УДК 004.7

# РЕШЕНИЕ УРАВНЕНИЯ СОСТОЯНИЙ СИСТЕМ УПРАВЛЕНИЯ

© 2013 А.А.Смагин, В.В. Кожевников, И.В. Круглова

Ульяновский государственный университет

Поступила в редакцию 21.06.2013

Статья посвящена решению систем линейных алгебраических уравнений, которые описывают поведение цифровых автоматов. Подробно рассмотрен частный случай, когда учитывается специфика описания функционирования цифрового автомата. Приводится пример решения задачи.
Ключевые слова: фундаментальные уравнения Мураты, цифровые автоматы, сети Петри, моделирование, решение системы уравнении.

Сети Петри являются мощным инструментом для исследования самых разнообразных систем, которые позволяют осуществлять моделирование для получения информации о структуре и динамическом поведении. Такая информации необходима для проведения последующего мониторинга, контроля и диагностики.

Сети Петри удобны для проведения анализа программного обеспечения, протоколов, цифровых автоматов, составляющих аппаратную часть моделируемой системы.

В настоящей статье делается акцент на моделирование цифровых устройств, представляющих собой множество логических схем с памятью (цифровые автоматы), как основы процессорных систем, реализующих общий функционал всей системы. Для описания модели цифровых автоматов выбрана система линейных алгебраических уравнений [4], представленная в матричном виде, который наиболее удобен для анализа моделей Петри большой размерности. Матричные системы уравнений представляют собой фундаментальные уравнения сетей Петри. Решение этой системы интерпретируются как векторы счета допустимых последовательностей срабатывания переходов и поэтому должны быть целыми неотрицательными числами, что задает специфику задачи [1].

Фундаментальные уравнения сетей Петри (уравнения Мураты) записываются следующим образом:

$$\overline{\mathbf{x}} \cdot \mathrm{A} = \Delta\overline{\mu},$$

где $\Delta\overline{\mu} = \mu - \Delta\overline{\mu}_0$, $\overline{\mathbf{x}}$ – вектор счета срабатываний переходов (вектор покрытия сети), $\mu$ – маркировка, А – транспонированная матрица инцидентности. Разрешимость фундаментального уравнения в неотрицательных целых числах является необходимым условием достижимости заданной маркировки, которая отображает множество позиций сети в множество неотрицательных целых чисел и служит для оценки достижимости устойчивых состояний цифрового автомата.

Решением однородной системы линейных уравнений называют такую их совокупность через которую линейно выражаются все остальные решения. Если ранг матрицы А равен числу неизвестных (r=n), то система имеет только нулевое решение. Если r<n, то бесконечное множество, причем фундаментальная система состоит из (n-r) векторов $\overline{\mathbf{x}}$. Отсюда все инварианты $\overline{\mathbf{x}}$ для маркированной сети можно получить из (n-r) базисных решений.

Система $\overline{\mathbf{x}} \cdot \mathbf{A} = \Delta\overline{\mu}$ имеет решение тогда и только тогда, когда вектор $\Delta\overline{\mu}$ ортогонален любому решению y соответствующий однородной системы уравнений $\mathbf{A}^* \cdot y = \mathbf{0}$, где $\mathbf{A}^*$ – сопряженная матрица ($\mathbf{A}^* = \mathbf{A}^T$) и все компоненты вектора $\overline{\mathbf{x}}$ были определены.

Целочисленный вектор $\overline{\mathbf{x}}$ является решением системы линейных уравнений и называется Р инвариантом, который характеризует все достижимые позиции сети с сохранением ее некоторых свойств – например свойства распределения меток по позициям сети. Однако компоненты вектора $\overline{\mathbf{x}}$ могут быть не только целыми числами, но и переменными, которые необходимо вычислять на каждом шаге решения системы уравнений. Эта задача может решаться в системе Maple, но этому препятствуют такие неудобства как использования специальных нотаций Maple и написания с ее помощью процедуры вычисления неизвестных. Кроме того, использование Maple в системе других программных модулей затруднено. Требуется ручной ввод исходных данных и вывод результатов. Аналогичные проблемы с другими математическими электронными сервисами.

