

Аннотация

Д.А. Зайцев Построение универсальной сети Петри // Материалы VI отчетной научно-практической конференции профессорско-преподавательского состава и студенчества, Международный гуманитарный университет, кафедра компьютерной инженерии, Одесса, 14-16 мая, 2010, с. 27-32.

Построена универсальная ингибиторная сеть Петри, которая исполняет произвольную заданную ингибиторную сеть Петри. Граф ингибиторной сети Петри, ее маркировка и последовательность срабатывания переходов зашифрованы как 10 неотрицательных целых скалярных переменных, представленные соответствующими позициями универсальной сети. По уравнению состояний построен алгоритм исполнения ингибиторной сети, который использует только указанные скалярные переменные; алгоритм закодирован ингибиторной сетью Петри. Используются подсети, которые реализуют арифметические и логические операции, копирование значений переменных.

Ключевые слова: *универсальная ингибиторная сеть Петри, универсальная машина Тьюринга, алгоритм, шифрование, поток управления*

Д.А. Зайцев Побудова універсальної сіті Петрі // Матеріали VI звітної науково-практичної конференції професорсько-викладацького складу та студентства, Міжнародний гуманітарний університет, кафедра комп'ютерної інженерії, Одеса, 14-16 травня, 2010, с. 27-32.

Побудовано універсальну інгібіторну сіть Петрі, яка виконує довільну задану інгібіторну сіть Петрі. Граф інгібіторної сіті Петрі, її маркірування і послідовність спрацьовування переходів зашифровано як 10 невід'ємних цілих скалярних змінних поданих відповідними позиціями універсальної сіті. За рівнянням станів побудовано алгоритм виконання інгібіторної сіті, який використовує лише вказані скалярні змінні; алгоритм закодовано інгібіторною сіттю Петрі. Використано підсіті, які реалізують арифметичні і логічні операції, копіювання значень змінних.

Ключеві слова: *універсальна інгібіторная сіть Петрі, універсальна машина Тюрінга, алгоритм, шифрування, потік управління*

D.A. Zaitsev Constructing Universal Petri Net // Proceedings of VI report scientific-practical conference of the faculty and students, International Humanitarian University, Chair of Computer Engineering, Odessa (Ukraine), May 14-16, 2010, p. 27-32.

The universal inhibitor Petri net was constructed that executes an arbitrary given inhibitor Petri net. The inhibitor Petri net graph, its marking and the transitions firing sequence were encoded as 10 scalar nonnegative integer numbers and represented by corresponding places of universal net. The algorithm of inhibitor net executing that uses scalar variables only was constructed on its state equation and encoded by universal inhibitor Petri net. Subnets which implement arithmetic, comparison and copying operations were employed.

Keywords: *universal inhibitor Petri net, universal Turing machine, algorithm, encoding, control flow*

Зайцев Д.А.

Построение универсальной сети Петри

Известны примеры построения универсальной машины Тьюринга [1]. Показано, что ингибиторная сеть Петри являются универсальной алгоритмической системой [2,3]. Целью настоящей работы является построение универсальной ингибиторной сети Петри, которая выполняет произвольную заданную ингибиторную сеть Петри.

Общая организация работы следующая: процесс функционирования ингибиторной сети описан уравнением состояний; введены правила шифрования (кодирования) ингибиторной сети Петри для ее представления фиксированным числом скалярных переменных (позиций универсальной сети); по уравнению состояний построен алгоритм выполнения ингибиторной сети с использованием полученного шифра ингибиторной сети; алгоритм закодирован ингибиторной сетью UIPN, которая и является универсальной.

Ингибиторная сеть Петри задана парой чисел, парой матриц и вектором: $N=(m, n, B, D, Q_0)$, где m, n – количество позиций и переходов, B, D – матрицы кратности входящих и исходящих дуг переходов соответственно, Q_0 – начальная маркировка. Уравнение состояний (1) описывает динамику сети:

$$\left\{ \begin{array}{l} q_j^k = q_j^{k-1} - x(b_{l,j}) + d_{l,j}, j = \overline{1, m} \\ u(t_i) = \bigwedge_{j=1, m} ((y(b_{l,j}) \wedge (q_j^{k-1} = 0)) \vee (\bar{y}(b_{l,j}) \wedge (q_j^{k-1} \geq b_{l,j}))), i = \overline{1, n} \\ u(t_l) = 1, l \in \overline{1, n} \\ k = 1, 2, \dots \\ x(b) = \begin{cases} b, b \geq 0 \\ 0, b = -1 \end{cases}, y(b) = \begin{cases} 0, b \geq 0 \\ 1, b = -1 \end{cases}. \end{array} \right. \quad (1)$$

