



УДК 519.712; 001.891.573:681.3.07

В. В. Кузьмук, д-р техн. наук
Отделение гибридных моделирующих
и управляющих систем в энергетике
Ин-та проблем моделирования
в энергетике им. Г.Е.Пухова НАН Украины
(Україна, 03150, Київ, ул. Щорса, 15, корп.1, оф. 35,
тел. (044) 5993113, Email: imt-kiev@mail.ru),

О. О. Супруненко, канд. техн. наук
Черкасский национальный университет им. Б. Хмельницкого
(Україна, 18031, Черкаси, б-р Шевченко, 81,
тел. (0472) 450475)

Расширение функциональных возможностей алгоритмического аппарата сетей Петри при моделировании параллельных процессов

Рассмотрены интерпретации и модификации безопасных сетей Петри как средства моделирования последовательных и параллельных алгоритмов. Поставлена задача распараллеливания последовательных алгоритмов для повышения эффективности их работы и приведены критерии анализа для их распараллеливания. Рассмотрен пример распараллеливания последовательного алгоритма, представленного функциональной модификацией управляющих сетей Петри.

Розглянуто інтерпретації та модифікації безпечних мереж Петрі як засоба моделювання послідовних та паралельних алгоритмів. Поставлено задачу розпаралелювання послідовних алгоритмів з метою підвищення ефективності їхньої роботи і наведено критерії аналізу послідовних алгоритмів для їх розпаралелювання. Розглянуто приклад розпаралелювання послідовного алгоритму, представленого функціональною модифікацією управляючих мереж Петрі.

Ключевые слова: моделирование, параллельные процессы, сети Петри.

Быстрое развитие многопроцессорной компьютерной техники обусловило возникновение задач эффективного использования вычислительных ресурсов: более равномерной загрузки процессоров, распараллеливания вычислений на уровне задач и потоков команд управления. Данные задачи можно решать распараллеливанием вычислений на программном, процедурном уровне, уровне инструкций и на битовом уровне. Организация параллельной обработки данных на уровне команд управления в операционной системе позволяет достичь прироста производительности до 20 %,

при этом коэффициент равномерности загруженности процессоров остается достаточно низким. Наиболее перспективным для повышения эффективности обработки информации является распараллеливание на уровне задач и процессов. В этом случае можно повысить эффективность в три-четыре раза по сравнению с предыдущим случаем, а равномерность загруженности процессоров может возрасти в несколько десятков раз.

Поэтому в настоящее время актуальна задача анализа последовательных алгоритмов для повышения эффективности их обработки. Один из путей решения этой задачи — распараллеливание последовательных алгоритмов для их обработки на определенном числе процессоров. Для эффективной реализации задачи распараллеливания алгоритмов требуется алгоритмический аппарат, позволяющий выполнять анализ последовательных и построение параллельных алгоритмов [1].

Теоретическим основанием для решения поставленной задачи могут быть графоаналитические методы анализа последовательных алгоритмов [2], среди которых наиболее известны граф-схемы алгоритмов, теория автоматов и графовые модели. В настоящее время для анализа алгоритмических конструкций параллельных процессов наибольшее распространение получили графовые модели: графы состояний и сети Петри [2, 3].

Сети Петри (Petri Netze (PN)), применяемые в данном случае, обеспечивают однозначное математическое описание характеристик элементов объекта моделирования и автоматизированный анализ топологии и функционирования объекта во время моделирования. За время существования PN появилось немало новых интерпретаций и модификаций сетей [1—3]. Модифицированные PN являются достаточно простым графическим инструментом моделирования параллельных процессов. Они состоят из двух типов вершин — вершин мест и вершин переходов, и двух типов дуг — позитивных и инверсных. Модификации PN имеют различные характеристики элементов и правила функционирования сети.

Известные интерпретации и модификации сетей Петри, используемые для алгоритмического описания и моделирования параллельных процессов, по особенностям реализации элементов можно разделить на три группы [4]:

1) оценочные сети Петри (BPN), различные интерпретации которых имеют целое число меток $N (N \geq 1)$ в вершинах мест, а целочисленные дуги — вес, равный числу меток, которые одновременно могут проходить по ним из одной вершины в другую;

2) числовые (EN) и макрочисловые (MEN) сети Петри, имеющие несколько типов вершин переходов, которые срабатывают при заданном числе меток N в сети, а структура вершин макропереходов может изменяться;

