

УДК 004.4'22

А.В. Марков, А.А. Воевода

## Проверка достижимости маркировки сетей Петри при помощи инвертирования деревьев состояний для протокола передачи данных

По окончании проектирования программного обеспечения необходим анализ, способный выявить возможные недостатки в системе. Одним из таких способов является исследование системы с произвольно выбранного состояния, в котором может оказаться спроектированная модель при её функционировании. Для использования данного подхода обязательным является доказательство достижимости выбранного состояния, которая будет заключаться в инверсии пространства состояний с последующим восстановлением сети. Данная операция однозначна и может быть полезна для моделей со сложной структурой. Инверсия продемонстрирована на примере протокола передачи данных при помощи правил, предложенных в работе по преобразованию простой сети Петри.

**Ключевые слова:** сети Петри, анализ пространства состояний, достижимость, инверсия, протокол передачи данных.

На завершающих этапах проектирования алгоритмов, моделирования систем и создания программного обеспечения (ПО) необходим их анализ. На данный момент перспективным является использование аппарата сетей Петри, который реализует анализ спроектированных систем, заключающийся в оценке достижимых маркировок, выявлении мертвых и тупиковых состояний, приводящих к прекращению работы системы, зацикливанию процессов и выявлению сценариев, не участвующих в работе системы. Воспользуемся достоинствами упомянутой структуры и применим его для анализа системы «Протокола передачи данных», но прежде разберём её более подробно.

Сеть Петри – это ориентированный граф, в котором взаимосвязаны между собой вершины двух типов: места и переходы [1]. Данные вершины соединены между собой направленными дугами; вершины одного типа не пересекаются. По сети возможно движение меток (фишек) при срабатывании перехода, которые символизируют всевозможные ресурсы системы. Математически сети Петри представляют в виде  $\mathcal{N} = (P, T, F, m_I)$  [2], где  $P = \{p_1, p_2, \dots, p_n\}$  – множество мест, а  $T = \{t_1, t_2, \dots, t_m\}$  – множество переходов, таких что  $P \cap T = \emptyset$ ,  $F \subseteq P \times T \cup T \times P$  – взаимосвязь вершин, и  $m_I : P \rightarrow \mathbb{N}$  [2]. Анализ систем, смоделированных при помощи сетей Петри, можно реализовать при помощи дерева достижимости [3–10]. Дерево достижимости – есть упорядоченное множество  $(V, E, src, trg)$ , где  $V$  – множество вершин,  $E$  – множество рёбер, а  $src, trg : E \rightarrow V$  отображает назначение для каждого ребра как источник или цель.

Основная проблема при анализе крупных систем – это возможность взрыва пространства состояний (англ. state explosion problem), которая заключается в экспоненциальном росте их количества. Данная проблема может привести к досрочному завершению анализа из-за недостатка нужных объёмов оперативной памяти. Следовательно, без оценки может остаться большое количество состояний дерева достижимости, в которых могут находиться зацикливания и тупиковые маркировки, свидетельствующие о некорректном построении системы. Используя постоянную память машины, можно избежать данной проблемы, но это приведет к значительному росту времени, требуемого на анализ [11]. Таким образом, для анализа сетей Петри воспользуемся способом, который основывается на анализе системы с произвольно выбранного состояния [12].

**Постановка задачи.** Для анализа систем, как предлагалось в работе [12], необходима проверка достижимости выбранного состояния. Данная задача была решена с применением инверсии. В работе [13] описываются простая сеть Петри и способы её преобразования, предложены правила инверсии простой сети, но примеров реализации приведено не было. Поэтому продемонстрируем инверсию системы «Протокола передачи данных», а также доказательство достижимости состояний

при помощи инверсии пространства состояний системы «Протокола передачи данных». Данная система была выбрана как общеизвестный пример [5], который можно проиллюстрировать в рамках представленной работы, тем не менее предлагаемый способ проверки достижимости применим и к более крупным системам.

