СБОРНИК **НАУЧНЫХ ТРУДОВ НГТУ. – 2013. – № 4(74).** – 97–121

УДК 62-50:519.216

ИНВЕРСИЯ СЕТЕЙ ПЕТРИ*

А.В. МАРКОВ

В работе был описан математический аппарат сетей Петри, перечислены основные типы сетей. Предлагается математическое определение простой ординарной сети Петри – наиболее простой из всех возможных вариантов сетей. Описана необходимость анализа смоделированных систем и перечислены способы анализа, как традиционные: построение дерева достижимости, матричное представление сети, генерация отчёта о пространстве состояний, так и нетрадиционные: *sweep-line method, bitstate hashing*. Представлено математическая интерпретация дерева достижимости. Детально описаны два основных свойства сетей: ограниченность и достижимость. Обоснована актуальность доказательства достижимости и необходимость подтверждения данного свойства у определенных состояний. Доказательство достижимости предлагается реализовать с помощью инверсии системы. Инверсия заключается в изменении структуры сети и приводит к получению начальной маркировки, что свидетельствует о достижимости выбранной ранее маркировки.

Для реализации инверсии простой ординарной сети Петри предложен алгоритм, содержащий команды по изменению ориентации взаимосвязей между вершинами сети и изучению дерева достижимости. Инвертирование представлено на примере простой ординарной сети Петри «Взаимодействие пользователя и банкомата». Доказательством корректной реализации служат схожие разметки деревьев достижимости прямой и обратной систем.

Для преобразования простых сетей Петри может потребоваться более сложные изменения структуры системы. Поэтому предложен набор из девяти примеров инверсии небольших простых сетей Петри, которые помогут реализовать инверсию в более крупных системах. В заключении представлены основные результаты проделанной работы и предложено её последующее развитие: реализация инверсии сети Петри «Протокола передачи данных» с помощью правил, представленных в данной работе.

Ключевые слова: сети Петри, CPN Tools, простая ординарная сеть Петри, дерево достижимости, пространство состояний, ограниченность, достижимость, начальная маркировка, инверсия, прямая инверсия, взрыв пространства состояний, место-счётчик, место-условие.

ВВЕДЕНИЕ

---

* Статья получена 06 августа 2013 г.

[cpn] Снимок экрана, сделанный в приложении CPN Tools (version 3.4.0).

Сетью Петри называют двудольный ориентированный граф, в котором взаимосвязаны между собой вершины двух типов: места и переходы [1]. Данные вершины взаимосвязаны между собой направленными дугами; вершины одного типа не пересекаются. По сети передвигаются метки при срабатывании перехода, символизирующие всевозможные ресурсы системы.

Существуют различные виды сетей Петри: временная сеть Петри, стохастическая сеть Петри, функциональная сеть Петри, цветная сеть Петри, сеть Петри с ингибиторными дугами, иерархическая сеть и др., а также их всевозможные комбинации.

Самыми первыми и самыми простыми являются простые ординарные сети Петри[1], описать которые можно упорядоченным множеством $\mathcal{N}=\left(P,T,F,m_I\right)$, где $P=\left\{p_1,p_2,\ldots,p_n\right\}$ – множество мест, а $T=\left\{t_1,t_2,\ldots,t_m\right\}$ – множество переходов, таких, что $P\cap T=\varnothing$, $F\subseteq P\times T\cup T\times P$ – взаимосвязь вершин[2], и $m_I:P\to\mathbb{N}$ [2, стр. 80].

В некоторых случаях используют нотацию пре- и пост-множеств узлов[3] $x\in P\cup T$, т.е. $\cdot x=\{y\in P\cup T\mid\left(y,x\right)\in F\}$ и $x\cdot=\{y\in P\cup T\mid\left(x,y\right)\in F\}$.

Переход $t\in T$ разрешён[4] в маркировке $m:P\to\mathbb{N}$, если $m\geq\cdot t$. При соблюдении этих условий достижима маркировка m' и можно записать, как $m\left[t\right\rangle m'$, где m' определена как $m'=\left(m-\cdot t\right)+t\cdot$.

Используем R для обозначения количества достижимых маркировок, т. е. $R=\left|\left[m_I\right\rangle\right|$.

Возможность оценить достижимые маркировки появляется при анализе смоделированных сетей. Также анализ позволяет выявить мертвые и тупиковые состояния, при которых происходит прекращение работы системы, зацикливание процессов и выявление сценариев, не участвующих в работе системы.

Существующие виды анализа спроектированных сетей заключаются в построении дерева достижимости, в матричном представлении сети, а также в генерации отчета о пространстве состояний при помощи среды моделирования CPN Tools (version 3.4.0) [3–23].

---

[1] Простая ординарная сеть Петри – сети, имеющие единственный цвет (тип данных) у меток и кратность дуг не больше единицы. Между узлами прокладывается ровно одна связь.

[2] Взаимосвязь вершин – *flow relation* (англ.) [2, с. 80].

[3] Узел – *node* (англ.) [2, с. 80].

[4] Разрешён (имеется возможность сработать) – *enabled* (англ.) [2, с. 80].

Один из способов анализа смоделированных сетей Петри – это построение графа достижимости. Граф достижимости простой сети Петри – это направленный граф, содержащий вершины для каждой достижимой маркировки и рёбра для каждого возможного перехода от одной достижимой маркировки к другой.

Дерево достижимости – есть упорядоченное множество (V,E,src,trg), где V – множество вершин[5], E – множество рёбер[6], а $src,trg : E \rightarrow V$ отображает назначение для каждого ребра, как источник[7] или цель[8] [2, стр. 81].

$V = [m_I\rangle$. Множество узлов – это множество достижимых маркировок. $E = \{(m,t,m') \in V \times T \times V | m[t\rangle m'\}$. Множество рёбер – есть множество переходов из одной достижимой маркировки к другой. src для данных значений $(m,t,m') = m$. trg для данных значений $(m,t,m') = m'$.

Представить можно только конечное дерево достижимости, но для простой сети Петри он может быть и бесконечным. Соблюдая несколько условий, можно добиться конечного пространства состояний. Первое, что стоит отметить это: простая сеть Петри $\mathcal{N} = (P,T,F,m_I)$ должна иметь конечное множество мест, $|P| < \infty$, и конечное множество переходов, $|T| < \infty$. Второе условие: сеть должна быть k-ограниченна [2, стр. 81] для некоторого $k \in \mathbb{N}, k > 0$.

Простая сеть Петри (P,T,F,m_I) k-ограниченна, если и только если для всех $m \in [m_I\rangle$ и для всех $p \in p : m(p) \leq k$.

