

Я. ТЕККО

## ПРИМЕНЕНИЕ ЯЗЫКА МОДЕЛИ КЕРКА ДЛЯ СРАВНЕНИЯ МОДЕЛИ КЕРКА С СЕТЬЮ ПЕТРИ

(Представил Ю. Яаксоо)

Одним из наиболее распространенных формализмов, применяемых для описания и анализа параллельно и асинхронно выполняемых программ, является сеть Петри, основные свойства которой изложены в [1, 2].

В связи с исследованием программного обеспечения встроенных систем [3] возникли некоторые принципиальные трудности, в частности, при анализе с помощью сетей Петри существенно асинхронного режима работы программ. Утверждается [3], что удобным средством анализа встроенных систем является модель Керка [4, 5].

Учитывая большой объем работы, проделанной при исследовании свойств сетей Петри и при разработке алгоритмов анализа сетей Петри, целесообразно добиться строгого сравнения свойств модели Керка с сетями Петри. В принципе не исключена возможность обоюдного заимствования алгоритмов анализа обеих моделей.

В [6] делается попытка сравнить мощность моделирования сетей Петри и модели Керка, используя методику, предложенную Дж. Питерсоном в [2]. В данной работе сравниваются свойства модели Керка и сетей Петри при помощи языков, генерируемых обоими формализмами. Используя преобразование из класса моделей Керка в класс сетей Петри конструируется для каждой модели Керка сеть Петри, так что обе модели функционируют в определенном смысле одинаково. Хотя предложенное преобразование пренебрегает некоторыми свойствами модели Керка — неадекватно преобразуются функции асинхронных каналов и почти все временные ограничения — многие структурные свойства сохраняются.

Вводится язык модели Керка по аналогии с языком сети Петри [1]. Доказывается, что язык модели Керка  $\Phi$  эквивалентен языку сети Петри, полученному применением предложенного в данной статье преобразования  $\Phi$  к модели Керка. Доказывается также, что модель Керка с информационным тупиком преобразуется в неживую сеть Петри.

### 1. Сети Петри

Определение 1. Сеть Петри  $\Omega$  [1] описывается совокупностью  $\Omega = (E, \Delta, \Lambda, M^0)$ , где

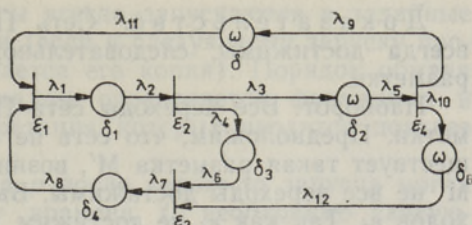
$E$  — список переходов:  $E = \{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n\}$ ;

$\Delta$  — список мест:  $\Delta = \{\delta_1, \delta_2, \dots, \delta_m\}$ ;

$\Lambda$  — список дуг:  $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_h\}$ , где  $\lambda_i = (x_{i1}, x_{i2})$ , здесь  $x_{i1} \in \Delta$  и  $x_{i2} \in E$ , или  $x_{i1} \in E$  и  $x_{i2} \in \Delta$ ;

$M^0$  — начальная разметка:  $M^0 = \{\mu_1^0, \mu_2^0, \dots, \mu_m^0\}$ , где  $\mu_i^0 \in \mathcal{N} \cup \{\omega\}$ ,  $\mu_i^0$  — начальная разметка места  $\delta_i$  ( $\omega$  — первый счетный ординал,  $\mathcal{N}$  — множество натуральных чисел).

Рис. 1.



Графически (см. рис. 1) в сети Петри места обозначаются кружками, переходы — барьерами, дуги — стрелками и разметка — точками (нужное количество точек внутри соответствующего кружка).

### 1.1. Язык сети Петри

Приведем теперь несколько понятий, которые нужны для описания языка сети Петри, опираясь на [1].

Функционирование сети Петри является результатом срабатывания переходов.

**Определение 2.** Переход  $\epsilon$  может сработать при разметке  $M$ , если для каждого  $\delta \in \Delta$  соблюдается условие: если существует такое  $\lambda_i \in \Lambda$ , что  $\lambda_i = (\delta, \epsilon)$ , то  $M(\delta) > 0$ .