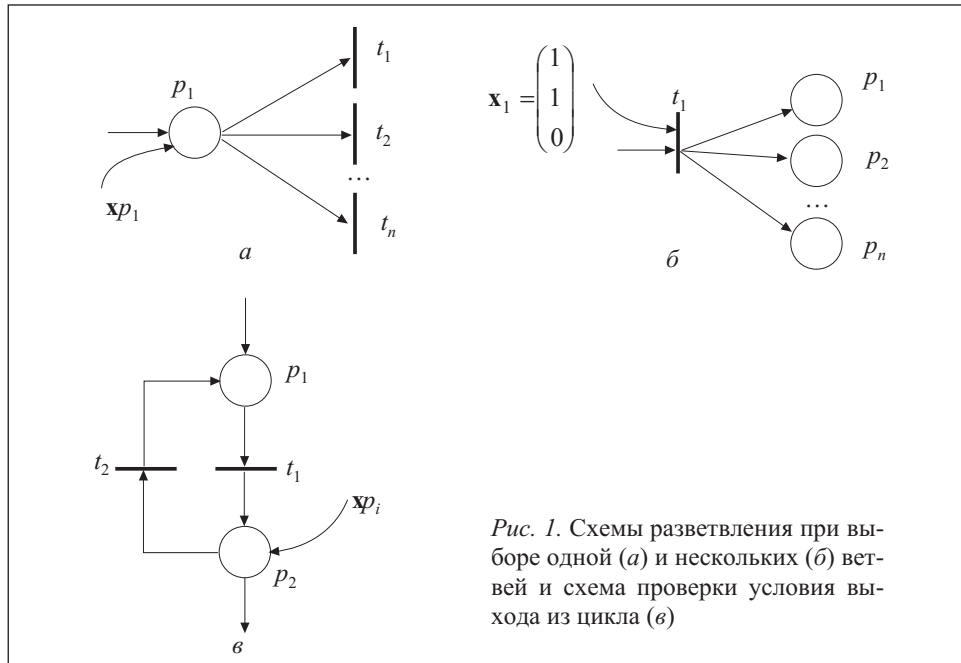
3) безопасные сети Петри (SPN), различные интерпретации которых могут иметь в каждой вершине места не более одной метки.

Оценочные сети Петри имеют достаточно сложные правила построения. Большое число меток и разновесовые оценочные дуги даже в моделях с небольшим числом элементов (до 30 единиц) ухудшают наглядность восприятия сетевой модели. Сложность формализации правил построения оценочных сетей усложняет автоматизацию процесса построения и отладки моделей. Отдельно можно выделить цветные сети Петри, которые являются разновидностью BPN с дополнительными функциями относительно числовых потоков и числа меток в вершинах мест. Их широкое применение обусловлено возможностью моделировать и анализировать в динамике поведение и характеристики нескольких потоков с различными характеристиками.

Разнообразие типов вершин переходов в EN и MEN позволяет расширить функциональность создаваемых моделей, но при потребности в добавлении новых функций возникает необходимость создания новых типов вершин переходов. Эта проблема может быть решена с помощью подготовки библиотек элементов, однако их наполнение и расширение требует дополнительной согласованной работы исследователей и программистов. Кроме того, чем большее число элементов, имеющих уникальные свойства, используется в модели, тем сложнее ее создавать, проверять на адекватность и анализировать функционирование. Специфичны и правила их применения к различным моделям.

Наибольшее распространение в настоящее время получили управляющие сети [2, 4], как интерпретация SPN [5]. Простота их построения и гибкость правил налаживания и управления процессом имитации работы объекта моделирования создают условия для автоматизации анализа, синтеза и моделирования параллельных и последовательно-параллельных алгоритмов. Кроме того, выбор интерпретации SPN обусловлен особенностями управления процессом обработки данных.

Недостатком PN считают отсутствие в базовой теории понятия процесса и неоднозначность выполнения параллельных процессов в построенной модели [6]. Однако с этим нельзя согласиться, поскольку процесс в PN ассоциируется с движением метки по элементам сети. Неоднозначность выполнения параллельных процессов контролируется перечнем правил их построения, управляющими элементами и дополнительными функциональными элементами сетей, которые имеют, например, управляющие сети Петри [4]; нарушение правил построения и функционирования проверяется по наличию критических свойств PN (динамических и статических), при устранении которых алгоритм имеет четкий однознач-



ный порядок выполнения. Указанная неоднозначность отражает и преимущество теории PN, поскольку она позволяет моделировать системы с независимыми во времени (асинхронными) событиями, проводить статистический анализ исследуемых объектов.