Шифр вектора является формой представления чисел в позиционной системе счисления с основанием r и задан рекуррентным выражением (2); для дешифрования использовано (3). При шифровании матриц использовано векторное представление с разложением по строкам.

$$s_j = s_{j-1} \cdot r + q_{m-1-j}, s_0 = q_{m-1}, j = \overline{1, m-1}, r = \max_j q_j + 1. \quad (2)$$

$$\begin{aligned} q_j &= s_{m-1-j} \bmod r, s_{m-1-(j+1)} = s_{m-1-j} \operatorname{div} r, s_{m-1} = s, \\ j &= \overline{0, m-1}. \end{aligned} \quad (3)$$

Шифр сети Петри представлен 10 позициями UIPN с именами $m, n, sB, rB, sD, rD, sQ, rQ, k, sZ$, где префикс s обозначает шифр (2), префикс r – использованное основание, символ Z обозначает последовательность срабатывания переходов, k – длина последовательности. Для шифрования вектора \bar{u} применены правила шифрования вектора (2) при $r = 2$.

По системе (1) в соответствии с выбранным способом шифрования построен алгоритм AUIPN выполнения ингибиторной сети Петри на Си-подобном псевдоязыке; обозначения: u – шифр индикатора возбужденных переходов, l – номер срабатывающего перехода, k – номер шага; CheckFire – проверка условий возбуждения переходов, PickFire – выбор срабатывающего перехода, Fire – срабатывание перехода; NonDeterministic –

недетерминированный выбор числа из множества $\{0,1\}$; процедуры MUL_ADD, MOD_DIV, реализуют рекуррентное шифрование/дешифрование в соответствии с (2), (3).

```

void AUIPN()
{
    uint u, l;
    inputXIPN();
    k=1; sZ=0;
    while(NonDeterministic())
    {
        CheckFire(&u);
        if(u==0) goto out;
        PickFire(u, &l);
        Fire(l);
        MUL_ADD(&sZ, n, l-1);
        k++;
    }
    out: outputXIPN();
}
void MUL_ADD(&x, y, z)
{
    (*x) = (*x) * y + z;
}
void MOD_DIV(&m, &x, y)
{
    (*m) = (*x) mod y;
    (*x) = (*x) div y;
}
void CheckFire(uint *u)
{
    uint i, j, qj, bij, ui, uij;
    uint sB1, sQ1;
    sB1=sB; &u=0;
    for(i=n; i>0; i--)
    {
        sQ1=sQ;
        ui=1;
        for(j=m; j>0; j--)
        {
            MOD_DIV(&qj, &sQ1, rQ);
            MOD_DIV(&bij, &sB1, rB);
            uij=1;
            if(bij==0) continue;
            bij--;
            if(bij==0) uij=(qj==0);
            else uij=(qj>=bij);
            ui=ui && uij;
        }
        MUL_ADD(&u, 2, ui);
    }
}

void PickFire(uint u, uint *l)
{
    uint ui, i;
    i=0;
    while(u>0)
    {
        MOD_DIV(&ui, &u, 2);
        i++;
        if(ui==0) continue;
        if(NonDeterministic()) goto out;
    }
    out: *l=i;
}

void Fire(uint l)
{
    uint rQ1, maxQ1, shift, qj, bij, dij, j;
    uint sB1, sD1, sQ1;

    sB1=sB; sD1=sD;
    sQ1=0; rQ1=rQ+rD-1; maxQ1=0;

    shift=(n-1)*m;
    while(shift-->0)
    {
        MOD_DIV(&b, &sB1, rB);
        MOD_DIV(&d, &sD1, rD);
    }

    for(j=m; j>0; j--)
    {
        MOD_DIV(&qj, &sQ, rQ);
        MOD_DIV(&bij, &sB1, rB);
        if(bij>0) bij--;
        MOD_DIV(&dij, &sD1, rD);
        qj=qj-bij+dij;
        maxQ1=MAX(qj, maxQ1);
        MUL_ADD(&sQ1, rQ1, qj);
    }
    sQ=0; rQ=maxQ1+1;

    for(j=m; j>0; j--)
    {
        MOD_DIV(&qj, &sQ1, rQ1);
        MUL_ADD(&sQ, rQ, qj);
    }
}

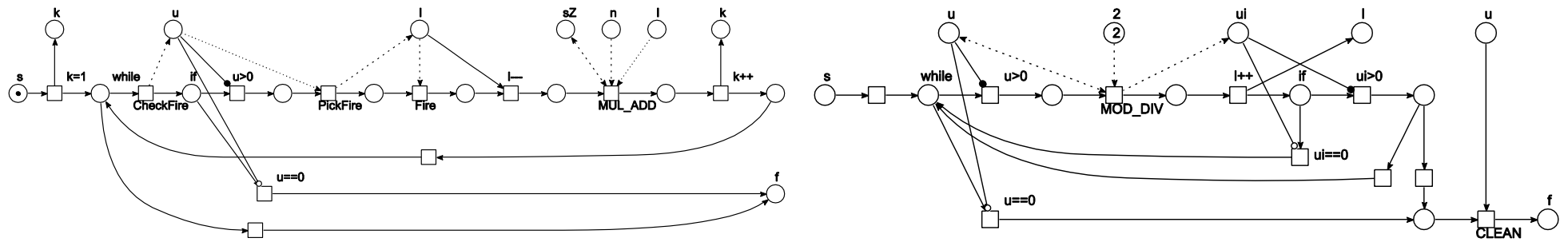
```

