвариантом сети Петри (рис.10).

ПРИМЕР 6. Фрагмент сети Петри, отображающий передачу с ограниченным количеством подтверждений. Протокол, описанный в [II,I4], разрешает передачу сообщений без подтверждения, но не более семи сообщений подряд. Фрагмент сети Петри, отображающий такие условия передачи, изображен на рис.II. Позиция p_1 (сообщение послано) может получить маркер либо при срабатывании t_1t_2 (с подтверждением), либо при срабатывании t_3 (без подтверждения). Начальная маркировка $M_0(p_3) = 7$ определяет существование последовательности $\sigma = (t_3t_4)^7$, причем $M(p_2)$ равно количеству посланных без подтверждения сообщений. Если $M(p_2) = 7$, то возможна передача только с подтверждением.

§4. Свойства сетей Петри, отображающие условия корректности протокола

Для того, чтобы проверять корректность протоколов путем анализа отображающих их сетей Петри, необходимо провести сопоставление условий корректности и свойств сетей Петри. Если бы мы распо-

лагали формальными определениями условий корректности, то должны были бы доказать каждый факт соответствия свойства сети какому-либо условию корректности. Но поскольку мы ограничиваемся содержательным пониманием условий корректности, то на таком же уровне придется показывать, как эти условия отображаются в свойства сетей Петри. Однако последние будём определять строго, поскольку распознавание их ориентировано на ЭВМ.

Свойства сетей Петри, характеризующие поведение отображаемых ими процессов, подразделяются на две группы: I) свойства, определяющие активность сети: живость, беступиковость, повторяемость, и 2) свойства функций маркировок: ограниченность, безопасность, консервативность. Рассмотрим отдельно свойства первой и второй групп.

ОПРЕДЕЛЕНИЕ I. Переход t маркированной сети Петри $\langle N, M_0 \rangle$ называется ж и в ы м, если из любой маркировки M, достижимой из M_0, существует последовательность срабатывания, включающая t. Маркированная сеть Петри называется ж и в о й, если все ее переходы живые.

Живая сеть Петри отображает поведение такой системы, в которой каждое событие остается потенциально выполнимым при любом достижимом состоянии системы (если после достижения цели она возвращается в исходное состояние).

Например, легко проверить, что сети Петри, изображенные на рис. 3,а, 3,б, 7-II - живые, а сеть Петри, изображенная на рис. I, - неживая, так как маркировка M = (0,0,I,0,0,0,I,0), достижимая из M_0 при $\sigma = t_3\, t_3'$, не порождает ни одной последовательности срабатывания.

ОПРЕДЕЛЕНИЕ 2. Последовательность срабатываний σ называется п о в т о р я ю щ е й с я [11], если любая маркировка M, порождающая σ, порождает также $\sigma^* = \sigma\sigma\sigma\ldots$.

ОПРЕДЕЛЕНИЕ 3. Подмножество переходов $T \in \Sigma$ называется п о в т о р я ю щ е й с я с о с т а в л я ю щ е й сети Петри N = $= \langle \Sigma, П, C \rangle$, если существует повторяющаяся последовательность σ такая, что множество входящих в σ переходов равно T и $C\bar{\sigma} \geq 0$. Если $C\bar{\sigma} = 0$, то T - стационарно повторяющаяся составляющая, а вектор $\bar{\sigma}$ называется T-и н в а р и а н т о м [I5] или t-полупотоком [II].

Сеть Петри может иметь несколько T -инвариантов. Множество T-инвариантов определяет множество возможных циклов в системе.

ОПРЕДЕЛЕНИЕ 4. Сеть Петри $N=(П,\Sigma,C)$ называется п о в т о р я ю щ е й с я (стационарно-повторяющейся), если Σ является повторяющейся (стационарно-повторяющейся) составляющей.