Создание новой модификации управляющих сетей Петри связано с потребностью моделирования параллельных алгоритмических конструкций, в которых необходимо управлять выбором одной или нескольких выходных ветвей из определенной вершины. В качестве вершины разветвления выбирают вершину места при выборе одной из веток (рис. 1, а), которая соответствует реализации логической функции «или», либо вершину перехода — при выборе нескольких веток (рис. 1, б), что соответствует выборочной реализации логической функции «и».

Для обеспечения универсальности модели приемлемым выбором обоих видов разветвлений будет вершина перехода. Однако при реализации бинарного выбора, часто встречаемого в алгоритмических конструкциях, более наглядным вариантом считается выбор вершины места (см. рис. 1, а, в), так как она ассоциируется с проверкой условий (окончания операции, окончания цикла и др.).

В процессе выбора веток для дальнейшего выполнения алгоритма используют управляющие функции x_{p_i} и x_{t_i} , которые являются векторами

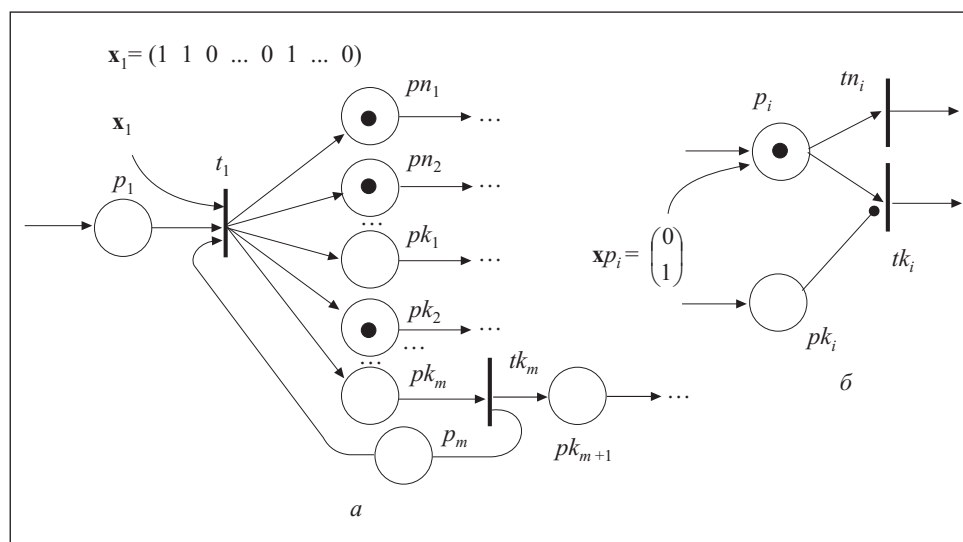


Рис. 2. Модели управления выбором нескольких процессов из множества (а) и одного процесса из множества возможных (б)

размерности, определяемой количеством выходных ветвей. Вектор x_{p_i} всегда имеет единичное значение только в одной переменной, все другие переменные равны нулю, что соответствует выбору одной из множества веток. Вектор x_i может иметь значение 0 или 1 во всех своих переменных, что соответствует выбору нескольких ветвей из множества возможных, фактически он определяет дальнейшую частичную разметку вершин мест, инцидентных вершине перехода в разветвлении.

При построении параллельного алгоритма на основе последовательного необходимо выделить участки алгоритма, которые могут быть выполнены независимо или условно-независимо. Для распараллеливания необходимо проанализировать последовательный алгоритм на наличие:

- 1) источников начальных данных и промежуточных результатов;
- 2) элементов в алгоритме, имеющих связь по данным и по командам;
- 3) элементов, управляемых внешними командами;
- 4) альтернативности выбора участков алгоритма.

По окончании анализа можем выделить условно-независимые участки алгоритма, отображаемые параллельными ветками. Для создания и моделирования параллельного алгоритма можно воспользоваться функциональной модификацией управляющих сетей Петри, которая позволяет реализовать логические конструкции «или» и «и» (в полном и выборочном вариантах). Логическая операция «не» в виде инверсных дуг реализована в SPN [4] и наследуется их интерпретациями и модификациями, в том числе и в функциональной модификации управляющих сетей [5].