Функция  $M(\delta_i)$  дает разметку места  $\delta_i$ , перед работой сети  $\forall_j (M(\delta_j) = \mu_j^0(\delta_j))$ .

В результате работы перехода  $\epsilon$  разметка  $M$  заменяется новой разметкой  $M'$  согласно следующему правилу:

для каждого  $\delta \in \Delta$ , если существует  $\lambda_i \in \Lambda$ , где  $\lambda_i = (\delta, \epsilon)$ , то  $M'(\delta) = M(\delta) - 1$ ;

для каждого  $\delta \in \Delta$ , если существует  $\lambda_i \in \Lambda$ , где  $\lambda_i = (\epsilon, \delta)$ , то  $M'(\delta) = M(\delta) + 1$ ;

для каждого  $\delta \in \Delta$ , если не существует  $\lambda_i \in \Lambda$ , где  $\lambda_i = (\delta, \epsilon)$  или  $\lambda_i = (\epsilon, \delta)$ , то  $M'(\delta) = M(\delta)$ .

После срабатывания перехода вычисляется новая разметка  $M'$  и произвольно выбирается переход среди работоспособных на  $M'$  переходов, который срабатывает в следующую очередь.

Как результат функционирования сети появляется последовательность переходов, которую можно рассматривать как слово в алфавите, состоящем из всех переходов  $E = \{\epsilon_1, \dots, \epsilon_n\}$  сети Петри.

**Определение 3.** Языком  $L(\Omega)$  [1] сети Петри называется множество всевозможных слов, возникающих в результате функционирования данной сети.

### 1.2. Свойство одного подкласса сетей Петри

**Определение 4** [1]. Переход  $\epsilon$  называют достижимым в сети Петри  $\Omega$ , если исходя от начальной разметки существует некоторая последовательность срабатывания переходов  $\epsilon_{i_1}, \dots, \epsilon_{i_n}$ , такая что в результате срабатывания этих переходов может срабатываться переход  $\epsilon$ .

**Определение 5** [1]. Сеть Петри живая, если все ее переходы достижимы от всех разметок, которые могут возникать при функционировании сети.

**Предложение 1.** Если сеть Петри  $\Omega$  синхрограф, то есть для каждого места  $\delta$  имеется точно одна входящая и точно одна выходящая дуга  $\lambda$ , то  $\Omega$  жива тогда и только тогда, если все ее переходы достижимы от начальной разметки.

**Доказательство.** Сеть Петри живая, значит, все переходы всегда достижимы, следовательно, они достижимы и от начальной разметки.

Наоборот. Все переходы сети Петри достижимы от начальной разметки. Предположим, что сеть не является живой. Следовательно, существует такая разметка  $M'$ , возникающая в ходе работы сети, что из  $M'$  не все переходы достижимы. Выберем один из недостижимых переходов  $\varepsilon_i$ . Так как  $\varepsilon_i$  не достижим, то существует место  $\delta_i$ , откуда в  $\varepsilon_i$  выйдет дуга, разметка которого всегда остается нулевой. Следовательно, переход  $\varepsilon_j$ , из которого в  $\delta_i$  выйдет дуга, тоже не достижим, если  $\varepsilon_i = \varepsilon_j$ , тогда  $\varepsilon_i$  не достижим от  $M^0$ , что противоречит предположению. Так как  $\varepsilon_j$  не может срабатывать, то существует  $\delta_j$ , а следовательно, и  $\varepsilon_k$ , который не достижим (если  $k=i$  или  $k=j$ , то  $\varepsilon_i$  или  $\varepsilon_j$  не достижим от  $M^0$ ). Продолжая подобным образом, после  $n$  шагов дойдем до противоречия ( $n$  — количество переходов в сети  $\Omega$ ).

## 2. Модель Керка

Теперь рассмотрим некоторые понятия и свойства модели Керка.