Также существуют нестандартные методики анализа пространства состояний, т. е. методики, использующие иные алгоритмы по отношению к традиционному – хранение всех посещенных состояний в оперативной памяти. Это *sweep-line method*[9] и *bit-state hashing*[10] и их возможные модификации. Данные методики имеют как положительные, так и отрицательные стороны. К достоинствам можно отнести – анализ систем с пространством состояний огромных

[5] Вершина – *enabled* (англ.) [2, с. 81].
[6] Ребро – *edge* (англ.) [2, с. 81].
[7] Источник – *source* (англ.) [2, с. 81].
[8] Цель – *target* (англ.) [2, с. 81].
[9] Метод плавающей линии – *sweep-line method* (англ.) [24]. Дословный перевод – метод линии-диапазона. Но более точно излагает суть перевод, предложенный авторами.
[10] Хеширование битового состояния – *bit-state hashing* (англ.) [25].

размеров, но явным недостатком является возможность частичного покрытия (объяснить термин) пространства состояний.

Одной из основной проблемой при анализ крупных систем является возможность взрыва пространства состояний[11]. Данная проблема заключается в экспоненциальном росте количества состояний, что приводит к досрочному завершению анализа из-за недостатка нужных объёмов оперативной памяти.

Следовательно, возможно без оценки останется значительная часть дерева достижимости, в которой могут присутствовать зацикливания и тупиковые маркировки, что свидетельствует о некорректном построении системы. Конечно, для хранения высчитанных состояний системы можно использовать постоянную память машины, но это приведет к значительному росту времени, требуемого на анализ [25, стр. 2].

Также во время моделирования системы может встать вопрос о проверке достижимости[12] полученной маркировки в целях начала анализа системы с выбранного состояния.

Данный вид анализа может потребоваться, когда перед проектировщиками стоит задача проверки определенной области состояний системы, либо исследования графа достижимости по частям [19]. При изучении пространства состояний системы частями отсутствует возможность проверки с начальной разметки и необходима уверенность в достижимости выбранного для анализа состояния.

Это означает, что выполнение свойства достижимости, присущее всем сетям Петри, является весьма важным, а задачи по его доказательству – актуальными.

В данной работе будет предложен способ доказать достижимость, выбранной маркировки, если заранее неизвестно попадает ли она в пространство состояний всей системы, а известно только начальная маркировка.

Данная работа устроена следующим образом: в пункте 1 описан предлагаемый способ доказательства достижимости состояния для простой ординарной сети, во 2-м пункте рассматривается доказательство для простой сети Петри, а в подпунктах пункта 2 описываются правила преобразования для простой сети Петри. В последней части работы приводится заключение и описываются полученные результаты, а также предположительные последующие работы.

---

[11] Взрыв пространства состояний – *state explosion problem* (англ.) [24].

[12] Достижимость сети Петри – возможность перехода сети из одного заданного состояния (характеризуемого распределением меток) в другое.

1. ИНВЕРСИЯ[13] ПРОСТОЙ ОРДИНАРНОЙ СЕТИ ПЕТРИ

Для проверки достижимости маркировки авторами работы предлагается произвести инверсию сети.

Инверсия будет происходить следующим образом: все дуги, присутствующие в системе, буду изменять свое направление на противоположное, тем самым, способствуя движению фишек в обратном направлении. Движение фишек в обратном направлении, в свою очередь, должно привести к начальному состоянию сети.

Покажем данную операцию на системе, состоящей из двух мест $P = \{A, B\}$ и одного перехода $T = \{a\}$ (рис. 1, слева). Инвертируем, представленную модель, по средствам смены ориентации взаимосвязей между вершинами (рис. 1, справа), а начальным состоянием выберем состояние № 2 непреобразованной сети.

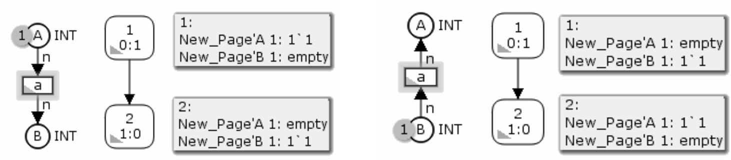


Рис. 1. Система, состоящая из двух мест: слева – сеть Петри и её дерево достижимости, справа – инвертированная сеть Петри и ее дерево

достижимости^{cpn}

После изучения инвертированной модели стоит отметить, что достигается начальная маркировка системы – состояние № 2 (рис. 1, справа).

Рассмотрим возможность инвертирования на примере простой ординарной сети Петри $\mathcal{N} = (P, T, F, m_I)$ (рис. 2). и

$T = \{Insert\,card,\ Inject\,PIN,\ PIN\,correct,\ Taking\,card,\ Withdrawal,$

$Request\,amount,\ Enter\,amount,\ Payment\,of\ money,\ Print\,chq\,III\}$,

$m_I = (1,0,0,0,0,0,0,0,0)$ (рис. 3), $p \in p : m(p) = 1$.

[13] Инверсия – изменение расположения, либо ориентации, элементов сети Петри в особом порядке, нарушающем обычный (прямой) порядок, с целью изменить движение меток на обратный.

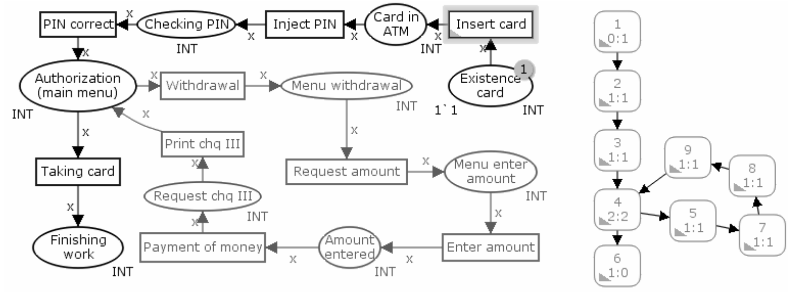


Рис. 2. Простая ординарная сеть Петри (слева) и её дерево достижимости (справа)^{cpn}

(1) 1 0 0 0 0 0 0 0 0

↓

(2) 0 1 0 0 0 0 0 0 0

↓

(3) 0 0 0 1 0 0 0 0 0

↓

(4) 0 0 0 0 1 0 0 0 0 → (9) 0 0 0 0 0 0 0 1 0 ← (8) 0 0 0 0 0 0 0 0 1

(4) 0 0 0 0 1 0 0 0 0 → (5) 0 0 0 0 0 1 0 0 0 → (7) 0 0 0 0 0 0 1 0 0 → (8)

↓

(6) 0 0 1 0 0 0 0 0 0

Рис. 3. Дерево достижимости простой ординарной сети Петри с маркировкой для каждого состояния

Представленная сеть является одним из сценариев работы [14] взаимодействия пользователя и банкомата. В работе [14] данный сценарий был проанализирован с последующим построением пространства состояний, достижимость всех маркировок была доказана. Идеей, лежащей в основе данного пункта, является инвертирование простой ординарной сети, с целью доказательства достижимости выбранной маркировки – возможность перехода к ней из начального состояния.

Маркировка, при которой происходило инвертирование, представлена на (рис. 4). При таком состоянии в системе место «Amount entered» имеет одну фишку.