Из определения 3 следует, что сеть имеет повторяющуюся составляющую T, если существует такой вектор $x = (x_1,\ldots,x_n)$ $(x_i \geq 0)$, что

$$Cx \geq 0, \qquad (2)$$

и имеет T-инвариант, если

$$Cx = 0, \qquad (3)$$

причем каждому x, удовлетворяющему (2) или (3), соответствует вектор $\bar{\sigma}$ повторяющейся последовательности σ.

Важным является тот факт, что свойство повторяемости (в отличие от живости) не зависит от начальной маркировки. Это следует понимать так, что для повторяющейся сети всегда найдется такое M_0, которое порождает повторяющуюся последовательность срабатываний. Следует отметить также, что живая сеть обязательно повторяющаяся. Однако обратное неверно: повторяющаяся сеть не обязательно живая.

Так, например, сеть из примера I (рис.I) является повторяющейся. Решения неравенства (2) для этой сети $x' = (0,I,I,I,I,0,0,0)$ и $x'' = (I,0,0,0,0,I,I,I)$ и все их линейные комбинации составляют множество T-инвариантов, определяющих повторяющиеся последовательности $\sigma_1=(t_3,t_1',t_4,t_2)^*$, $\sigma_2 = (t_3', t_1', t_4', t_2')^*$ и все их конкатенации, а также повторяющиеся составляющие $T_1=\{t_2,t_3,t_4,t_1'\}$, $T_2=\{t_1,t_2',t_3',t_4'\}$ и $T_3=T_1 \cup T_2 = \Sigma$. Поскольку $T_3 = \Sigma$ (согласно определению 4) сеть - повторяющаяся. Однако, как уже было показано, она неживая.

ОПРЕДЕЛЕНИЕ 5. Сеть Петри называется б е с т у п и к о в о й, если существует такая начальная маркировка, что любая, достижимая из нее порождает хотя бы одну последовательность срабатываний.

Существенно отметить то обстоятельство, что повторяющаяся сеть Петри всегда бесступиковая. Однако обратное неверно, хотя бесступиковая сеть обязательно имеет хотя бы один T-инвариант. Особенно важно помнить, что бесступиковая сеть может быть неживой.

Сеть Петри, изображенная на рис.I2, бесступиковая при $M_0 = (I,0,0)$. Она не является ни живой, ни повторяющейся, но имеет два инварианта $T_1 = \{t_1\}$ и $T_2 = \{t_2\}$.

Повторяющиеся последовательности срабатываний и всевозможные их конкатенации образуют бесконечные последовательности срабатываний.

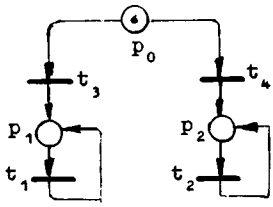


Рис. I2

ОПРЕДЕЛЕНИЕ 6. Бесконечная последовательность срабатываний называется **незаконной**, если существует переход $t \in \Sigma$, который находится в возбужденном состоянии при всех маркировках, сопровождающих эту последовательность. Название "незаконная" произошло от того, что реальное существование такой последовательности бесконечным быть не может, поскольку противоречит действующему в сетях Петри "закону" о том, что переход не может находиться в возбужденном состоянии бесконечно долго. Так, например, в сети Петри, изображенной на рис. I3, бесконечная последовательность срабатываний $\sigma = (t_5, t_6)^\omega$ незаконная, так как переход t_1 находится в возбужденном состоянии. Незаконные последовательности обычно не принимаются во внимание при анализе сетей Петри, так как не отражают реальной возможности существования бесконечной последовательности [I6].