**Определение 6.** Модель Керка  $Q$  [5] описывается совокупностью  $Q = (P, C, Cf, Ct, Ph)$ , где

$P$  — список процессов:  $P = \{p_1, p_2, \dots, p_n\}$ ;

$C$  — список каналов:  $C = \{c_1, c_2, \dots, c_m\}$ , где  $c_i = (p_i, p_{i_2})$  и  $p_i, p_{i_2}$  процессы из списка  $P$ ;

$Cf$  — список функции каналов:  $Cf = \{f_1, f_2, \dots, f_m\}$ , где  $f_i = (a_i, b_i)$  функция канала  $c_i$  и  $a_i, b_i \in \mathcal{N}$ ;

$Ct$  — список типов каналов:  $Ct = \{t_1, t_2, \dots, t_m\}$ , где  $t_i \in Z$  тип канала  $c_i$  и  $Z$  множество возможных типов канала  $Z = \{s, a, p, n\}$ ;

$Ph$  — список множеств пусковых моментов процессов:  $Ph = \{H_1, H_2, \dots, H_n\}$ , где  $H_i = \{t_i, \dots\}$  множество пусковых моментов процесса  $p_i$ ;

**З а м е ч а н и е.** Обычно в определении модели Керка даны допустимые максимальные и минимальные времена работы процессов  $\tau_i$  и  $\tau_i'$ ,  $i=1, \dots, n$ . В данной работе предполагается, что максимальное время работы  $\tau_i$  является неограниченным при всех  $i=1, \dots, n$ . Такое дополнительное предположение существенно упрощает структуру получаемой сети Петри, но не влияет на исследуемые в данной работе свойства модели Керка.

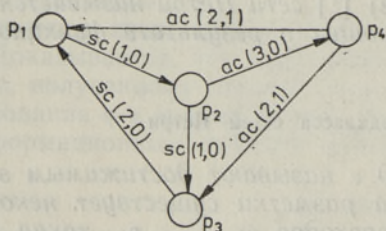


Рис. 2.

При графическом изображении модели (см. рис. 2) процессы обозначаются кружками, а каналы — стрелками вместе с пометкой о соответствующих параметрах, о типе и функции.

### 2.1. Язык модели Керка

В результате функционирования модели Керка возникает последовательность сработанных процессов модели. Функционирование модели Керка зависит от порядка работы процессов и от связей для передачи

данных между процессами. Процессы всегда запускаются в заданные в описании модели моменты времени (если к следующему запуску процесс еще не завершился, то запускается его копия). Порядок обмена данных между процессами по каналам определяется функциями и типами каналов, частичная интерпретация которых вводится позднее (подробнее см. [4]).

Для того, чтобы работа процесса  $p$  после какого-то запуска могла нормально завершиться к моменту времени  $t$ , необходимо следующее:

- а) рассматриваемый запуск должен состояться раньше момента времени  $t$ ;
- б) для рассматриваемого запуска должны существовать исходные данные.

Если множество процессов  $P$  модели Керка  $Q$  рассматривать как алфавит, то язык модели Керка определяется следующим образом.

**Определение 7.** Язык  $L(Q)$  модели Керка состоит из слов, каждое из которых совпадает с возможной последовательностью завершения работы процессов модели Керка.

## 2.2. Свойства слов языка модели Керка

Введем следующие обозначения:

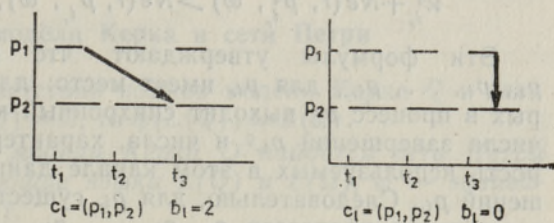
$w_i$  — слово, которое имеет форму  $p_{i_1} p_{i_2} \dots p_{i_r}$ , где  $r$  количество букв в слове;

$p_{i_j}^k, k=1, 2, \dots, u_j$  — процессы, производящие начальные данные для процесса  $p_{i_j}$ ;

$Ne(n, y, w_i)$  — функция, значение которой равно числу появления буквы  $y$  в слове  $w_i$  среди первых  $n-1$  букв;

$\kappa_{i_j}^k$  — функциональный параметр, значение которого равно числу  $\omega$ , если канал  $c_l = (p_{i_j}^k, p_{i_j})$  асинхронный ( $a$ ), если же  $c_l$  синхронный ( $s$ ) или типа Петри ( $p$ ), то значение параметра  $\kappa_{i_j}^k$  равно значению параметра  $b_l$  функции  $f_l$  канала  $c_l$ , который указывает место получения исходных данных (см. рис. 3).

Рис. 3.