*Смагин Алексей Аркадьевич, профессор, доктор технических наук, заведующий кафедрой телекоммуникационных технологий и сетей. E-mail: smaginaa1@mail.ru*
*Кожевников Валерий Владимирович, кандидат технических наук, доцент кафедры телекоммуникационных технологий и сетей. E-mail: vvk2861955@mail.ru*
*Круглова Ирина Владимировна, аспирантка кафедры «Прикладная математика» E-mail: pm@ulsu.ru*

Альтернативный вариант использовать вычислительные мощности современных компьютеров и проводить полный перебор всех значений $\overline{\mathbf{x}}$ с поиском решения. Из вариантов перебора $\overline{\mathbf{x}}$ определяется тот, который удовлетворяет осуществлению анализируемого перехода. Такие процедуры необходимо проводить для каждого вектора $\overline{\mathbf{x}}$ и общее количество переборов в простейшем случае подчиняется мультипликативному закону.

Таким образом, для решения фундаментального уравнения необходимо выполнить определенные условия, которые касаются вектора $\overline{\mathbf{x}}$. Как указывалось выше, часть вектора $\overline{\mathbf{x}}$ может быть задана заранее и она может вычисляться программно исходя из анализа схемы сети Петри. Например, на переходе может срабатывать только один выход. В этом случае для этого перехода в соответствующем месте вектора должна стоять единица, в остальных ноль. Таким образом, вектор $\overline{\mathbf{x}}$ имеет строгую определенность позиций 0 и 1 и его генерация должна учитывать реальные процессы в цифровом автомате. Кроме этого, значения маркировок также влияют на решение фундаментального уравнения. Для решения общее число маркировок равное числу позиций можно уменьшить и часть их заменить нулями. Нули получают только те позиции, которые отвечают за внутренние состояния автомата (связанные с дугами обратной связи). Такой прием позволят упростить решение и свести его к решению однородного уравнения. Ниже такой подход, который учитывает особенности цифровых автоматов, рассматривается на конкретных примерах.

Процедура решения уравнения состояний цифровых автоматов  включает себя следующие этапы [5, 6]:

1. Представление исходной структурной схемы путем интерпретации входов и выходов логических схем и структурных компонентов позициями маркированного графа, а самих компонентов и линий соединения составными и простыми переходами соответственно.

2. Переход от графической формы представления логических схем к их математическому представлению в виде матрицы инцидентности

3. Построение равнения состояний цифровых автоматов.

Система $Ax=\Delta\mu$ имеет решение тогда и только тогда, когда вектор $\Delta\mu$ ортогонален любому решению $y$ соответствующей однородной системы уравнений $A^*y=0$, где $A^*$ – сопряженная матрица ($A^*=A^T$). Тадао Мурата в своей работе [3] представляет множество решений однородной системы (с матрицей $A$ размерности $m$ на $n$) в виде выражения

$$B_f = [I \,|\, -A_{11}^T (A_{12}^T)^{-1}], \qquad (1)$$

где $I$ – единичная матрица порядка $m$-rank($A$), а $A_{12}$ – невырожденная квадратная матрица порядка rank($A$), $A = \begin{bmatrix} A_{11} A_{12} \\ A_{21} A_{22} \end{bmatrix}$.

Таким образом, условие ортогональности вектора $\Delta\mu$ любому решению однородной системы $A^*y=0$ представляется в виде $B_f \cdot \Delta\mu = 0$.

4. Генерация векторов x.

Каждому переходу $i$, по которому осуществляется перебор, соответствуют $2^{Ki}$ неизвестных $x_{i1}$, $x_{i2}$, ..., $x_{i,2^{Ki}}$, где $Ki$ – число входных позиций. Перебирая по $h$ от 1 до $2^{Ki}$, получаем следующие варианты значений неизвестных для $i$-го перехода:

$$x_{ij} = \begin{cases} 1, j = h \\ 0, j \neq h \end{cases}, j = 1, \dots, 2^{Ki}. \qquad (1)$$

Пусть $P$ – множество номеров переходов, по которым производится перебор значений $x_{ij}$. Тогда количество комбинаций переходов определяется выражением: $\prod_{i\in P} 2^{Ki}$, а их генерацию можно провести с помощью следующей рекурсивной процедуры:

```
00: Generate_combination (i)
01: по h от 1 до 2^Ki:
02:         присвоить значения x_ij
            по формуле (2)
03:         если i – не последний переход
            из множества P:
04:                 Generate_combination(next(i))
05:         иначе: сохранить текущие
            значения x_ij
```

Процедура Generate_combination запускается для первого номера $i$ из множества $P$. Функция next($i$) в строке 04 возвращает следующий после $i$ номер перехода из $P$.