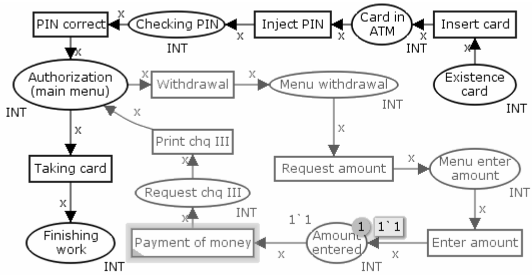


Рис. 4. Простая ординарная сеть Петри в состоянии «Amount entered»^{cpn}

Чтобы проверить, находится ли выбранное состояние в пространстве состояний всей системы, нужно доказать достижимость этой маркировки из начального состояния. Для чего нужно инвертировать сеть (рис. 2) посредством простой смены ориентации всех дуг $\mathcal{N}_i = (P,T,F_i,m_i)$, где $\mathcal{N}_i$ – инвертированная простая сеть Петри, F_i – инвертированная направленная взаимосвязь, m_i – маркировка, при которой произошла инверсия.

Алгоритм 1. Инверсия простой ординарной сети Петри и вычисление дерева достижимости

$$while\, F \subseteq P \times T \cup T \times P$$

$$select\; p \xrightarrow{t} p'$$

$$F := F \setminus \left\{ \left(p \xrightarrow{t} p' \right) \right\}$$

$$for\,all\,(t, p')\,such\,that\; p \xrightarrow{t} p'\,do\,invert\; p' \xrightarrow{t} p$$

$$return(P,T)$$

$$V := \{S_i\}$$

$$W := \{S_i\}$$

$$E :\neq 0$$

$$while\, W \neq \{S_I\}\, do$$

$$select\, an\, s \in W$$

$$W := W\ \{s\}$$

$$for\ all\ t, s'\ such\ that\ s \xrightarrow{t} s'\ do$$

$$E := E \cup \{(s\ ,t,s')\}$$

$$if\ s' \notin\ V\ then$$

$$V := V \cup \{S'\}$$

$$W := W \cup \{S'\}$$

$$return(V,E,W)$$

После выполнения алгоритма, представленного выше, имеем сеть (рис. 5) и следующую маркировку состояний (рис. 6).

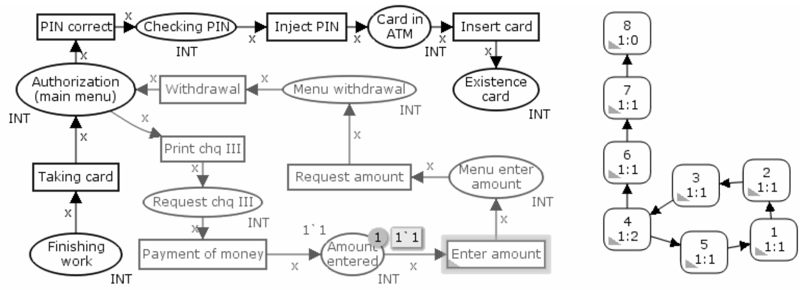


Рис. 5. Инвертированная сеть Петри (слева) и дерево достижимости инвертированной сети Петри (справа)^{cpn}

(8) 1 0 0 0 0 0 0 0 0

↑

(7) 0 1 0 0 0 0 0 0 0

↑

(6) 0 0 0 1 0 0 0 0 0

↑

(4) 0 0 0 0 1 0 0 0 0

(3) 0 0 0 0 0 1 0 0 0 ← (2) 0 0 0 0 0 0 1 0 0

(5) 0 0 0 0 0 0 0 1 0 → (1) 0 0 0 0 0 0 0 0 1

Рис. 6. Дерево достижимости инвертированной сети Петри с маркировкой для каждого состояния

Построенное дерево достижимости для инвертированной сети Петри (рис. 5), (V, E_i, src, trg), где E_i – инвертированное множество рёбер, показало, что есть возможность достичь начальной маркировки сценария (рис. 2) из состояния «Amount entered». Если сравнить деревья, представленные на рис. 2 и рис. 5, видно, что состояние 6 (рис. 2) отсутствует в дереве инвертированной сети (рис. 5). Это говорит о возможности потери состояний во время инверсии сетей. А именно, часть сети, состоящая из места «Finishing work» и перехода «Taking card», не присутствует в данном дереве достижимости, так как нет возможности достигнуть этих вершин из выбранной разметки (рис. 4), тем самым они выпадают из пространства состояний. Но так как задачей ставилось доказательство достижимости выбранной маркировки, данной потерей можно пренебречь.

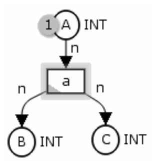
Стоит отметить, что прямая инверсия[14] возможна только для простых ординарных сетей. При наличии набора условий у переходов и дуг стоит ввести определенный набор правил, при выполнении которых разработчик получит начальную маркировку, используя данный способ.

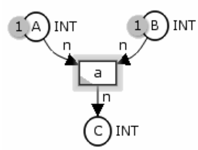
2. ИНВЕРСИЯ ПРОСТОЙ СЕТИ ПЕТРИ

В примере, приведенном выше, рассматривалась простая ординарная сеть. Инверсия данной сети осуществлялась сменой ориентации дуг, соединяющих места и переходы. Результатом чего стала возможность достижения начальной маркировки из любого другого состояния системы, что может служить доказательством достижимости выбранной маркировки.

Ниже будут рассмотрены правила реверсирования простой сети, у которой, в отличии от простой ординарной, появляется возможность использовать любое количество дуг, направленных от мест к переходам и от переходов к местам.

Правила были выведены, опираясь на анализ небольших сетей Петри (рис. 7–15), которые могут являться фрагментами более крупной сети Петри.

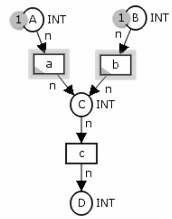




[14] Прямая инверсия – инверсия, при которой происходит только смена ориентации дуг между вершинами сети.

Рис. 7. Пример № 1 сети Петри^{cpn}

Рис. 8. Пример № 2 сети Петри^{cpn}



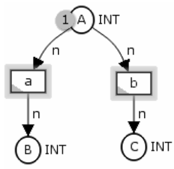
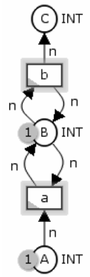


Рис. 9. Пример № 3 сети Петри^{cpn}

Рис. 10. Пример № 4 сети Петри^{cpn}



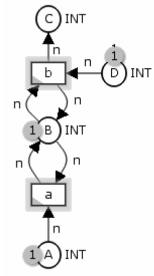
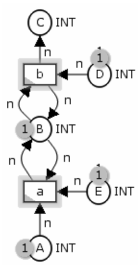


Рис. 11. Пример № 5 сети Петри^{cpn}

Рис. 12. Пример № 6 сети Петри^{cpn}



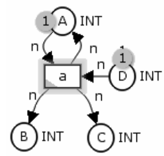


Рис. 13. Пример № 7 сети Петри^{cpn}

Рис. 14. Пример № 8 сети Петри^{cpn}

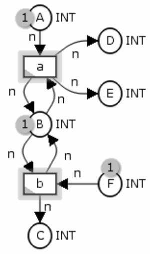


Рис. 15. Пример № 9 сети Петри^{cpn}

Рассмотрим приведённые примеры более подробно с отображением дерева состояний и представлением сетей Петри инвертированного вида.