**Лемма.** Слово  $w_i$  является словом языка модели Керка тогда и только тогда, когда для всех  $j=1, \dots, r$  ( $r=|w_i|$ ) соблюдается неравенство

$$\kappa_{i_j}^k + Ne(j, p_{i_j}^k, w_i) > Ne(j, p_{i_j}, w_i) \quad (1)$$

при всех  $k=1, 2, \dots, u_j$ .

**Доказательство.** Доказательство леммы состоит из двух частей: в первой части покажем, что каждое слово языка модели Керка удовлетворяет неравенству (1), во второй части покажем при помощи индукции, что каждое слово, удовлетворяющее неравенству (1), принадлежит языку модели Керка.

Предположим, что слово  $w_i$ , принадлежащее языку модели Керка, не удовлетворяет неравенству (1) при всех  $j \in [1, r]$ . Следовательно, существуют  $a \in [1, r]$  и  $q \in [1, u_j]$ , для которых

$$\kappa_{i_a}^q + Ne(a, p_{i_a}^q, w_i) \leq Ne(a, p_{i_a}, w_i). \quad (2)$$

Пусть  $Ne(a, p_{i_a}^q, w_i) = A$  и  $Ne(a, p_{i_a}, w_i) = B$ , тогда имеем

$$\kappa_{i_a}^q + A \leq B, \quad (3)$$

т. е. к моменту завершения  $B+1$ -го запуска процесса  $p_{i_a}$ , завершено  $A$  запусков процесса  $p_{i_a}^q$  и процессы  $p_{i_a}$  и  $p_{i_a}^q$  связаны синхронным каналом или каналом Петри (иначе  $\kappa_{i_a}^q = \omega$ ), причем запоздание исходных данных от процесса  $p_{i_a}^q$  равно  $\kappa_{i_a}^q$ .

Из свойств синхронного канала и канала Петри [4, 5], можно заключить, что для существования начальных данных для  $B+1$ -го запуска процесса  $p_{i_a}$  все процессы производители, связанные с  $p_{i_a}$  синхронным каналом или каналом Петри, удовлетворяют условию  $\kappa_{i_a}^q + A \geq B+1$ .

Следовательно,  $\kappa_{i_a}^q + A > B$ .

Получили противоречие с условием (3).

Пусть  $w_i$  удовлетворяет неравенству (1). Покажем, что  $w_i$  принадлежит языку модели Керка. Применим индукцию.

Базис индукции. Пусть слово  $w_i$  состоит из одной буквы. В таком случае  $A=B=0$ . Из (1) получаем  $\kappa_{i_j}^k > 0$ ,  $k=1, 2, \dots, u_j$ . Следовательно, к началу работы системы для процесса  $p_{i_j}$  существуют все начальные данные. Это значит, что процесс  $p_{i_j}$  в принципе может закончиться первым (в моменты запуска каждого процесса запускается нужное количество копий этого процесса, если этот процесс или его копии еще работают). Следовательно, слово  $w_i$  принадлежит языку модели Керка.

Шаг индукции. Пусть слова, удовлетворяющие (1) и имеющие длину  $r=n$ , принадлежат языку модели Керка. Пусть для  $w$  с длиной  $n+1$  имеют место (1).

Выпишем (1) для последней буквы  $w$  ( $r=n+1$ )

$$\kappa_{i_r}^q + Ne(r, p_{i_r}^q, w) > Ne(r, p_{i_r}, w), \quad \text{где } q=1, 2, \dots, u_r.$$

Эти формулы утверждают, что после завершения процессов  $p_{i_1}, p_{i_2}, \dots, p_{i_{r-1}}$  для  $p_{i_r}$  имеет место: для всех процессов  $p_{i_r}^q$ , из которых в процесс  $p_{i_r}$  выходит синхронный канал или канал Петри, сумма числа завершений  $p_{i_r}^q$  и числа, характеризующего относительную старость используемых в этом канале данных, больше, чем число завершений  $p_{i_r}$ . Следовательно, для  $p_{i_r}$  существуют начальные данные после завершения процессов  $p_{i_1}, p_{i_2}, \dots, p_{i_{r-1}}$  и  $p_{i_r}$  может закончиться следующим.