*Пример*

Пусть сеть состоит из 6 переходов, определим $P$={3,4}, количество входов - $K3$=2, $K4$=2. Тогда получим $2^{K3}\cdot 2^{K4}$=16 комбинаций:

5. Решение однородной системы линейных уравнений $A'x$=0

Матрица $A'$ состоит из строк матрицы $A$, соответствующих нулевым значениям в векторе $\Delta\mu$ Считаем, что строки матрицы линейно независимы. Часть значений переменных определена вышеописанным перебором, таким образом, получаем систему из m уравнений и m неизвестных. Решение этой системы состоит из следующих шагов:

Шаг 0. Определим вектор-столбец свободных членов $b$: $b_i$=0.

Шаг 1. Для всех значений $x_j$, определенных перебором, переносим значения $a_{ij}x_j$ в правую часть уравнений: $b_i$:=$b_i$-$a_{ij}$*$x_j$, $i$=1..$m$.

Таким образом, получили неоднородную систему $A'x'$=$b$, где $x'$ – вектор неизвестных переменных. Заметим, что так как матрица $A'$ целочисленна, вектор $b$ также целочисленный.

**Таблица 1.** Генерация комбинаций переходов

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $t_1$ | $x_1$ | $x_1$ | $x_1$ | $x_1$ | $x_1$ | $x_1$ | $x_1$ | $x_1$ | $x_1$ | $x_1$ | $x_1$ | $x_1$ | $x_1$ | $x_1$ | $x_1$ | $x_1$ |
| $t_2$ | $x_2$ | $x_2$ | $x_2$ | $x_2$ | $x_2$ | $x_2$ | $x_2$ | $x_2$ | $x_2$ | $x_2$ | $x_2$ | $x_2$ | $x_2$ | $x_2$ | $x_2$ | $x_2$ |
| $t_{31}$ | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $t_{32}$ | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $t_{33}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| $t_{34}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| $t_{41}$ | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| $t_{42}$ | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| $t_{43}$ | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| $t_{44}$ | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| $t_5$ | $x_5$ | $x_5$ | $x_5$ | $x_5$ | $x_5$ | $x_5$ | $x_5$ | $x_5$ | $x_5$ | $x_5$ | $x_5$ | $x_5$ | $x_5$ | $x_5$ | $x_5$ | $x_5$ |
| $t_6$ | $x_6$ | $x_6$ | $x_6$ | $x_6$ | $x_6$ | $x_6$ | $x_6$ | $x_6$ | $x_6$ | $x_6$ | $x_6$ | $x_6$ | $x_6$ | $x_6$ | $x_6$ | $x_6$ |

Шаг 2. Решаем полученную систему $A'x'=b$ методом Гаусса с учетом целочисленности системы или любым другим методом решения целочисленных систем (алгоритм Диксона [7], алгоритм Сергеева [8]).

Шаг 3. Объединить найденный вектор $x'$ со значениями, полученными перебором, в вектор $x$.

6. Умножив вектор $x$ на исходную матрицу $A$, получим $\Delta\mu=Ax$.

*Пример*

$$A=\begin{pmatrix} -1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\;-1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 1\,0\,0\,0-1-1\,0\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0\,0-1\,0-1\,0\,0 \\ 0\,0\,1\,1\,1\,0\,0\,0\,0\,0-1\,0 \\ 0\,0\,0\,0\,0\,0\,1\,1\,1\,0\,0-1 \\ 0\,0\,0\,0\,0\,0\,0\,0-1-1\,1\,0 \\ 0\,0\,0-1\,0-1\,0\,0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1 \end{pmatrix},\quad \Delta\mu=\begin{pmatrix} \Delta\mu_1 \\ \Delta\mu_2 \\ 0 \\ 0 \\ 0 \\ 0 \\ \Delta\mu_6 \\ \Delta\mu_7 \\ \Delta\mu_8 \\ \Delta\mu_9 \\ \Delta\mu_{10} \end{pmatrix}$$

Тогда для 16 комбинаций из табл. 2 получим следующие решения (табл. 2).