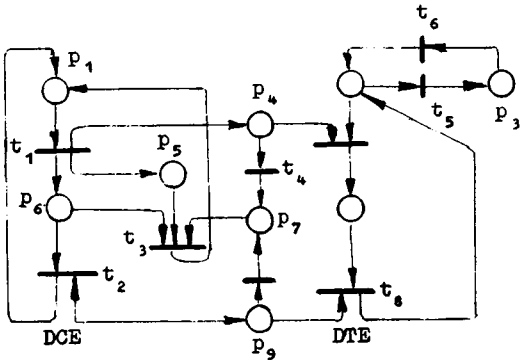


Рис. I3

ОПРЕДЕЛЕНИЕ 7. Бесконечная законная последовательность срабатываний называется **несправедливой** по отношению к переходу t, если она его не содержит.

Несправедливая последовательность срабатываний отображает несогласованность событий, в результате которой какое-либо событие, периодически приобретая готовность, теряет ее, не свершившись.

ПРИМЕР 7. Фрагмент протокола X.2I, рекомендованного МККТТ для реализации связи между терминальным оборудованием (DTE) и ап-

паратурой передачи данных (DCE) . Фрагмент определяет процедуру посылки запроса от DCE к DTE и отображается сетью Петри, изображенной на рис.I3. Событие посылки запроса отображается переходом t_1. При этом взводится таймер (позиции p_5 и p_6), и сигнал попадает в линию (позиция p_4). Позиция p_2 отображает условие готовности DTE к приему запроса. Кроме того, предусмотрена возможность прерывания DTE (переход t_5), когда готовность к приему запроса временно нарушается (условие p_3). Очевидно, что запрос может быть послан, когда DTE не готов к его приему ($M(p_2) = 0$, $M(p_3) = 1$). Поскольку запрос не принят, таймер срабатывает (последовательность t_3, t_4) так, как будто запрос потерян. Пока восстанавливается состояние DCE для повторного запроса (условие p_1), DCE может прийти в состояние готовности и вновь его потерять. Эта последовательность событий отображается законной бесконечной последовательностью срабатываний $\sigma = (t_1, t_5, t_4, t_3, t_6)^*$, при которой переход t_7 (прием запроса) хотя и возбуждается периодически, но не срабатывает, что свидетельствует о несогласованности циклов работы таймера и прерывания. Этот недостаток был выявлен при анализе корректности протокола X.21 [I3], и было рекомендовано внести в спецификацию изменения.

ОПРЕДЕЛЕНИЕ 8. Позиция $p \in П$ маркированной сети Петри $\langle N, M_0 \rangle$ ($N = \langle П, \Sigma, C \rangle$) называется k-о г р а н и ч е н н о й (k – целое положительное), если для любой достижимой из M_0 маркировки $M(p) \leq k$. Сеть Петри $\langle N, M_0 \rangle$, в которой все позиции k-ограничены, называется k-ограниченной. Если сеть I-ограничена, то она называется б е з о п а с н о й.

ОПРЕДЕЛЕНИЕ 9. Сеть Петри N называется о г р а н и ч е н н о й, если для любой начальной маркировки M_0 найдется такое k, при котором сеть k-ограничена.

Важно отметить, тот факт, что если сеть Петри со свободным выбором имеет начальную маркировку M_0, такую что для любого $p \in П$, $M_0(p) \leq 1$ и является ограниченной, то она обязательно безопасна [I7].

ПРИМЕР 8. Простейшая сеть Петри, отображающая передачу сигналов от компоненты A к компоненте B без подтверждений (рис.I4), является неограниченной. Последовательность срабатываний $\sigma = t_1, t_2, t_1, t_2 \ldots$ сопровождается последовательностью маркировок (I,0,0,I,0), (0,I,I,I,0), (I,0,I,I,0), (0,I,2,I,0), (I,0,2,I,0)..., в которой $M(p_3)$ неограниченно возрастает.

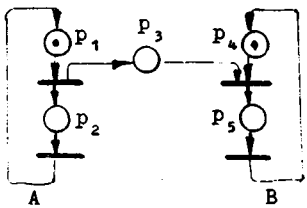


Рис.I4

Свойство ограниченности достаточно хорошо изучено. В частности, известны следующие важные для нас факты.