Так как подслово, которое образуется из  $n$  первых букв слова  $w$ , принадлежит по условию индукции языку модели Керка, то тому же языку принадлежит каждое слово длиной  $n+1$ , которое удовлетворяет (1). Лемма доказана.

### 3. Преобразование моделей

При помощи  $\Phi$  приставляется к модели Керка  $Q$  (см. опр. 6) сеть Петри  $\Omega$  (см. опр. 1) следующим способом:

а) каждый процесс  $p_i$  модели Керка преобразуется в переход сети Петри  $\varepsilon_i$ ;

- б) каждый ненулевой  $(a, s, p)$  канал  $c_i = (p_{i_1}, p_{i_2})$  модели Керка преобразуется в тройку, которую образуют место  $\delta_i$  сети Петри и дуги, соединяющие это место с переходами сети Петри, соответствующими процессам  $p_{i_1}$  и  $p_{i_2}$  модели Керка (направление дуг совпадает с направлением канала);
- в) функции и типы каналов модели Керка преобразуются в начальную разметку сети Петри следующим образом: если тип канала асинхронный  $(a)$ , то разметка  $\mu_i^0$  места  $\delta_i$  сети Петри, которая соответствует этому каналу, равна  $\omega$ , если канал не асинхронный,  $(s, p)$ , то разметка  $\mu_i^0$  равна второму параметру  $b_i$  функции этого канала.

Более формально преобразование  $\varphi$  из класса моделей Керка в класс сетей Петри определяется следующим образом.

Определение 8. Преобразование  $\varphi$  является функцией, которая удовлетворяет следующим требованиям:

- а)  $\varphi(p_i) = \varepsilon_i$ ;
- б) Пусть  $c_i = (p_{i_1}, p_{i_2})$ , тогда если  $t_i(c_i) \neq n$ , то  $\varphi(c_i) = \lambda_{j_1} \times \delta_i \times \lambda_{j_2}$ , где  $\lambda_{j_1} = (\varepsilon_{i_1}, \delta_i)$ ,  $\lambda_{j_2} = (\delta_i, \varepsilon_{i_2})$ ,  $\varphi(p_{i_1}) = \varepsilon_{i_1}$  и  $\varphi(p_{i_2}) = \varepsilon_{i_2}$ , здесь  $i_1, i_2 \in [1, n]$  и  $j_1, j_2 \in [1, k]$ ;
- в)  $\varphi(f_i, t_i) = \mu_i^0$ , где  $f_i = (a_i, b_i)$  и

$$\mu_i^0 = \begin{cases} b_i, & \text{если } t_i = s \text{ или } t_i = p \\ \omega, & \text{если } t_i = a. \end{cases}$$

З а м е ч а н и е. Точнее, класс моделей Керка преобразуется в подкласс сетей Петри (в синхрографы), так как каждое место сети Петри, полученного при помощи  $\varphi$  (см. опр. 8 б) имеет точно одну входящую и выходящую дугу.

Применение  $\varphi$  для  $L(Q)$  определяется следующим образом:

$$\varphi(L(Q)) = \{w_i^* \mid w_i^* = \varepsilon_{i_1}, \varepsilon_{i_2}, \dots, \varepsilon_{i_j}, \dots \ \& \ \forall k (\varepsilon_{i_k} = \varphi(p_{i_k})) \ \& \ w_i = p_{i_1}, p_{i_2}, \dots, p_{i_j}, \dots \ \& \ w_i \in L(Q)\}.$$

Значит  $\varphi(L(Q))$  определяет язык на алфавите  $E = \{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n\}$ .

#### 4. Связь языков модели Керка и сети Петри

Докажем результат о соответствии языков модели Керка  $Q$  и сети Петри  $\Omega$ . Конкретнее, если  $\varphi(Q) = \Omega$ , то  $\varphi(L(Q)) = L(\Omega)$ .

Теорема. Для любой модели Керка  $Q$  найдется сеть Петри  $\Omega = \varphi(Q)$ , так что соответствующие языки  $L(Q)$  и  $L(\Omega)$ ,  $\varphi$  — эквивалентные, (т. е.  $\varphi(L(Q)) = L(\Omega)$ ).