2.1. Пример сети Петри № 1, № 2

Для данного случая сеть Петри имеет три места: $P=\{A,B,C\}$ и один переход $T=\{a\}$ и ее дерево достижимости имеет две возможные маркировки $m_1[a\rangle m_2=(1,0,0)[a\rangle(0,1,1)$, а $m_I=m_1$ (рис. 16).

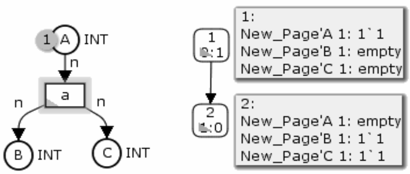


Рис. 16. Пример № 1 сети Петри
и дерева достижимости^{cpn}

Как и в случае простой ординарной сети, данная система будет преобразовываться путём прямой инверсии. Трансформированная сеть с деревом достижимости представлена на рис. 17.

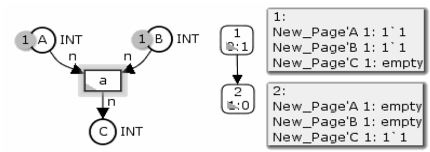


Рис. 17. Инвертированная сеть с деревом
достижимости примера № 1^{cpn}

Стоит отметить, что пример сети Петри №1 и №2 являются инвертированными сетями по отношению к друг другу. То есть при выполненной инверсии[15] для первого случая можно с уверенностью сказать, что будет получена сеть, которая представлена в примере № 2 (рис. 8).

2.2. Пример сети Петри № 3

В данном примере сеть имеет четыре места и три перехода с начальной маркировкой $m_I = (1,1,0,0)$ (рис. 18).

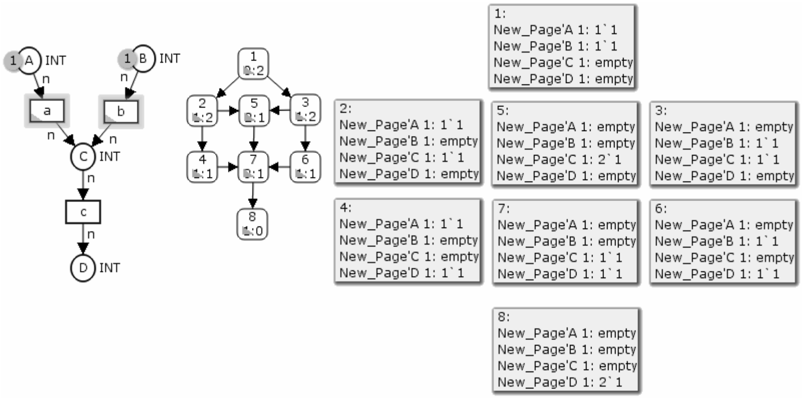


Рис. 18. Пример № 3 сети Петри и дерева достижимости[cpn]

Дерево достижимости (рис. 18) имеет восемь состояний $V = \{m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8\}$ и десять рёбер между ними $E = \{m_1[b\rangle m_2,\ m_1[a\rangle m_3,\ m_2[a\rangle m_5,\ m_3[b\rangle m_5,\ m_2[c\rangle m_4,\ m_3[c\rangle m_6,$
$m_5[c\rangle m_7,\ m_4[a\rangle m_7,\ m_6[b\rangle m_7,\ m_7[b\rangle m_8\}$

К данному примеру также применим способ прямой инверсии, который при смене направления взаимосвязей между вершинами, приведёт к начальной маркировки из любого состояния сети.

[15] Выполненная инверсия – инверсия, после которой получена полностью трансформированная сеть с возможностью воссоздания дерева достижимости.

После инвертирования сеть принимает форму (рис. 19), схожую с начальной сетью. Отличие заметно лишь в направлении дуг.

Стоит отметить, что в дереве, представленном на рис. 19, после инвертирования появились дополнительные маркировки m_8, m_{10}, наличие которых в примере № 3 не обговариваются. Следовательно, данные состояния нужно считать ошибочными, но необходимо учесть, что начальная маркировка достигается. Тем самым выполняется поставленная цель для инвертирования – достижение начальной маркировки из выбранного состояния.

Зададим для сети примера № 3 дополнительно две разных маркировки, которые исключают фишку из одного места: $m_I = (1,0,0) \neg m_I = (0,1,0)$.

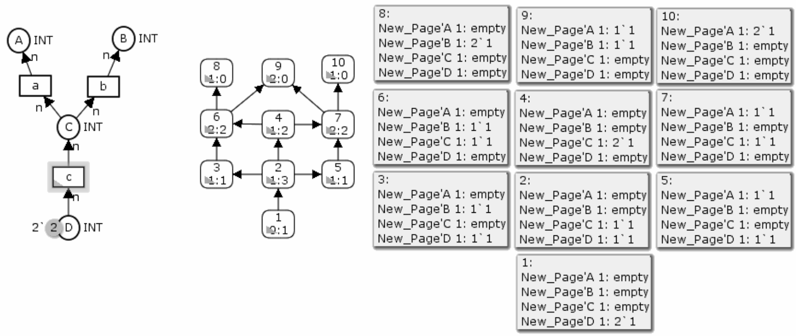


Рис. 19. Инвертированная сеть с деревом достижимости примера № 3^{cpn}

В первом случае сеть выглядит (рис. 20) так же, как и на рис. 18, только в месте *B* отсутствует фишка.

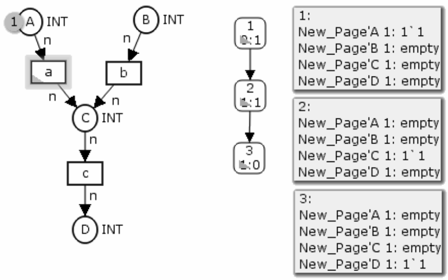


Рис. 20. Сеть Петри и дерево достижимости примера № 3
с начальной маркировкой $m_I = (1,0,0)^{cpn}$

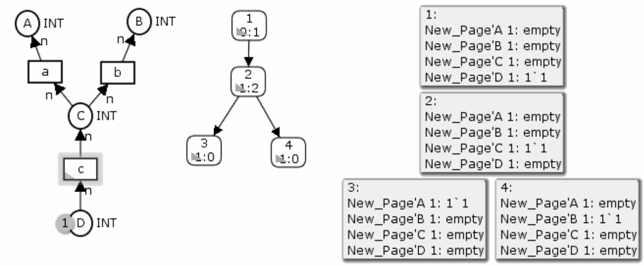


Рис. 21. Инвертированная сеть Петри и дерево достижимости примера №3 с начальной маркировкой $m_I = (1,0,0)^{cpn}$

При инвертировании сети (рис. 21) у дерева достижимости появляется лишняя маркировка m_4, которая невозможна у прямой сети Петри[16].

Данная маркировка m_4 является ошибочной, но и при её наличии можно достигнуть начального состояния. Аналогичные результаты были получены при начальной маркировки $m_I = (1,0,0)$.