Теорема 1. Алгоритм AUIPN реализует динамику произвольной заданной ингибиторной сети Петри.

Теорема 2. Алгоритм AUIPN может быть представлен ингибиторной сетью Петри.

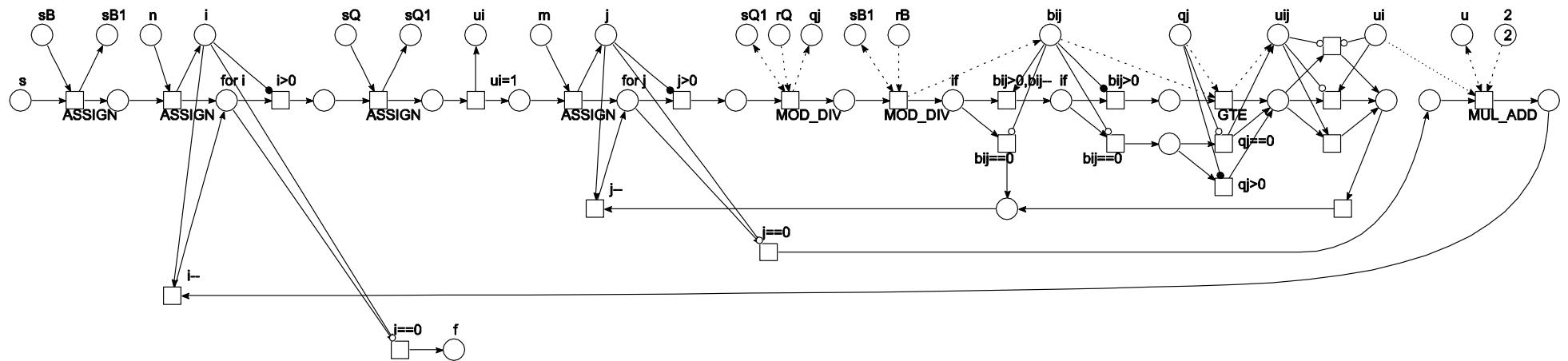
Алгоритм AUIPN закодирован ингибиторной сетью Петри UIPN (рис. 1).

Теорема 3. Сеть UIPN является универсальной ингибиторной сетью Петри.