**Протокол передачи данных.** На примере системы «Протокола передачи данных» (рис. 1) продемонстрируем возможность инверсии сети при помощи правил, предложенных в [13]. Моделируемая система предназначена для передачи информации между отправителем и приёмником, т.е. отправитель может передавать необходимое количество сообщений приёмнику. Предложенная структура системы имеет три основных потока: отправление сообщения от источника к приёмнику, подтверждение получения сообщения от приёмника к источнику и запрос на получение данных от приёмника к источнику. Таким образом, отправитель передаёт пакеты данных получателю и принимает подтверждение об их доставке. В приёмник поступают отправленные сообщения, после чего он отправляет информацию о получении. Данная сеть (см. рис. 1) состоит из 14 мест  $P$ , восьми переходов  $T$  и начальной маркировки  $m_I$ :

$$P = \{F, E, \text{Received}, B, C, S3, D, A, \text{Nextsend}, \text{Send}, S1, S2, \text{Count}, \text{Count1}\};$$

$$T = \{\text{Transmit packet2}, \text{Send packet2}, \text{Receive}, \text{ReceiveAck}, \text{Send Packet}, \text{Receive Packet2}, \text{Transmit2}, \text{TransmitAck}\};$$

$$m_I = (0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1).$$

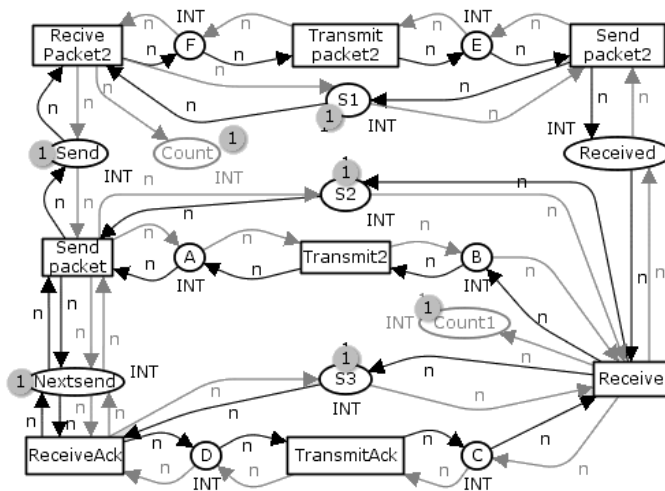


Рис. 1. Сеть «Протокола передачи данных»

провести анализ данной сети и реализовать для неё инверсию. Полученная сеть (см. рис. 1, сеть с направленными стрелками серого цвета без мест серого цвета) имеет ту же структуру, а отсутствие мест  $\text{Count}$  и  $\text{Count1}$  на логику всей системы никак не влияет. Дерево достижимости (рис. 5, слева) для системы (см. рис. 1, сеть с направленными стрелками серого цвета и без мест серого цвета) имеет 18 состояний:  $V = \{m_1, \dots, m_{18}\}$ . Для инверсии сети (см. рис. 1, сеть с направленными стрелками серого цвета, без мест серого цвета) воспользуемся правилами, предложенными в работе [14]. Для данной системы инверсия будет заключаться в простой смене ориентации дуг (алгоритм).

Представленная ранее система (см. рис. 1, сеть с направленными стрелками серого цвета) в работе [5, рис. 1.2] была проанализирована при помощи генерации пространства состояний программного пакета CPN Tools с последующим построением дерева достижимости. Поскольку анализ проводился в системе с местами-счётчиками  $\text{Count}$  и  $\text{Count1}$ , способствующими циклическому анализу, пространство состояний было вычислено и показано частично. А проверка частичного, хоть и повторяющегося пространства состояний не может полностью доказать корректность построения системы. Поэтому было принято решение об удалении этих мест с целью

---

*Алгоритм. Смена ориентации взаимосвязей между вершинами*

---

*while*  $F \subseteq P \times T \cup T \times P'$  // Пока существует цепочка данного вида

*select*  $p \xrightarrow{t} p'$  // Делаем выборку цепочки данного вида

$F := F \setminus \left\{ \left( p \xrightarrow{t} p' \right) \right\}$  // Удаляем выбранные цепочки из всего множества

*for all*  $(t, p')$  *such that*  $p \xrightarrow{t} p'$  *do*  $\text{invert } p' \xrightarrow{t} p$  // Реализуем инверсию

*return*  $(P, T)$  // Возвращаем преобразованные данные

---

Визуально, полученная инвертированная сеть (см. рис. 1, сеть с направленными стрелками чёрного цвета) не отличается от прямой сети Петри (см. рис. 1, сеть с направленными стрелками серого цвета без мест серого цвета). Но чтобы убедиться в этом, нужно представить вместе пространства состояний прямой и обратной сети (рис. 2).