Доказательство. Покажем, что если  $\Omega = \varphi(Q)$ , то  $L(\Omega) = \varphi(L(Q))$ . Пусть  $w = \varepsilon_{i_1}, \dots, \varepsilon_{i_k}$  и  $w \in L(\Omega)$ . Докажем, что для  $w^* = p_{i_1}, \dots, p_{i_k}$ ,  $\forall j (\varphi(p_{i_j}) = \varepsilon_{i_j})$ , верно  $w^* \in L(Q)$ . Так как  $w \in L(\Omega)$ , то для каждой позиции  $l$  слова  $w$  переход  $\varepsilon_{i_l}$  может срабатываться, т. е. для каждого  $\delta \in \Delta$ , если существует  $\lambda_{ij} = (\delta_j, \varepsilon_{i_l})$ , то  $M(\delta_j) > 0$ .

Следовательно, для перехода  $\varepsilon_{i_l}$  в позиции  $l$  и для любого  $\delta_l \in \Delta$  имеем следующее: если существует  $\lambda_{lj} = (\delta_j, \varepsilon_{i_l})$ , то

$$\mu(\delta_j) + A > B, \quad (3')$$

где  $A = Ne(l, \varepsilon_{j_x}, w)$  и  $B = Ne(l, \varepsilon_{i_l}, w)$ .

Здесь  $\varepsilon_{j_x}$  единственный переход, из которого направляется дуга в  $\delta_j$  (т. е.  $\forall \varepsilon (\exists \lambda (\lambda = (\varepsilon, \delta_j)) \Rightarrow \varepsilon = \varepsilon_{j_x})$ ).

Формула (3') верна, поскольку  $\varepsilon_{j_k}$  единственный переход, срабатывание которого генерирует фишки в место  $\delta_j$ .

Следовательно, если  $\omega \in L(\Omega)$ , то для каждой позиции  $k$  слова  $\omega$  имеем:

$$\forall \varepsilon_{j_k}^x \forall \delta_{j_k}^x [\exists \lambda_{1_k}^x \exists \lambda_{2_k}^x (\lambda_{1_k}^x = (\varepsilon_{j_k}^x, \delta_{j_k}^x) \& \lambda_{2_k}^x = (\delta_{j_k}^x, \varepsilon_{j_k}^x)) \Rightarrow \mu^0(\delta_{j_k}^x) + Ne(k, \varepsilon_{j_k}^x, \omega) > Ne(k, \varepsilon_{j_k}^x, \omega)] \quad (4)$$

Учитывая определение преобразования  $\varphi$ , получим из условия (4), что для всех процессов  $p_{j_k}^x$ , которые связаны с процессом  $p_{j_k}$  через канал (предположение импликации (4)), имеет место:

$$\mu_{j_k}^x + Ne(k, p_{j_k}^x, \omega^*) > Ne(k, p_{j_k}, \omega^*) \quad (5)$$

для каждой позиции  $k$  слова  $\omega^*$ .

По доказанной выше лемме  $\omega^* \in L(Q)$ , если соблюдено (5).

Доказательство обратного. Если  $\omega^*$  принадлежит языку  $L(Q)$ , то  $\omega^*$  удовлетворяет требованиям (5). Применяя  $\varphi$  получим, что соответствующая  $\omega$  удовлетворяет требованиям (4) и (3') и следовательно принадлежит языку сети Петри  $L(\Omega)$ . Теорема доказана.

## 5. Преобразование модели Керка с информационным тупиком

Выясним, как преобразуется информационный тупик модели Керка  $Q$  в соответствующую сеть Петри  $\Omega = \varphi(Q)$

**Определение 9.** Процессы  $p_1, p_2, \dots, p_q$  модели Керка находятся в информационном тупике, если они одновременно ждут друг от друга исходных данных и поэтому никогда не могут завершить работу. (В терминах модели Керка: для каждого  $p_i, i \in [1, q]$  существует  $c_{h_i} = (p_j, p_i)$ , так что  $j \in [1, q]$  и  $f_{h_i}(c_{h_i}) = (a_{h_i}, 0)$  и  $t(c_{h_i}) = s$  или  $t(c_{h_i}) = p$ ).

Условия существования тупика доказаны в [5].

Используем для анализа информационного тупика вышедоказанную теорему.

**Предложение 2.** Сеть Петри жива  $\Omega = \varphi(Q)$  тогда и только тогда, когда в модели Керка  $Q$  не существует тупика.