I. Если в маркированной сети Петри $\langle N, M_0 \rangle$ какая-либо достижимая из M_0 маркировка M порождает последовательность срабатываний, приводящую к $M' \geq M$, то сеть $\langle N, M_0 \rangle$ неограниченная при M_0.

2. Сеть ограниченная (при любом M_0), если существует вектор $y = (y_1, \ldots, y_m)$, где y_i – целое положительное число такое, что

$$yC \leq 0. \qquad (4)$$

Вектор y имеет длину, равную числу позиций в сети, y_i – весовой коэффициент позиции p_i. Число z_j – элемент результирующего вектора $z = yC$ – характеризует взвешенное приращение количества маркеров в сети при срабатывании перехода t_j.

ОПРЕДЕЛЕНИЕ I0. Подмножество позиций $P \subseteq \Pi$, которое сохраняет взвешенное число маркеров $K = \Sigma\, y_i M(p_i)$, $y_i \geqslant 0$, при всех достижимых из M_0 маркировках при любом M_0, называется консервативной составляющей сети Петри. Консервативная составляющая существует, если существует решение уравнения

$$yC = 0, \qquad (5)$$

где $y = (y_1, \ldots y_m)$, причем y_i – целое неотрицательное число. Вектор y, удовлетворяющий уравнению (5), называется P-инвариантом сети [I5] или p-полупотоком [II].

Например, P-инвариантами сети Петри (рис.I0) являются векторы $y' = (I,I,0,0,0,0)$, $y'' = (0,0,I,I,0,0)$, $y''' = (0,0,0,0,I,I)$, причем $K' = 2$, $K'' = 2$, $K''' = n$, P – инвариант сети Петри (рис.II) равен $y = (0,7,I,I,0)$, причем $K = 7$.

ОПРЕДЕЛЕНИЕ II. Если множество позиций сети Петри Π является ее P-инвариантом, то сеть называется консервативной.

Консервативная сеть обязательно ограниченная (обратное неверно). Это обстоятельство позволяет использовать знания о консервативности компонент для суждения о консервативности и, возможно, ограниченности всей сети.

§5. Отображение условий корректности свойствами сетей Петри

Покажем теперь, что если сеть Петри жива и ограничена, то условия общей корректности соответствующего протокола удовлетворены. Рассмотрим отдельно каждое из приведенных на с. 30 условий.

1. О т с у т с т в и е т у п и к о в ы х с и т у а ц и й непосредственно следует из определения 2, которое утверждает невозможность тупиковых маркировок в живой сети Петри.

2. П о л н о т а обеспечивается двумя условиями. Во-первых, необходимо, чтобы ни одна ситуация, указанная в спецификации протокола, не осталась не отображенной в сети Петри. Во-вторых, все состояния системы, соответствующие этим ситуациям, должны быть достижимы из исходного состояния. Первое условие относится к способу построения сети Петри по спецификации протокола, во многом зависит от однозначности толкования содержащихся в ней указаний и лежит вне рамок формальной автоматизированной проверки. Второе условие обеспечивается тем, что в живой сети для каждого перехода найдется достижимая маркировка, при которой он сработает.

3. О т с у т с т в и е б е с п о л е з н ы х з а ц и к л и в а н и й - условие корректности, выдвинутое в [2,3] без четкого определения этого понятия. Однако вполне правомерно дать ему следующую трактовку. Бесполезное зацикливание - это не предусмотренная спецификацией бесконечная, повторяющаяся последовательность событий (цикл), из которой система не может выйти без постороннего воздействия. В терминах сетей Петри это бесконечная повторяющаяся последовательность срабатываний σ, в которой каждый срабатывающий переход является единственным возбужденным переходом и которая не содержит всех переходов. Понятно, что живая сеть Петри не может обладать соответствующим Т-инвариантом, так как его наличие означало бы неживость не вошедших в σ переходов.