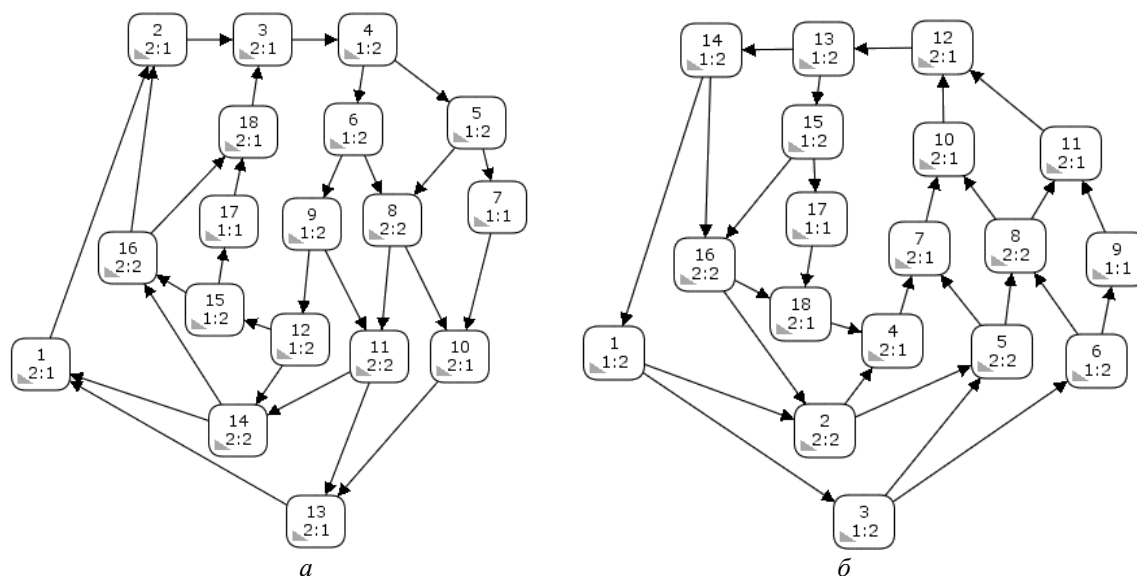


Рис. 2. Дерево достижимости сети Петри «Протокола передачи данных» прямой сети (а) и инвертированной сети (б)

Как видно из рис. 2, структуры деревьев достижимости прямой сети Петри и инвертированной совпадают, что свидетельствует о корректном преобразовании системы. Тем не менее для большей наглядности представим разметку пространств состояний для прямой сети Петри (рис. 3), а для инвертированной сети разметка соответствует рис. 3 с одним условием, что ориентация для всех связей между вершинами будет изменена.

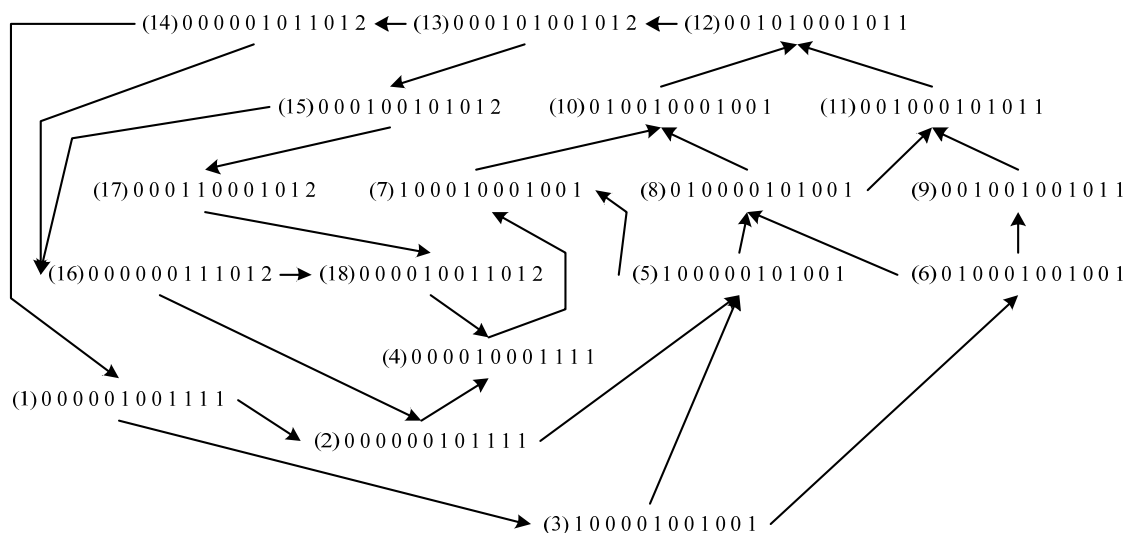


Рис. 3. Разметка пространства состояний прямой сети Петри

Таким образом, после исследования разметки пространства состояний прямой и инвертированной сети системы «Протокола передачи данных» можно сделать вывод о подобии структур деревьев достижимости, что свидетельствует о корректном преобразовании посредством инверсии и возможности использования правил, предложенных в работе [5].