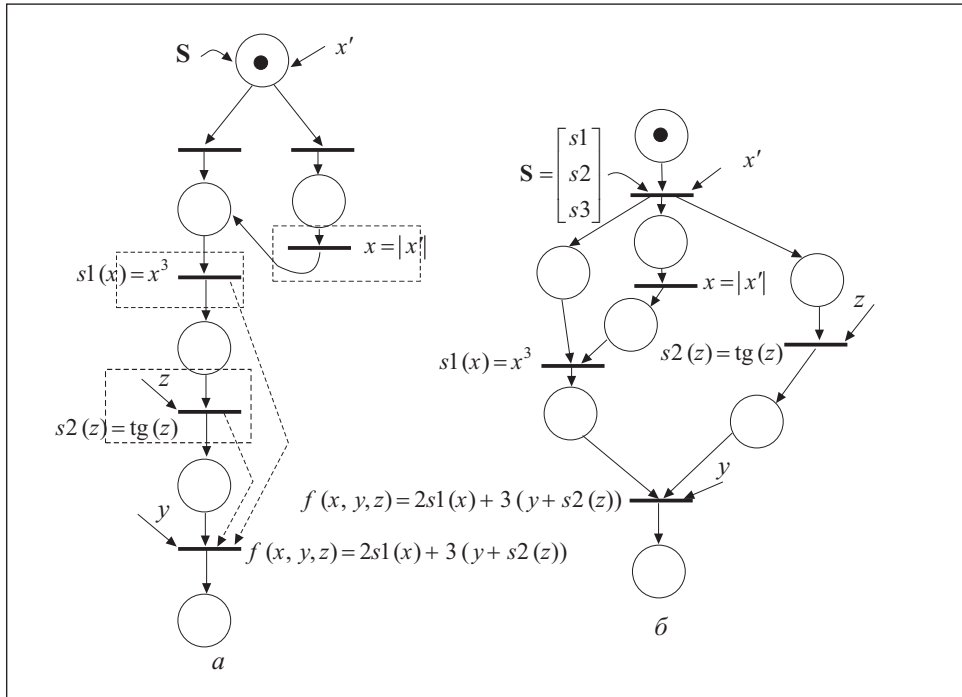


Рис. 3. Последовательный алгоритм вычисления функции $f(x, y, z) = 2|x^3| + 3(y + \text{tg}(z))$ с разветвлением (а), представленный функциональной модификацией PN, и его преобразование в параллельный алгоритм (б)

При моделировании систем управления в некоторых случаях необходимо одновременное выполнение нескольких рабочих и контролирующих процессов, что соответствует выборочной реализации логической функции «и». Для одновременного запуска таких процессов в функциональной модификации PN используется управляемая вектором x_1 вершина перехода (рис. 2, а). При реализации бинарных операций применяются управляемые вершины мест (рис. 2, б). Данная реализация разветвления позволяет управлять выполнением алгоритмов в каждой ветви отдельно через другие конструкции сети, а не только с помощью управляемой вектор-функции.

Таким образом, вариант разветвления из вершины места позволяет моделировать выбор каждой ветки в разветвлении отдельно с помощью конструктивных элементов сети — внутренними средствами PN (например, инверсная дуга из вершины места pk_i (см. рис. 2, б)). В разветвлениях из вершины перехода внутренний контроль за выполнением операций в параллельных ветвях является общим для всех ($tk_m \rightarrow p_m \rightarrow t_1$) и может осуществляться в местах разветвления только с помощью вектор-функции x_j .

Из приведенных примеров видно, что для построения параллельных алгоритмов функционирования подобных систем необходимо четко разграничить внутренние и внешние функции управления: внутренние функции реализуются через топологию PN, а внешние — посредством вектор-функций $x p_i$ и x_i .

На рис. 3 представлены последовательный алгоритм с разветвлением и его параллельная версия. Алгоритмы представлены функциональной модификацией управляющих PN, которая позволяет проводить выбор одной из веток в разветвлении из вершины места (рис. 3, а) и нескольких альтернативных веток — из вершины перехода (рис. 3, б).

При распараллеливании последовательного алгоритма проводим его анализ и определяем условно-независимые участки, которые на рис. 3, а выделены штриховыми линиями, выбираем тип вершины для разветвления. Поскольку данная вершина, увеличивая число меток в сети, передает метки не во все выходные вершины, по определению такая операция соответствует вершине перехода с управляющей вектор-функцией. Анализируем также внешние управляющие функции и функции введения начальных данных.

В последовательном алгоритме с разветвлением (см. рис. 3, а) альтернативный путь выбран по управляющей функции S , которая принимает значения 0 или 1 в зависимости от верного или ложного выполнения логического выражения. При преобразовании в параллельный алгоритм модификация управляющих сетей позволяет провести выбор варианта расчета и распараллеливания потоков в одной вершине перехода по вектор-функции $S = [s_1, s_2, s_3]^T$. Это сокращает число вершин переходов, которые осуществляют управляющие функции, что повышает эффективность преобразования модели алгоритма в программный код.