2.3. Пример сети Петри № 4

В данном случае (рис. 22) может сработать любой из переходов $\left(A[a\rangle B \neg A[b\rangle C\right)$. После чего место *B* либо место *C* будет содержать фишку.

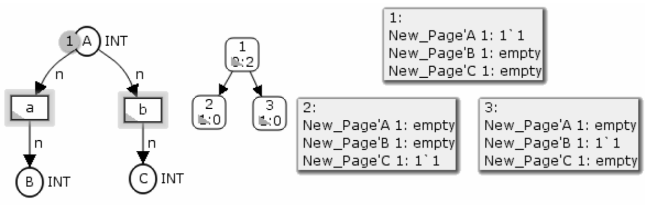


Рис. 22. Пример № 4 сети Петри и дерева достижимости^{cpn}

Так как в данном примере возможно инвертировать два состояния $m_2 = (0,0,1)$ и $m_3 = (0,1,0)$, представим инвертированные сети и их дерева достижимости для обоих случаев (рис. 23).

[16] Прямая сеть Петри – сеть Петри до инверсии.

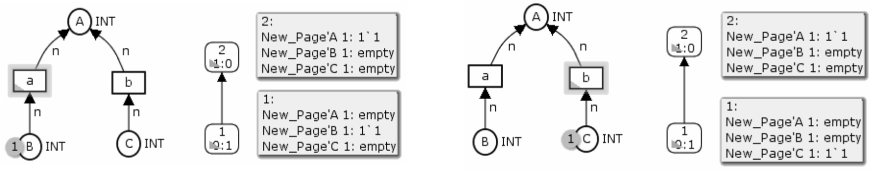


Рис. 23. Инверсированные сети с деревьями достижимости примера № 4[cpn]

Во время инверсии в пространстве состояний каждого из случаев (рис. 23) происходит потеря одного из места, тем не менее начальная маркировка достигается с сохранением структуры прямого дерева, что свидетельствует о корректно проведенной инверсии.

2.4. Пример сети Петри № 5

В этом примере (рис. 11) сеть усложняется добавлением возвратных дуг и добавлением места-счётчика[17].

В связи с тем, что в моделируемой среде *CPN Tools* (version 3.4.0) нет возможности сгенерировать пространство состояний системы, содержащей счётчик, представим дерево достижимости, построенное вручную (рис. 24).

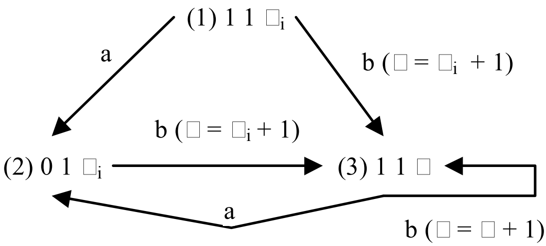


Рис. 24. Дерево достижимости сети Петри примера № 5

Здесь ω_i – начальное значение счётчика, а ω – текущее значение.

Поскольку увеличения счётчика всегда происходит на единицу, к некоторым дугам дерева состояний добавлено условие, которое инкрементирует ω, тем самым увеличивая его значение в самом дереве состояний. Данный способ

---

[17] Место-счётчик – счётчиком можно считать место, в котором количество фишек увеличивается на постоянную величину.

может компактно и наглядно отобразить деревья достижимости систем, в которых находятся счётчики [26].

Нужно отметить, что место-счётчик инверсии не имеет, так как заранее неизвестно, сколько фишек он накопит и как неопределенное количество фишек будет преобразовываться в определенное количество.

2.5. Пример сети Петри № 6

Данный пример (рис. 25) отличается от примера № 5 тем, что место-счётчик *C* может накапливать только одну фишку. Дерево достижимости для данной сети имеет четыре состояния.

При прямом инвертировании данной структуры получаем сеть (рис. 26), которая была получена при изменении ориентации взаимосвязей между вершинами.

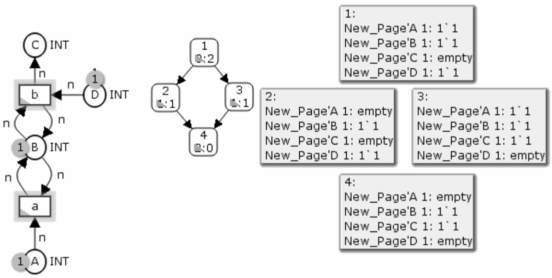


Рис. 25. Пример № 6 сети Петри и дерева достижимости^{cpn}

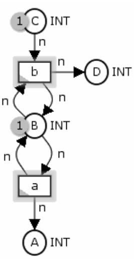


Рис. 26. Пример прямой инверсии сети Петри № 6^{cpn}

Стоит обратить внимания, что для данного случая прямая инверсия не подходит, так как ребро (*B, a, A*) образует счётчик. Место-счётчик *A* значительно повлияет на структуру дерева достижимости, а его сходство с пространством состояний (рис. 25) будет незначительным.

Для сети (рис. 25) прямая инверсия не подходит и стоит использовать иной метод преобразования, который сможет сохранить структуру пространства состояний (рис. 27).

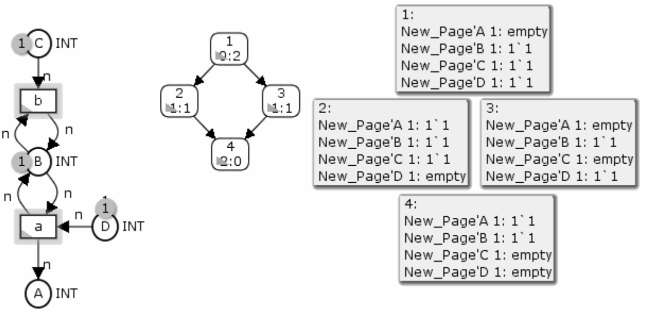


Рис. 27. Инверсированная сеть с деревом достижимости примера № 6^{cpn}

Из данной сети видно, что в каждой разметке дерева достижимости: $V = \{(0,1,1,1), (1,1,1,0), (0,1,0,1), (1,1,0,0)\}$, четвёртая цифра соответствует месту-условию[18] и не влияет на всю картину пространства состояний. Следует отметить, что изначально у обоих деревьев четвёртая цифра в разметке равна единице, а у последнего состояния равна нулю. Необходимо учитывать данную особенность при анализе инвертированных пространств состояний.

2.6. Пример сети Петри № 7

Сеть (рис. 28) является расширением примера № 6 – добавлено дополнительное место-условие E.

При инвертировании сети (рис. 28) посредством смены ориентации дуг у рёбер $F_{DbC} \subseteq D \times b \cup b \times C$ и $F_{EaA} \subseteq E \times a \cup a \times A$, получим сеть (рис. 29), в которой появляется сразу два места-счетчика A и E.

[18] Место-условие – место, при наличии фишки в котором, разрешено срабатывания связанного с ним перехода.