Математическое моделирование цифровых автоматов проводимое авторами показало, что с увеличением сложности схем автоматов распет и количество линейных уравнений, описывающих функционирование растет по экспоненциальному закону, что приводит к увеличению времени решения уравнений. Для упрощения моделирования можно использовать два пути. Первый – декомпозиция схем цифровых автоматов [1], второй – композиция входов схем, составляющих цифровой автомат. Результаты исследований могут найти применение в моделировании и анализе систем управления, состоящих их цифровых автоматов.

*Работа выполнена в рамках государственного задания Министерства образования и науки Российской Федерации.*

**Таблица 2.** Решение для примера

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| $x_2$ | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| $x_5$ | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| $x_6$ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| $\mu_1$ | 0 | 0 | -1 | -1 | 0 | 0 | -1 | -1 | 0 | 0 | -1 | -1 | 0 | 0 | -1 | -1 |
| $\mu_2$ | 0 | 0 | 0 | 0 | -1 | -1 | -1 | -1 | 0 | 0 | 0 | 0 | -1 | -1 | -1 | -1 |
| $\mu_6$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $\mu_7$ | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | -1 |
| $\mu_8$ | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | -1 | 0 | -1 |
| $\mu_9$ | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| $\mu_{10}$ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

## СПИСОК ЛИТЕРАТУРЫ


1. *Зайцев Д.А.*Решение фундаментального уравнения сетей Петри в процессе композиции функциональных подсетей // Искусственный интеллект. 2005. №1. С.59-68.
2. *Питерсон Дж.* Теория сетей Петри и моделирование систем: Пер с англ. М.: Мир, 1984. 264 с.
3. *Мурата Т.* Сети Петри. Свойства, анализ, приложения // ТИИЭР. 1989. Т.77. №4. 1989. С.41-85.
4. *Хан А.А., Хура Г.С., Сингх Х., Нанда Н.К.* О нахождении решения уравнения состояний сетей Петри из класса уравнений Мурата // ТИИЭР.1989. №44. С.36-41.
5. *Кожевников В.В., Смагин А.А.* Процедуры анализа достижимости устойчивых состояний цифровых автоматов // Ученые записки УлГУ. 2012. №1 (4). С. 175-190.
6. *Кожевников В.В.* Концепция математического моделирования микропрограммируемых устройств. // Известия РАН. Техническая кибернетика. 1992. №4. С.175-179.
7. *Dixon J. D.* Exact solution of linear equations using P-adic expansions // Numer. Math.. 40. 1982. P. 137-141.
8. *Сергеев М.Б.* Гибридный разрядный метод решения систем уравнений в целочисленной арифметике. // Информационно-управляющие системы. 2003. №2-3. С. 16-18.


## THE SOLUTION OF MANAGEMENT SYSTEMS STATE EQUATION

© 2013 A.A. Smagin, V.V. Kozhevnikov, I.V. Kruglova

Ulyanovsk State University

The article is devoted to solving systems of linear algebraic equations that describe the behavior of the digital machines. We consider in detail the special case where the specificity describe the functioning of the digital machine. An example of the state equation solving of digital machine.
Keywords: fundamental equation Murata, digital machines, Petri nets, modeling, solving the system of equations.

*Alexey Smagin, Doctor of Technics, Professor, Head at the Telecommunications Technologies and Networks Department. E-mail: smaginaa1@mail.ru*
*Valery Kozhevnikov, Candidate of Technics, Associate Professor at the Telecommunications Technologies and Networks Department. E-mail: vvk2861955@mail.ru*
*Irina Kruglova, Graduate Student at the Applied Mathematics Department. E-mail: pm@ulsu.ru*