4. Н е д о п у с т и м о с т ь п е р е п о л н е н и й полностью гарантируется свойством ограниченности соответствующей сети Петри.

5. С о о т в е т с т в и е т е р м и н а л ь н ы х с о с т о я н и й поставленным целям трансформируется в свойство достижимости терминальных маркировок, отображающих эти цели. Эти терминальные маркировки соответствуют терминальным состояниям взаимодействующих компонент и порождают последовательности сраба-

тываний (состоящие из замыкающих переходов), приводящие к начальной маркировке M_0. В живых ограниченных сетях каждая маркировка, из которой достижима M_0, является достижимой. Следовательно, достижимы и терминальные.

Таким образом, чтобы убедиться в общей корректности протокола, необходимо соответствующую сеть проанализировать на живость и ограниченность.

Для важного частного случая, когда сеть Петри, отображающая протокол, относится к сетям со свободным выбором с начальной маркировкой, в которой $M(p_i) = 1$ для всех $p_i \in \Pi$, общая корректность обеспечивается живостью и безопасностью сети Петри.

ПРИМЕР 9. Упрощенный фрагмент протокола X.21, который определяет возможность посылки запроса на установление связи как от DTE к DCE, так и от DCE к DTE. Фрагмент отображается сетью Пет-

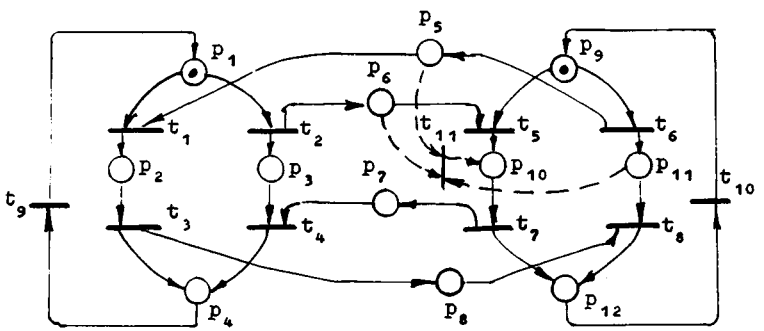


Рис. I5

ри (рис.I5 без перехода t_{11}), состоящей из двух компонент. Последовательность срабатываний $\sigma_1 = t_2, t_5, t_7, t_4$ отображает запрос-ответ от DTE к DCE, а $\sigma_2 = t_6, t_1, t_3, t_8$ - запрос-ответ в обратном направлении. Если обе компонентные машины независимо друг от друга пошлют сигналы запроса, то система окажется в тупиковой ситуации. Сеть Петри отображает эти события последовательностью $\sigma = t_2, t_6$ (или $\sigma' = t_6\ t_2$), которая приводит к тупиковой маркировке M = = (0,0,I,0,I,I,0,0,0,0,I,0). Эта ситуация предусмотрена в спецификации, причем предусмотрено приоритет установления связи отдать компоненте DTE. Следовательно, необходимо ввести такое событие

(переход t_{11}), которое в случае наличия в линии обоих запросов, переведет систему в состояние, соответствующее принятому запросу от DTE (маркировка M = (0,0,I,0,0,0,0,0,0,I,0,0)). Легко проверить, что полученная сеть (рис.I5 с переходом t_{11}) – живая и безопасная, что гарантирует общую корректность системы.

Покажем теперь, какие условия частной корректности могут быть проверены путем анализа инвариантов сети Петри. В соответствии с двумя классами инвариантов выделим два класса условий частной корректности: I) условия, связанные с наличием циклов в процедурах взаимодействий, и 2) условия, выражающие ограничения на совместимость состояний (сигналов, данных), одновременно присутствующих в системе.

Следующие примеры иллюстрируют отображения частных условий корректности первого класса.