**Доказательство.** Сеть Петри  $\Omega$  жива, следовательно, из всех разметок все переходы достижимы, и следовательно, язык сети Петри  $\Omega$  имеет следующее свойство: для всех слов  $\omega$  найдется слово  $\omega'$ , так что  $\omega'$  можно рассматривать как  $\omega\omega''$  (первые буквы  $\omega'$  образуют слово  $\omega$  и остальные слово  $\omega''$ ), причем в  $\omega''$  содержатся все буквы алфавита (все переходы  $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$ ). Но тогда в модели Керка  $Q$  не существует последовательности  $\omega'''$  работы процессов, после которой некоторые процессы не могут закончить работу. Это значит, что в модели Керка  $Q$  нет тупика.

Доказательство обратного аналогичное.

**Предложение 3.** В сети Петри  $\Omega = \varphi(Q)$  каждый переход достижим от начальной разметки тогда и только тогда, когда в модели Керка  $Q$  не существует тупика.

**Доказательство.** Так как по предложению 1 достижимость каждого перехода и живость сети эквивалентны, то по предложению 2 утверждение правильно.

**Предложение 4.** Информационный тупик не может возникнуть в модели Керка  $Q$  в ходе работы.

Доказательство. Предположим, что тупик возник в модели Керка в ходе работы. Следовательно, перед его возникновением все процессы могли завершать работу и не существует процесса, который не включает в состав ни одного слова языка модели Керка  $Q$ . Учитывая, что  $\Omega = \varphi(Q)$ , получим, что все переходы сети Петри  $\Omega$  достижимы от начальной разметки. Применяя предложение 3, получим, что в модели Керка  $Q$  не имеется тупика. Противоречие.

### Заключение

Допущение о неограниченности максимального времени выполнения всех процессов модели Керка дает возможность разумно построить для каждой модели Керка некоторую сеть Петри.

Метод построения сети Петри (преобразование  $\varphi$ ) выбран так, что язык исходной модели Керка сохраняется, т. е. язык получаемой сети Петри совпадает с языком исходной модели Керка. Это значит, что возможные варианты функционирования обеих моделей совпадают. Кроме того, одни свойства модели Керка (наличие информационного тупика) превращаются в другие свойства сети Петри (неживость сети).

### ЛИТЕРАТУРА

1. Котов В. Е. Сети Петри. М., «Наука», 1984.
2. Питерсон Дж. Теория сетей Петри и моделирование систем. М., «Мир», 1984.
3. Мытус Л. Л. Техн. кибернетика, № 4, 149—155 (1985).
4. Mötus, L., Lomp, A. In: Preprints 9th IFAC World Congress, 2. Budapest, 1984, 144—148.
5. Mötus, L., Käärmees, K. In: Proc. 4th IFAC Workshop on DCCS (eds R. W. Gellie, R. R. Tavast). New York, Pergamon Press, 1983, 93—101.
6. Вайн Ю. Изв. АН ЭССР. Физ. Матем., 36, № 3, 324—333 (1987).

Институт кибернетики  
Академии наук Эстонской ССР

Поступила в редакцию  
19/1 1987

### J. TEKKO

#### PETRI VÖRGU JA QUIRKI MUDELI VÖRDLUS QUIRKI MUDELI KEELE ABIL

On võrreldud kahte paralleelsete süsteemide kirjeldamise formalismi — üldlevinud ja põhjalikult uuritud Petri võrke ning vähemtuntud Quirki mudelit. Petri võrkude ja Quirki mudeli võrdlus nende poolt genereeritud keelte alusel osutub formalismide põhimõttelisest erinevusest tingituna võimalikuks vaid Quirki mudeli lihtsustamisel. Nagu tõestatud, ei mõjuta aga tehtud lihtsustus formalismide uuritavaid omadusi.

### J. TEKKO

#### THE COMPARATIVE ANALYSIS OF QUIRK MODEL AND PETRI NETS BASED ON QUIRK MODEL LANGUAGE

In this paper two formalisms for describing parallel systems are compared — thoroughly studied Petri nets and a less known Quirk model. Because of principally different formalisms the comparing of Petri nets and Quirk model by this method is possible only by means of simplifying the Quirk model. It is proved that the investigated qualities of the formalisms are independent of this simplification.