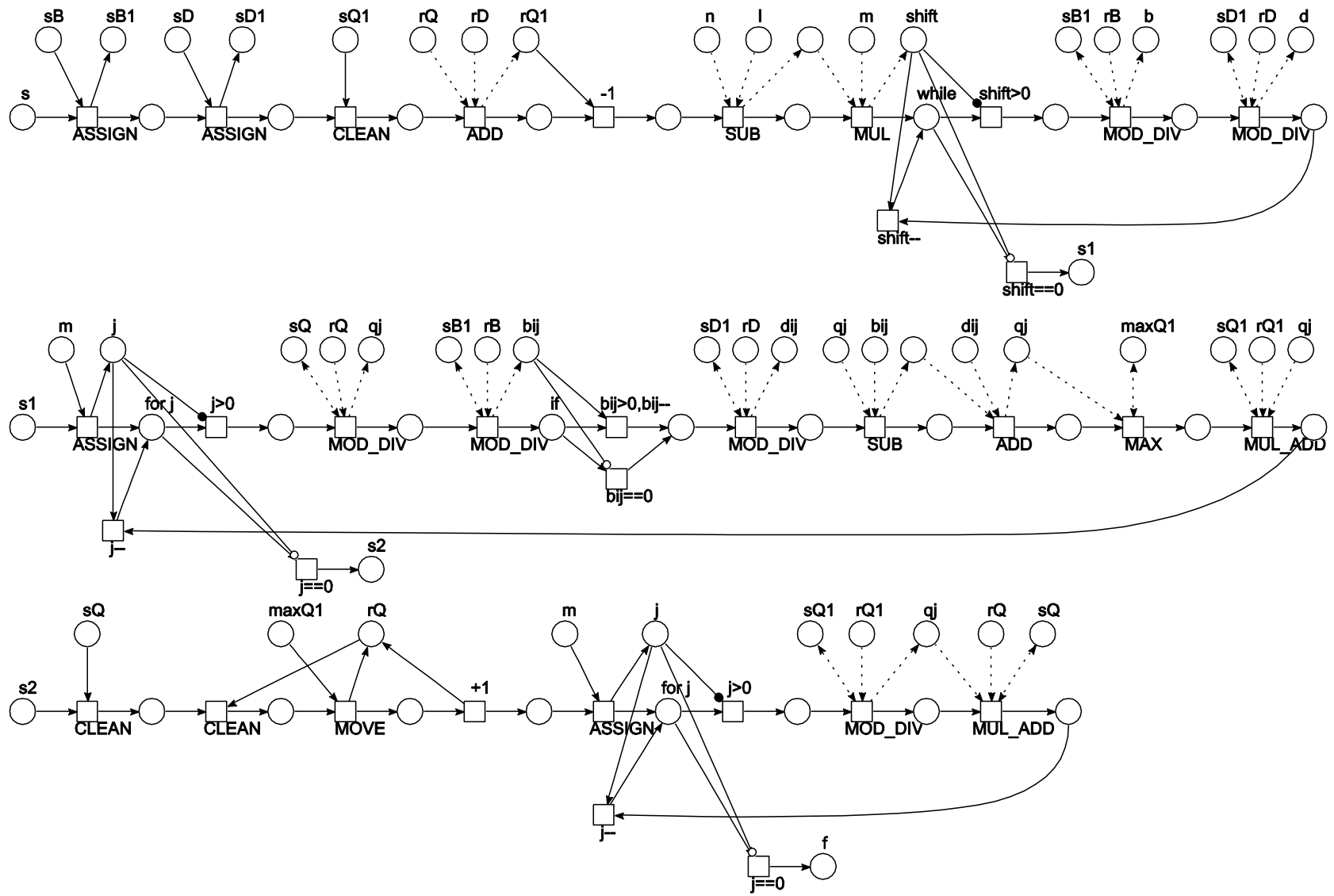
a) UIPN

b) PickFire



c) CheckFire

Рис. 1. Универсальная ингибиторная сеть Петри UIPN (начало)



d) Fire

Рис. 1. Универсальная ингибиторная сеть Петри UIPN (завершение)

Использованы вспомогательные сети, реализующие арифметические, логические операции и правила копирования значений переменных в позиции универсальной сети: CLEAN – очистка, COPY – копирование, MOVE – перемещение, ASSIGN – присваивание, ADD – сложение, SUB – вычитание, MUL – умножение, GTE – сравнение, MAX – максимум, MUL_ADD – добавление к шифру, MOD_DIV – извлечение из шифра, а также правила кодирования последовательных, ветвящихся, циклических алгоритмов.

Таким образом, построена универсальная ингибиторная сеть Пери; направлением дальнейших исследований является построение универсальной сети с минимальным количеством позиций (переходов), минимальным значением маркировки.

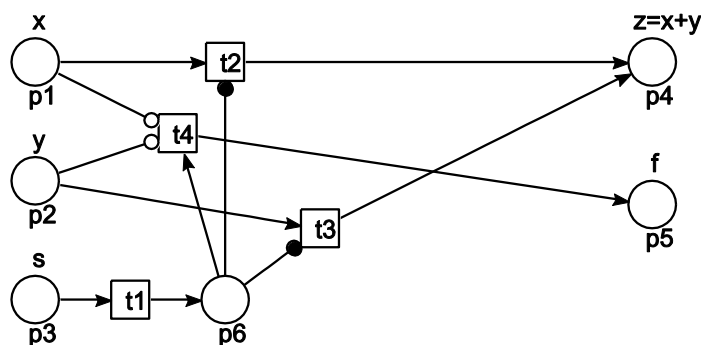
ЛИТЕРАТУРА

1. The Universal Turing Machine. A Half-Century Survey / Rolf Herken (ed.), Springer-Verlag, Wien New York, 1994, 609 p.
2. Agerwala T. A Complete Model for Representing the Coordination of Asynchronous Processes, Baltimore, John Hopkins University, Hopkins Computer Science Program, Res. Rep. No. 32, July 1974.
3. Котов В.Е. Сети Петри. – М.: Наука, 1984. – 160 с.

Приложение: Примеры вспомогательных сетей

1) Сложение ($z = x + y$)

ADD::



2) Добавить к шифру ($z = x \cdot y + v$)

MUL_ADD::