I. Наличие запрос-ответного цикла установления связи. Например, в протоколе из примера 9 (рис.I5) должны существовать циклы событий, необходимых для установления связи как от DTE и DCE, так и от DCE к DTE. Эти циклы соответствуют T-инвариантам $\bar{\sigma}_1 = (0,I,0,I,I,0,I,0,I,I)$ и $\bar{\sigma}_2 = (I,0,I,0,0,I,0,I,I,I)$.

2. Выполнение заданных в спецификации отношений количеств разных событий в циклической процедуре. Так, в примере 5 (рис. II) должен существовать цикл, состоящий из семи событий посылки сообщений, одной посылки запроса и одного приема ответа. Это соответствует T-инварианту $\sigma = (I,I,7,8)$. Равенство

$$C\bar{\sigma} = \begin{vmatrix} -I & 0 & -I & I \\ I & -I & 0 & 0 \\ -7 & 0 & I & 0 \\ 0 & 7 & -I & 0 \\ 0 & I & I & -I \end{vmatrix} \cdot \begin{vmatrix} I \\ I \\ 7 \\ 8 \end{vmatrix} = \begin{vmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{vmatrix}$$

убеждает в существовании нужного инварианта.

3. Согласованность циклических процедур (отсутствие несправедливых циклов). Пример 7 (рис.I3) иллюстрирует наличие несправедливой последовательности $\sigma = (t_1,t_2,t_4,t_3,t_6)^*$, которой соответствует T-инвариант $\bar{\sigma} = (I,0,I,I,I,I,0,0)$. Легко проверить, что этого инварианта не будет, если ввести условие взаимного исключения событий запрос-от –

вет от DCE (t_1,t_2 или t_1,t_3) и события ухода на прерывание DTE (t_5,t_6). В сети Петри это условие отображается введением позиции p_{10}, такой что ${}^{\bullet}p_{10} = \{t_2,t_3,t_6\}$, $p_{10}^{\bullet} = \{t_1,t_5\}$.

Частные условия корректности второго класса выявляют путем анализа P-инвариантов. Следующие типы условий могут служить иллюстрацией.

I. Суммарное количество свободных и занятых ячеек буфера. Так, в сети Петри (рис.I0) $M(p_5) + M(p_6)=n$ при всех достижимых маркировках, что соответствует P-инварианту $y = (0,0,0,0,I,I)$. В этом нетрудно убедиться, проверив условие

$$yC = \underline{0\,0\,0\,0\,I\,I} \cdot \begin{vmatrix} -I & 0 & I & 0 \\ I & 0 & -I & 0 \\ 0 & -I & 0 & I \\ 0 & I & 0 & -I \\ -I & I & 0 & 0 \\ I & -I & 0 & 0 \end{vmatrix} = \underline{0\,0\,0\,0\,0\,0}$$

2. Отсутствие переполнений и возврат в исходное состояние компоненты взаимодействия могут быть проконтролированы проверкой того, составляет ли множество позиций компонентной сети Петри консервативную составляющую в объединенной сети, отображающей работу всей системы. Так, для компонентных сетей Петри из примера (рис.I0) консервативными составляющими являются $P_1 = \{p_1,p_2\}$ и $P_2 = \{p_3,p_4\}$.

3. Условие взаимного исключения. Необходимо, чтобы позиции, отображающие взаимно исключающие состояния, в сумме имели не больше одного маркера. Фрагмент сети, обеспечивающий это (арбитр), имеет P-инвариант $y = (I,I,I,0,0)$, для которого $K = M(p_1) + M(p_2) + M(p_0) = 1$, (см. рис.9).

§6. Существующие средства анализа свойств сети Петри

Существует три подхода к анализу свойств сетей Петри: I) построение графа (или дерева [II]) достижимости, 2) исследование структуры графа сети Петри и 3) поиск инвариантов.