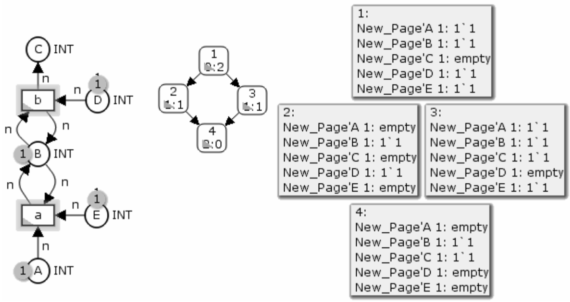


Рис. 28. Пример № 7 сети Петри и дерева достижимости^{cpn}

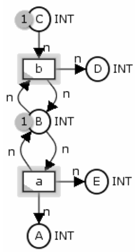


Рис. 29. Пример прямой инверсии сети Петри № 7^{cpn}

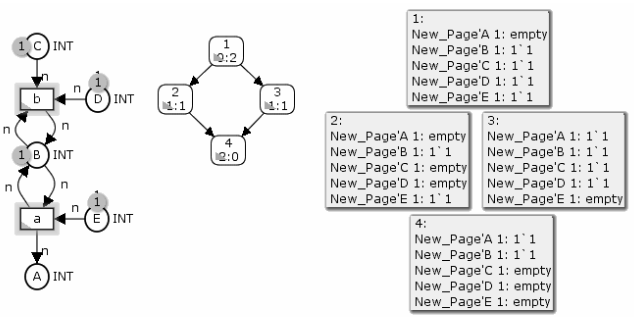


Рис. 30. Инвертированная сеть с деревом достижимости примера № 7^{cpn}

Так как в системе присутствует два места-счётчика, обратное дерево достижимости будет отличаться от прямого, следовательно, преобразование сети (рис. 28) выполнено неверно.

В данном случае прямая инверсия не удовлетворяет условию одинаковой структуры деревьев достижимости прямой и обратной сетях. Для корректного преобразования следует изменить направление дуг только у рёбер $F_{BaA} \subseteq B \times a \cup a \times A$ и $F_{CbB} \subseteq C \times b \cup b \times B$ (рис. 30).

После предложенной инверсии, отсутствует надобность менять расположение мест D и E.

2.7. Пример сети Петри № 8

Сети Петри (рис. 31) имеет два места-счётчика B и C и одно место-условие D.

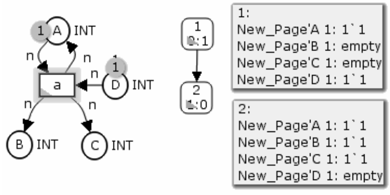


Рис. 31. Пример № 8 сети Петри и дерева достижимости^{срп}

При прямой инверсии сети (рис. 31) получаем обратную сеть с деревом состояний (рис. 32), структура которой подобна пространству состояний прямой сети.

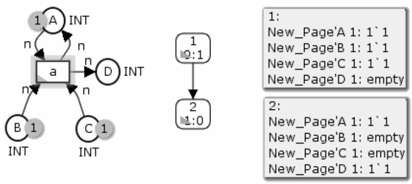


Рис. 32. Инвертированная сеть с деревом достижимости примера № 8^{срп}

В данном случае видно, что дерево состояний сети (рис. 31) является зеркальным отражением пространства состояний сети (рис. 32), что означает о корректном преобразовании.

Стоит отметить, что для структуры системы данного виды (рис. 33) можно использовать прямую инверсию и быть уверенным, что не будут утеряны или получены новые состояния.

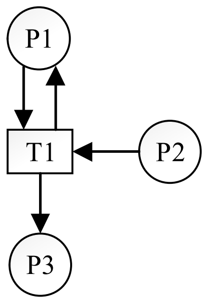


Рис. 33. Пример структуры сети Петри, при которой можно использовать прямую инверсию

2.8. Пример сети Петри № 9

Опираясь на правила, предложенные выше, можно утверждать, что структура системы (рис. 34) $P=\{A,B,C,D,E,F\}$, $T=\{a,b\}$, $m_I=(1,1,1,0,0,0)$; $\{A,B\}[a\rangle\{B,D,E\}$, $\{B,F\}[b\rangle\{B,C\}$, позволяет использовать прямое инвертирование сети без потери информации о пространстве состояний для нахождения начального состояния системы.

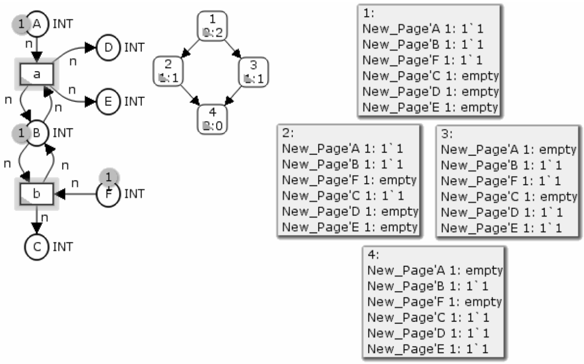


Рис. 34. Пример № 9 сети Петри и дерева достижимости^{cpn}

Обратная сеть выглядит следующим образом: $P=\{A,B,C,D,E,F\}$, $T=\{a,b\}$, $m_I=(0,1,0,1,1,1)$; $\{B,D,E\}[a\rangle\{A,B\}$, $\{C,B\}[b\rangle\{B,F\}$ (рис. 35).

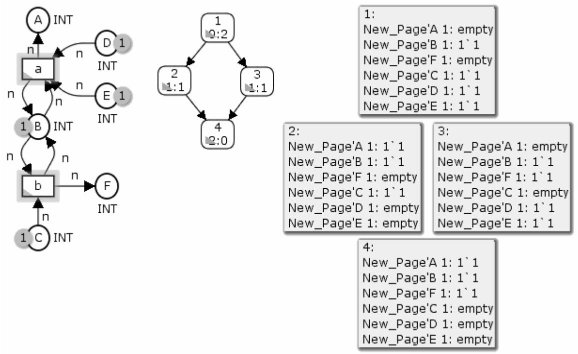


Рис. 35. Инвертированная сеть с деревом достижимости примера № 9^{cpn}

А дерево достижимости имеет схожую структуру: $V=\{m_1,m_2,m_3,m_4\}$, $E=\{(m_1,a,m_2),\,(m_1,b,m_3),\,(m_2,b,m_4),\,(m_3,a,m_4)\}$.

ЗАКЛЮЧЕНИЕ

В данной работе с представленным определением сети Петри приводится описание её свойств: ограниченности и достижимости.

Доказательство достижимости одно из основных свойств сети Петри, которое необходимо проектировщику для проверки наличия данного состояния во всём пространстве состояний системы. Это может потребоваться, когда разработчик будет анализировать определенную часть дерева достижимости или ему необходимо начать анализ с определённой разметки сети.

Также это может потребоваться для анализа пространства состояний по частям, когда бывает тяжело сразу определить достижимость выбранного состояния.

Чтобы доказать достижимость в данной работе предлагается использовать инверсию системы. Данный способ был детально показан на примере простой ординарной сети Петри. Стоит уточнить, что для простой ординарной сети подходит прямая инверсия. Для простых сетей Петри, в свою очередь, стоит добавить ряд правил, так как в системе возможно неограниченное количество входных и выходных дуг у вершин.