**Инверсия пространства состояний.** При инвертировании сети Петри существует возможность столкнуться с различными проблемами, связанными со сложностью структуры системы. Например, будет недостаточно изменить направление дуг между вершинами, а потребуется преобразование

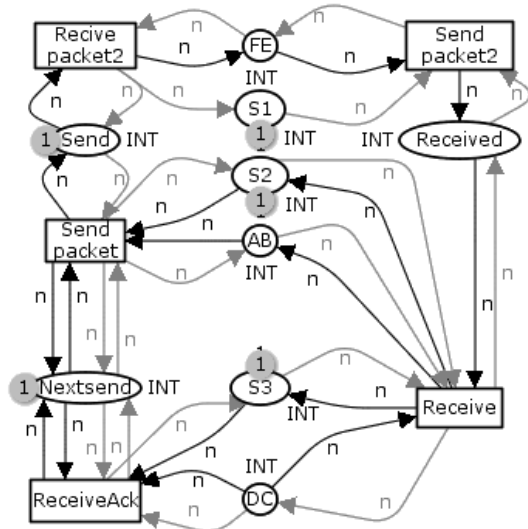


Рис. 4. Преобразованная сеть  
«Протокола передачи данных»

$$P = \{FE, Received, S3, DC, AB, Nextsend, Send, S1, S2\};$$

$$T = \{Send\ packet2, Receive, ReceiveAck, Send\ packet, Receive\ packet2\};$$

$$m_I = (0, 0, 1, 0, 0, 1, 1, 1, 1)$$

и взаимосвязями между вершинами:

$$\begin{aligned} &\{S1, Received\} [Send\ packet2] \{FE\}, \{S2, AB, S3\} [Receive] \{Received, DC\}, \{DC, Nextsend\} [ReceiveAck], \\ &\{S3, Nextsend\}, \{Send, Nextsend\} [Send\ packet] \{S2, AB, Nextsend\}, \{FE\} [Receive\ packet2] \{S1, Send\}. \end{aligned}$$

Количество мест в сети уменьшилось с четырнадцати до девяти, что привело к уменьшению количества состояний в дереве достижимости (рис. 4, сеть с направленными стрелками серого цвета).

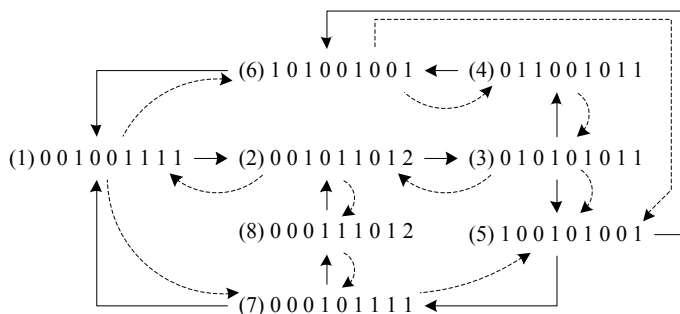


Рис. 5. Разметка пространства состояний  
преобразованной сети Петри

После инвертирования пространства состояний восстановим сеть Петри из конечного дерева достижимости. Восстановим структуру сети, а именно: при изучении первого состояния можно прийти к выводу, что в системе присутствует девять мест (рис. 6). Взаимосвязи (переходы и дуги) между этим типом вершин (местами) нам неизвестны, следовательно, на данном этапе мы получаем сеть, состоящую только из мест. При анализе первого и второго состояния, с соответствующей разметкой для каждого, воссоздание сети продвинется ещё на один шаг: появится один переход, связывающий места 5, 6, 7, 9. Как видно из рис. 5 (дерево достижимости с пунктирными линиями), второе состояние было получено из третьего состояния. Это означает, что для преобразования дерева достижимости в сеть Петри необходимо сравнить эти два состояния. После сравнения разметок был добавлен переход  $R$  и взаимосвязи между ним и пятью местами 2, 3, 4, 5, 9. В третье состояние система может попасть из четвёртого и пятого, рассмотрим последовательно эти два случая. После сравнения состояний четыре и три (рис. 7), добавим переход  $rA$ , связывающий места

структуры сети, что при больших системах весьма затруднительно. В связи с этим реализуем инверсию дерева достижимости на примере протокола передачи данных. Явным преимуществом данного способа является однозначность, которая заключается в чёткой смене ориентаций связей между состояниями, а недостатком – необходимость знать дерево достижимости целиком, что позволит безошибочно преобразовать пространство состояний.