Прежде чем коротко охарактеризовать эти подходы, следует указать, что сложность алгоритмов во всех случаях экспоненциально зависит от числа вершин сети. Поэтому много внимания уделялось приемам уменьшения алгоритмической сложности. Известно два таких приема: редукция и декомпозиция.

Редукция заключается в следующем. Определяется набор преобразований сети Петри, уменьшающих число ее вершин и сохраняющих анализируемые свойства. Преобразования применяются многократно до тех пор, пока не получается в результате такая сеть, что ни одно преобразование редукции неприменимо. Эта нередуцируемая сеть подвергается затем анализу одним из известных способов. Известны работающие алгоритмы и программы редукции для обобщенных [II] и ординарных сетей Петри [I8]. Опыт их применения и специальный анализ редуцируемости показывают большую эффективность этого приема.

Приемы декомпозиции разработаны пока только на уровне идей и пожеланий [2,II]. Очевидно, их применение целесообразно для частных случаев, когда компоненты декомпозиции заранее предопределяются в процессе построения сети, и когда при этом удается доказать, что объединенная сеть сохраняет свойства живости и ограниченности компонентных составляющих. Именно такие частные случаи характерны для сетей Петри, отображающих запрос-ответные взаимодействия. Однако, к сожалению, нет пока ответа на вопрос: какой характер связи между живыми и ограниченными компонентными сетями сохраняют эти свойства для объединенной сети.

Анализ свойств маркированной сети Петри по графу достижимости дает наиболее полную характеристику ее поведения. Это понятно, поскольку граф достижимости перечисляет все маркировки и все последовательности срабатываний. Свойство живости распознается по наличию последовательностей срабатываний, начинающихся в каждой M и ведущих в M_0. Свойство ограниченности - по отсутствию двух маркировок M' и M", сопровождающих одну и ту же последовательность срабатываний и таких, что M' > M". Существующие алгоритмы и их машинные реализации [II,I8] ориентированы на распознавание именно этих двух свойств, самых нужных, так как они определяют общую корректность системы. Кроме того, очевидно, что эти алгоритмы определяют достижимость или недостижимость заданной маркировки. Реализованные алгоритмы и программы оказались вполне работоспособными и при хранении достижимых маркировок на внешних носителях могут использоваться для сетей Петри с числом вершин $|П| + |Σ| \leq 1000$.

Методы, использующие анализ структуры сети Петри для проверки ее на живость и ограниченность, известны для класса сетей со свободным выбором. Этот класс сетей Петри имеет достаточно широкое применение, поскольку моделирует параллельные граф-схемы. Сеть, отображающая протокол передачи данных, принадлежит классу сетей

со свободным выбором, если каждая компонента находится в этом классе и объединение компонент выполнено совмещением входных позиций одной компоненты с выходными позициями другой. Сети со свободным выбором исследованы наиболее полно с точки зрения живости и безопасности [I2]. Разработанные на основе результатов,изложенных в [I2],алгоритмы и программы [20] как по времени,так и по памяти существенно экономичнее, чем алгоритмы построения графа достижимости. Их сложность для класса встречающихся на практике сетей Петри близка к полиномиальной (со степенью $n \approx 3$).

Методы анализа сетей Петри, связанные с поиском инвариантов, заключаются в определении базовых множеств решений уравнений вида (3) и (5). Эти решения перечисляют бесконечные последовательности срабатываний переходов и консервативные подмножества позиций, тем самым определяя многие частные свойства взаимодействий в системе. Известно, что существует программная реализация этих методов. Программы разработаны на основе методов линейной алгебры (алгоритмы Гомори) и применимы к сетям с числом вершин ≤ 1000 [II].

Заключение

Из изложенного следует, что обычные (неинтерпретированные) сети Петри могут быть использованы для проверки общей корректности протокола, а также многих случаев частной корректности. В то же время ясно, что они бесполезны для выявления каких-бы то ни было количественных отношений.