При использовании моделирующей среды на основании функциональной модификации PN для анализа последовательных алгоритмов и их распараллеливания решается непростая задача создания параллельных и параллельно-последовательных алгоритмов в моделях, логика реализации которых, даже в последовательном варианте, чрезвычайно сложна [6]. Аппарат PN позволяет представить в виде графа и проанализировать в автоматизированном режиме алгоритмические конструкции, что особенно важно для анализа наиболее сложных конструкций управления — разветвлений. Таким образом, достигается эффективный анализ и синтез параллельных алгоритмов, что позволяет более равномерно нагружать имеющиеся процессоры при обработке преобразованных параллельно-последовательных программ.

Кроме указанных преимуществ для первичного анализа построенной модели в полуавтоматическом режиме могут быть использованы стати-

ческие и динамические критические свойства PN, а также созданные на их основе функциональные конструкции корректирования построенной модели [5]. Это одно из важных достоинств PN [4], которое позволяет применять их для анализа больших программных кодов.

Аппарат PN имеет наглядные графические средства для построения логических абстракций, позволяет отлаживать модель и алгоритм внешнего управления, применяется для построения и исследования моделей большой размерности, имеет определенные примитивы для формирования на основе графовой модели программных кодов, что повышает эффективность создания программного обеспечения для параллельных микропроцессорных систем.

Вывод. Предложенная новая модификация PN позволяет расширить алгоритмические возможности отображения параллельных и параллельно-последовательных процессов, определить правила построения и автоматизированной проверки моделей, что способствует повышению эффективности создания и моделирования параллельных алгоритмов, т. е. упрощает анализ моделей и позволяет автоматизировать процесс отладки и окончательного формирования алгоритмов большой размерности.

Interpretations and modifications of the dangerless Petri nets have been considered as the means of modeling of sequential and parallel algorithms. The authors have posed the problem of parallelizing the sequential algorithms for raising the efficiency of their work and have given the analysis criteria for their parallelizing. An example of parallelizing the sequential algorithm represented by a functional modification of the controlling Petri nets has been considered.

1. Хопкрофт Д. С., Мотвани Р., Ульман Д. Д. Введение в теорию автоматов, языков и вычислений. 2-е изд. / Пер. с англ. — М. : Изд. дом «Вильямс», 2002. — 528 с.
2. Касьянов В. Н., Евстигнеев В. А. Графы в программировании: обработка, визуализация, применение. — С-Пб. : БВХ - Петербург, 2003. — 1104 с.
3. Вербицкий И. Б. Семантические модели в теории параллелизма. — Новосибирск: Изд-во ИСИ СО РАН, 2000. — 522 с.
4. Васильев В. В., Кузьмук В. В. Сети Петри, параллельные алгоритмы и модели мультипроцессорных систем. — Киев : Наук. думка, 1990. — 216 с.
5. Кузьмук В. В., Супруненко О. О. Принципы побудови моделюючого середовища для відображення функціонування технічних систем різної фізичної природи // ИТ в XXI веке: Сб. докладов и тезисов II-го Междунар. науч.-практ. форума / Под ред. акад. НАНУ В. В. Пилипенко и др. — Днепропетровск: ИПК ИКЦ УГХТУ, 2004. — С. 95—96.
6. Богатырев Р. В. Об автоматном и асинхронном программировании // Открытые системы. — 2001. — № 3. — С. 68—69.

Поступила 03.09.08;
после доработки 21.04.09

КУЗЬМУК Валерий Валентинович, д-р техн. наук, проф. Отделения гибридных моделирующих и управляющих систем в энергетике Ин-та проблем моделирования в энергетике им. Г. Е. Пухова НАН Украины. В 1974 г. окончил Киевский технологический ин-т легкой промышленности. Область научных исследований — теория алгоритмов (сети Петри), теория моделирования, биологическая и медицинская кибернетика; прикладная математика, мультипроцессорные системы и многомашинные комплексы; параллельное программирование.

СУПРУНЕНКО Оксана Александровна, канд. техн. наук, доцент кафедры кибернетики Черкасского национального университета им. Б. Хмельницкого. В 1993 г. окончила Черкасский инженерно-технологический ин-т. Область научных исследований — параллельные алгоритмы, сети Петри, параллельное программирование; биологическая и медицинская кибернетика; численные методы в информатике; элементы инженерной и компьютерной графики.