Реализуем данное преобразование на примере протокола передачи данных (см. рис. 1, сеть с направленными стрелками серого цвета, без мест серого цвета). Для упрощения исходного дерева достижимости были произведены изменения, которые заключались в объединении мест  $F$  и  $E$ ,  $A$  и  $B$ ,  $D$  и  $C$  и удалении переходов между ними (рис. 4, сеть с направленными стрелками серого цвета). После преобразования система содержит девять мест  $P$ , пять переходов  $T$  и начальную маркировку  $m_I$ :

Разметка для дерева достижимости показана на рис. 5, (сплошные линии). Как предлагалось ранее, реализуем инверсию не структуры системы, т.е. сети Петри, а непосредственно дерева достижимости. Преобразование будет заключаться в изменении направления связей между вершинами состояний. Инвертированное пространство состояний  $(V_i, E_i, src, trg)$ , повторяет структуру исходного дерева, а взаимосвязи меняют своё направление (см. рис. 5, пунктирные линии).

3, 4, 6. А при сравнении третьего и пятого состояния добавим в сеть переход  $SP2$  с дугами между местами 1, 2, 8. Как видно из рис. 5 (дерево достижимости с пунктирными линиями), состояние шесть получено из первого, а его потомками являются четвёртое и пятое состояние, следовательно, при анализе взаимосвязей между данным состоянием с родителем и потомками в сети не появится новых переходов и дуг. При сравнении седьмого состояния и пятого добавляем в сеть переход  $RP2$ , связывающий места 1, 7, 8. После добавления перехода  $RP2$ , его входных и выходных дуг, сеть полностью восстановлена, а состояние восемь (рис. 8) может служить для проверки корректности построенной сети из дерева достижимости.

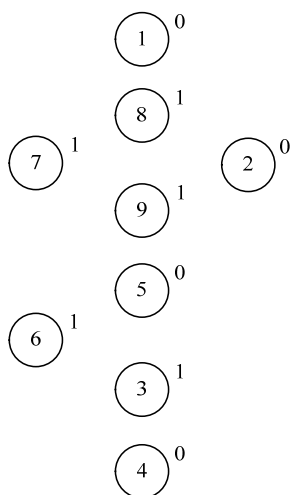


Рис. 6. Восстановление сети Петри: состояния №1

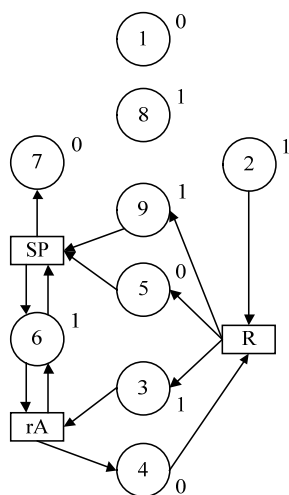


Рис. 7. Восстановление сети Петри: состояния №4

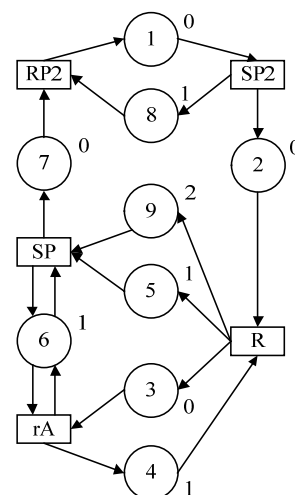


Рис. 8. Восстановление сети Петри: состояния №8

Таким образом, после восстановления сети из инвертированного дерева достижимости получаем инвертированную сеть (см. рис. 4, сеть с направленными стрелками чёрного цвета) системы. Если сравнить две сети, то можно сказать, что для инверсии прямой сети непосредственно, достаточно изменить ориентацию дуг, что много проще при неизвестном дереве достижимости. Но применив инверсию к дереву достижимости, мы использовали значительное преимущество – однозначность трансформации пространства состояний, что может быть весьма полезно для систем со сложной структурой сети. А восстановление сети из пространства состояний занимает определённое количество времени, но трудности не вызывает. Следовательно, проектировщик может сделать выбор между сложностью преобразования сети и временными ресурсами на воссоздание системы из инвертированного дерева достижимости.