Удобство работы с аппаратом сетей Петри во многом зависит от трудоемкости процедуры построения сети, а это,в свою очередь, определяется степенью формализации задания протокола. Ясно,что формальное описание, основанное на любой сетевой модели, приведет к облегчению перехода к обычной сети Петри и к возможности использования имеющегося математического обеспечения их анализа для проверки корректности протоколов передачи данных.

Литература

I. ЗАЙЦЕВ С.С., МУРАДЯН Н.А. Методы проверки правильности функционирования сетевых протоколов. -М., 1983. -66 с. (Препринт/ АН СССР).

2. SUNSHINE C. Formal Techniques for Protocol Specification and Verification.-Computer,1979,Sept.,p.20-25.

3. WEST C.H. General Technique for Communication Protocols Validation.-IBM Journal of Research and Development, 1978,v. 22, N 4,p.393-404.

4. DIAZ M. Modeling and Analysis of Communication and Cooperation Protocols Using Petri Nets Based Models. - Computer Networks,1982,v.6,N 6,p.419-443.

5. ПИТЕРСОН Дж. Теория сетей Петри и моделирование систем. -М.: Мир, 1984. -264 с.

6. КОТОВ В.Е. Сети Петри. -М.: Наука, 1984. -160 с.

7. ГОБЗЕМИС А.Ю., КИЗУБ В.А. Применение расширенных сетей Петри для моделирования интерфейсов многомашинных систем.-В кн.: Вычислительные сети коммутации пакетов, ч.I,1981, с.103-106.

8. АНИСИМОВ Н.А. Средства формального описания сервиса и протоколов сетей ЭВМ с использованием сетей Петри. -Владивосток,Б.и, 1983. -32 с. (Препринт/ИАПУ ДВНЦ АН СССР).

9. GENRICH H.J., LAUTENBACH K. System Modeling with High Level Petri Nets. - Theoretical Computer Science,1981,v.13, p.109-136.

10. MERLIN P.M., FARBER D.J. Recoverability of Communication Protocols.- Implication of a Theoretical Study. - IEEE Transactions on Communications, 1976,Sept.,p.1036-1043.

11. BRAMS G.W. Reseaux de Petri: Theorie et Pratique. - Paris: Masson,1983. - V.1, 184 p.; V.2, 160 p.

12. HACK M. Analysis of Production Schemata by Petri Nets. - In: Techn.Rept.94. Project MAC, MIT, Cambrige,1972,p.119.

13. WEST C., ZAFIROPULO P. Automated Validation of a Communication Protocol: the CCITT X.21 Recommendation.- IBM Research and Development,1978,v.22, Jan.,p.60-71.

14. DONNAN R.A., KERSEY J.R. Synchronous Data Link Control: A Perspective.- IBM System Journal,1974,v.13,p.140-162.

15. MEMMI G., ROUCAIROL G. Linear Algebra in Net Theory. -In: Net Theory and Applications, Lecture Notes in Computer Science, 1979,v.84, Springer,p.213-224.

16. KWONG Y.S. On the Absence of Livelocks in Parallel Programs. - In: Semantics of Concurrent Computations: Lecture Notes in Computer Science,1979,v.70,p.172-190.

17. БАНДМАН О.Л. Корректность асинхронных параллельно-потоковых систем обработки данных. -Программирование, 1985, №6,с.20-34.

18. АНИШЕВ П.А. Редуцируемость сетей Петри. - Программирование, 1982, № 4, с.36-42.

19. ЕСИКОВА Т.Н. Алгоритм построения множества достижимых маркирований для анализа сетей Петри. -В кн.: Однородные вычислительные системы из микро-ЭВМ (Вычислительные системы, вып. 97). Новосибирск, 1983, с.28-52.

20. АНИШЕВ П.А. Один способ анализа корректности граф-схем алгоритмов. - Программирование, 1981, № 1, с. 20-28.

Поступила в ред.-изд.отд.
22 апреля 1985 года