Предложенный набор состоит из девяти правил, который будут справляться с инвертированием простых сетей Петри.

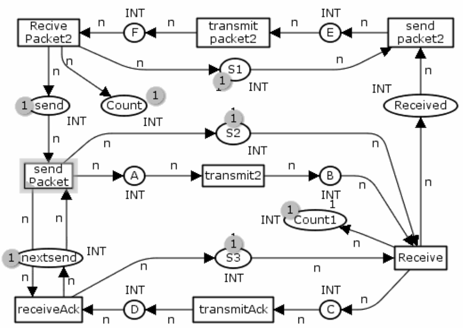


Рис. 36. Протокол передачи данных[cpn]

Проверку предложенных правил желательно осуществить на примере более сложной сети, которая является классической задачей. Для данных целей будет выбрана сеть Петри «Протокола передачи данных» [11] (рис. 36).

[1] *Питерсон Дж.* Теория сетей Петри и моделирование: пер. с англ. / Дж. Питерсон. – М.: Мир, 1984.

[2] *Westergaard M.* Behavioral verification and visualization of formal models of concurrent systems: PhD dissertation / M. Westergaard. – Aarhus: University of Aarhus, 2007.

[3] *Воевода А.А.* О компактном представлении языков раскрашенных сетей Петри / А.А. Воевода, Д.О. Романников // Сб. науч. тр. НГТУ. – 2008. – № 3(53). – С. 105–108.

[4] *Воевода А.А.* О компактном представлении языков сетей Петри: сети с условиями и временные сети / А.А. Воевода, А.В. Марков // Сб. науч. тр. НГТУ. – 2010. – № 2(60). – С. 77–83.

[5] *Коротиков С.В.* Применение сетей Петри в разработке программного обеспечения центров дистанционного контроля и управления: дис. … канд. техн. наук / С.В. Коротиков. – Новосибирск: НГТУ, 2007.

[6] *Марков А.В.* Моделирование процесса поиска пути в лабиринте при помощи сетей Петри / А.В. Марков. // Сб. науч. тр. НГТУ. – 2010. – № 4(62). – С. 133–141.

[7] *Романников Д.О.* Обзор работ посвященным разработке ПО с использованием UML и сетей Петри / Д.О. Романников, А.В. Марков, И.В. Зимаев // Сб. науч. тр. НГТУ. – 2011. – № 1(63). – С. 91–104.

[8] *Марков А.В.* Моделирование процесса поиска пути в лабиринте при помощи сетей Петри для системы из двух связных звеньев / А.В. Марков, А.А. Воевода // Сб. науч. тр. НГТУ. – 2011. – № 3(65). – С. 95–104.

[9] *Марков А.В.* Поиск манипулятором кратчайшего пути в лабиринте / А.В. Марков // Сб. науч. тр. НГТУ. – 2011. – № 4(66). – С. 75–91.

[10] *Романников Д.О.* Пример применения методики разработки ПО с использованием UML-диаграмм и сетей Петри / Д.О. Романников, А.В. Марков // Научный вестник НГТУ. – 2012. – № 1(67). – С. 175–181.

[11] *Романников Д.О.* Разработка программного обеспечения с применением UML-диаграмм и сетей Петри для систем управления локальным оборудованием: дис. … канд. техн. наук / Д.О. Романников. – Новосибирск: Изд-во НГТУ, 2012.

[12] *Воевода А.А.* Рекурсия в сетях Петри / А.А Воевода, А.В. Марков // Сб. науч. тр. НГТУ. – 2012. – № 3(69). – С. 115–122.

[13] *Воевода А.А.* Понятие рекурсии в сетях Петри: факториал числа, числа Фибоначчи / А.А. Воевода, А.В. Марков // Сб. науч. тр. НГТУ. – 2013. – № 1(71). – С. 72–77.

[14] *Марков А.В.* Анализ сетей Петри при помощи деревьев достижимости / А.В. Марков, А.А. Воевода // Сб. науч. тр. НГТУ. – 2013. – № 1(71). – С. 78–95.

[15] *Марков А.В.* Разработка программного обеспечения при совместном использовании UML-диаграмм и сетей Петри (обзор) / А.В. Марков // Сб. науч. тр. НГТУ. – 2013. – № 1(71). – С. 96–131.

[16] *Марков А.В.* Анализ иерархических сетей Петри / А.В. Марков, А.А. Воевода // Вестник МГТУ им. Н.Э. Баумана. – 2013. – в печати.

[17] *Марков А.В.* Развитие системы «Перемещение манипулятора в пространстве с препятствиями» при помощи рекурсивных функций / А.В. Марков, А.А. Воевода // Автоматика и программная инженерия. – 2013. – № 2(4). – С. 35–41.

[18] *Марков А.В.* Матричное представление сетей Петри / А.В. Марков // Сб. науч. тр. НГТУ. – 2013. – № 2(71). – С. 61–67.

[19] *Марков А.*В. Анализ отдельных частей дерева достижимости сетей Петри / А.В. Марков // Сб. науч. тр. НГТУ. – 2013. – № 3(73). – С. 58–74.

[20] *Воевода А.А.* Применение UML-диаграмм и сетей Петри при разработке встраиваемого программного обеспечения / А.А. Воевода, Д.О. Романников // Научный вестник НГТУ. – 2009. – № 4 (37). – С. 169–174.

[21] *Воевода А.А.* Редуцирование пространства состояний сети Петри для объектов из одного класса / А.А. Воевода, Д.О. Романников // Научный вестник НГТУ. – 2011. – № 4 (45). – С. 146–150.

[22] *Воевода А.А.* О модификации полного покрывающего дерева и графа разметок сети Петри / А.А. Воевода, С.В. Коротиков // Научный вестник НГТУ. – 2005. – № 1 (19). – С. 171–172.

[23] *Коротиков С.В.* Применение сетей Петри в разработке программного обеспечения центров дистанционного управления и контроля / С.В. Коротиков, А.А. Воевода // Научный вестник НГТУ. – 2007. – № 4 (29). – С. 16–30.

[24] *Christensen S.* A sweep-line method for state space exploration / S. Christensen, L.M. Kristensen, T. Mailund // Springer-Verlag Berlin Heidelberg. – 2001. – P. 450–464.

[25] *Holzmann G.J.* An analysis of bitstate hashing / G.J. Holzmann // Symposium on Protocol Specification. – 1995. – № 15. – P. 301–314.

[26] *Geeraerts G.* ω-Petri Nets / G. Geeraerts, A. Heußner, M. Praveen, J-F. Raskin // Application and Theory of Petri Nets and Concurrency. Italy. – 2013. – № 34. – P. 49–69.

REFERENCES

[1] Piterson Dzh. Teorija setej Petri i modelirovanie: per. s angl. Moscow, Mir, 1984, 264 p.

[2] Westergaard M. Behavioral verification and visualization of formal models of concurrent systems: PhD dissertation. Aarhus: University of Aarhus, 2007.