**Заключение.** Результатом исследования стала реализация инверсии не сети, а именно пространства состояний, для упрощения которого сеть Петри была преобразована через соединение двух мест и перехода между ними в одно место. Основным достоинством данного метода является однозначность преобразования, которая заключается в смене ориентации взаимосвязей между вершинами, после чего реализовывалось восстановление инвертированной сети из преобразованного дерева достижимости. Стоит отметить, что значительным преимуществом является однозначность трансформации пространства состояний, это может быть полезно для систем со сложной структурой сети. Восстановление сети из пространства состояний занимает определённое количество времени, но трудности не вызывает. Таким образом, приходится делать выбор между сложностью преобразования сети и временными ресурсами на восстановление системы из инвертированного дерева достижимости.

Для проверки правил, предложенных в работе [13], была выбрана сеть Петри протокола передачи данных. На её примере реализовывалась инверсия, заключающаяся в изменении ориентации дуг между вершинами сети. Для проверки корректности преобразования были построены разметки пространства состояния прямой сети и инвертированной. Сравнение полученных результатов показало, что сеть была инвертирована корректно, что свидетельствует о работоспособности предложенного способа и правил.

В последующих работах планируется описать правила для инверсии цветных сетей и иерархических, а также их возможных комбинаций.

#### *Литература*

1. Питерсон Дж. Теория сетей Петри и моделирование. – М.: Мир, 1984. – 264 с.
2. Westergaard M. Behavioral verification and visualization of formal models of concurrent systems: PhD dissertation. – Aarhus: University of Aarhus, 2007. – 183 с.
3. Марков А.В. Моделирование процесса поиска пути в лабиринте при помощи сетей Петри // Сборник научных трудов НГТУ. – 2010. – № 4(62). – С. 133–141.
4. 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.
5. Романников Д.О. Разработка программного обеспечения с применением UML диаграмм и сетей Петри для систем управления локальным оборудованием: дис. ... канд. техн. наук. – Новосибирск: Изд-во НГТУ, 2012. – 195 с.
6. Воевода А.А. Понятие рекурсии в сетях Петри: факториал числа, числа Фибоначчи / А.А. Воевода, А.В. Марков // Сборник научных трудов НГТУ. – 2013. – № 1(71). – С. 72–77.
7. Марков А.В. Анализ сетей Петри при помощи деревьев достижимости / А.В. Марков, А.А. Воевода // Сборник научных трудов НГТУ. – 2013. – №1(71). – С. 78–95.
8. Марков А.В. Разработка программного обеспечения при совместном использовании UML-диаграмм и сетей Петри (обзор) // Сборник научных трудов НГТУ. – 2013. – № 1(71). – С. 96–131.
9. 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.
10. Марков А.В. Матричное представление сетей Петри // Сборник научных трудов НГТУ. – 2013. – № 2(71). – С. 61–67.
11. 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.
12. Holzmann G.J. An analysis of bitstate hashing // Symposium on Protocol Specification. – 1995. – № 15. – P. 301–314.
13. Марков А.В. Инверсия простой ординарной сети Петри / А.В. Марков, А.А. Воевода // Научный вестник НГТУ. – 2013. – № 4(53). – С. 146–150.

---

#### **Марков Александр Владимирович**

Аспирант каф. автоматизации Новосибирского гос. технического университета (НГТУ)

Тел.: 8-952-926-14-04

Эл. почта: muviton3@gmail.com

#### **Воевода Александр Александрович**

Д-р техн. наук, профессор каф. автоматизации НГТУ

Тел.: 8 (383) 344-49-98

Эл. почта: muviton3@gmail.com

Markov A.V., Voevoda A.A.

#### **Checking the accessibility of marking Petri nets by inverting state trees for data transmission protocol**

The final stage of software design needs analysis, capable to detect possible flaws in the system. One of such a method is to study the system randomly at any selected state in which the model can be functioning. To use this approach, it is obligatory to prove the accessibility of a selected state, which will consist of state space inversion followed by reduction of the network. This operation is straightforward and can be useful for models with complex structures. The inversion is demonstrated by data transmission protocol using the rules proposed in the paper to transform the simple Petri net.

**Keywords:** Petri net analysis of the state space, accessibility, inversion, data transfer protocol.