[3] Voevoda A.A., Romannikov D.O. O kompaktnom predstavlenii jazykov raskrashennyh setej Petri. Sb. nauch. tr. NGTU, 2008, no. 3(53), pp. 105–108.

[4] Voevoda A.A., Markov A.V. O kompaktnom predstavlenii jazykov setej Petri: seti s uslovijami i vremennye seti. Sb. nauch. tr. NGTU, 2010, no. 2(60), pp. 77–83.

[5] Korotikov S.V. Primenenie setej Petri v razrabotke programmnogo obespechenija centrov distancionnogo kontrolja i upravlenija. Diss. kand. tehn. nauk. Novosibirsk, NGTU, 2007.

[6] Markov A.V. Modelirovanie processa poiska puti v labirinte pri pomoshhi setej Petri. Sb. nauch. tr. NGTU, 2010, no. 4(62), pp. 133–141.

[7] Romannikov D.O., Markov A.V., Zimaev I.V. Obzor rabot posvjashhennym razrabotke PO s ispol'-zovaniem UML i setej Petri. Sb. nauch. tr. NGTU, 2011, no. 1(63), pp. 91–104.

[8] Markov A.V., Voevoda A.A. Modelirovanie processa poiska puti v labirinte pri pomoshhi setej Petri dlja sistemy iz dvuh svjaznyh zven'ev. Sb. nauch. tr. NGTU, 2011, no. 3(65), pp. 95–104.

[9] Markov A.V. Poisk manipuljatorom kratchajshego puti v labirinte. Sb. nauch. tr. NGTU, 2011, no. 4(66), pp. 75–91.

[10] Romannikov D.O., Markov A.V. Primer primenenija metodiki razrabotki PO s ispol'zovaniem UML-diagramm i setej Petri. Nauchnyj vestnik NGTU, 2012, no. 1(67), pp. 175–181.

[11] Romannikov D.O. Razrabotka programmnogo obespechenija s primeneniem UML-diagramm i setej Petri dlja sistem upravlenija lokal'nym oborudovaniem. Diss. kand. tehn. nauk. Novosibirsk, Izd-vo NGTU, 2012.

[12] Voevoda A.A., Markov A.V. Rekursija v setjah Petri. Sb. nauch. tr. NGTU, 2012, no. 3(69), pp. 115–122.

[13] Voevoda A.A., Markov A.V. Ponjatie rekursii v setjah Petri: faktorial chisla, chisla Fibonachchi. Sb. nauch. tr. NGTU, 2013, no. 1(71), pp. 72–77.

[14] Markov A.V., Voevoda A.A. Analiz setej Petri pri pomoshhi derev'ev dostizhimosti. Sb. nauch. tr. NGTU, 2013, no. 1(71), pp. 78–95.

[15] Markov A.V. Razrabotka programmnogo obespechenija pri sovmestnom ispol'zovanii UML-diagramm i setej Petri (obzor). Sb. nauch. tr. NGTU, 2013, no. 1(71), pp. 96–131.

[16] Markov A.V., Voevoda A.A. Analiz ierarhicheskih setej Petri. Vestnik MGTU im. N.Je. Baumana, 2013, v pechati.

[17] Markov A.V., Voevoda A.A. Razvitie sistemy «Peremeshhenie manipuljatora v pro-stranstve s prepjatstvijami» pri pomoshhi rekursivnyh funkcij. Avtomatika i programmnaja inzhenerija, 2013, no. 2(4), pp. 35–41.

[18] Markov A.V. Matrichnoe predstavlenie setej Petri. Sb. nauch. tr. NGTU, 2013, no. 2(71), pp. 61–67.

[19] Markov A.V. Analiz otdel'nyh chastej dereva dostizhimosti setej Petri. Sb. nauch. tr. NGTU, 2013, no. 3(73), pp. 58–74.

[20] Voevoda A.A., Romannikov D.O. Primenenie UML-diagramm i setej Petri pri razrabotke vstraivaemogo programmnogo obespechenija. Nauchnyj vestnik NGTU, 2009, no. 4(37), pp. 169–174.

[21] Voevoda A.A., Romannikov D.O. Reducirovanie prostranstva sostojanij seti Petri dlja obektov iz odnogo klassa. Nauchnyj vestnik NGTU, 2011, no. 4(45), pp. 146–150.

[22] Voevoda A.A., Korotikov S.V. O modifikacii polnogo pokryvajushhego dereva i grafa razmetok seti Petri. Nauchnyj vestnik NGTU, 2005, no.1(19), pp. 171–172.

[23] Korotikov S.V., Voevoda A.A. Primenenie setej Petri v razrabotke programmnogo obespechenija centrov distancionnogo upravlenija i kontrolja. Nauchnyj vestnik NGTU, 2007, no. 4(29), pp. 16–30.

[24] Christensen S., Kristensen L.M., T. Mailund T. A sweep-line method for state space exploration. Springer-Verlag Berlin Heidelberg, 2001, pp. 450–464.

[25] Holzmann G.J. An analysis of bitstate hashing / G.J. Holzmann // Symposium on Protocol Specification. – 1995. – № 15. – P. 301–314.

[26] Geeraerts G., Heußner A., Praveen M., Raskin J-F. ω-Petri Nets. Application and Theory of Petri Nets and Concurrency, Italy, 2013, no.34, pp. 49–69.

Марков Александр Владимирович – аспирант кафедры автоматики Новосибирского государственного технического университета. Основное направление научных исследований – системный анализ, управление и обработка информации в промышленности. Имеет более 15 публикаций. E-mail: muviton3@gmail.com

A.V. Markov

Inversion of Petri nets

The work was described mathematical formalism of Petri nets are the main types of nets. The mathematical definition of a simple ordinary Petri nets - the simplest of all possible nets. Described the need to analyze the simulated systems and methods of analysis are listed as traditional: tree building reachability, matrix representation of net, report generation state space and non-traditional: sweep-line method, bitstate hashing. Presented mathematical interpretation reachability tree. Described in detail two main properties of Petri nets: limited and achievable. The urgency of evidence admissibility and need to confirm this property in certain states.

Proof reachability proposed to implement with inversion system. Inversion is to change the structure and results in the initial marking, indicating that the previously selected reachable marking. To implement a simple inversion of the ordinary Petri net, an algorithm that contains the commands to change the orientation relationships between the vertices of the net and exploration the reachability tree.

Inverting exemplified by a simple ordinary Petri nets "User interaction and an ATM". Proof of correct implementation are similar markup trees reachable direct and inverse systems.To convert ordinary Petri nets may require more complex changes in the structure of the system. Therefore, proposed a set of nine small simple examples of inversion of Petri nets, which will help to realize the inversion in larger systems. In conclusion, the main results of the work done and suggested its further development: implementation of the inversion Petri net "Protocol Data" using the rules presented in this paper.

Key words: Petri Nets, CPN Tools, ordinary Petri net, reachability tree, the state space, limited, achievable, initial marking, inversion, direct inversion, state space explosion, place-counter, place-condition.