Модел. и анализ информ. систем. Т. **20**, № **4** (2013) 23–40

УДК 519.71+004.021

О разрешимости бездефектности для сетей потоков работ с неограниченным ресурсом

Башкин В.А., Ломазова И.А.[1]

Ярославский государственный университет им. П. Г. Демидова
150000 Россия, г. Ярославль, ул. Советская, 14

Национальный исследовательский университет "Высшая школа экономики"
101000 Россия, г. Москва, ул. Мясницкая, д. 20,
Институт программных систем РАН
152021 Россия, Ярославская обл., Переславский р-н, с. Веськово, ул. Петра I, д. 4 «а»

e-mail: v_bashkin@mail.ru, i_lomazova@mail.ru

получена 4 сентября 2013

Ключевые слова: сети Петри, потоки работ, ресурс, бездефектность, верификация, моделирование

Рассматривается моделирование схем потоков работ (workflow) при помощи сетей Петри. Определяется класс сетей потоков работ с ресурсами (RWF-сетей) — обычных workflow-сетей, в которых дополнительно добавлено множество ресурсных позиций, содержащих какую-то начальную разметку (начальный ресурс). Ресурсы могут уничтожаться и производиться при срабатываниях переходов. Мы не накладываем ограничений ни на промежуточные, ни на финальные ресурсные разметки, поэтому сеть может порождать бесконечное множество различных достижимых состояний.

RWF-сеть с данной начальной ресурсной разметкой называется бездефектной, если, во-первых, она всегда корректно завершает свою работу, и, во-вторых, любое увеличение начального ресурса не нарушает свойства корректного завершения. Неразмеченная RWF-сеть бездефектна, если она бездефектна при некоторой начальной ресурсной разметке. В данной работе доказана разрешимость обоих вариантов бездефектности для важного подкласса RWF-сетей — сетей с одномерным ресурсом (одной ресурсной позицией). Также представлен алгоритм вычисления наименьшего бездефектного ресурса.

1. Введение

В последние годы всё более активно развивается ещё одна важная область применения сетей Петри [19] — моделирование потоков работ (workflow). Потоки работ используются для формализации управления всевозможными технологическими процессами, бизнес-процессами, web-сервисами, распределенными вычислениями и т.д.

[1]Работа поддержана РФФИ (проекты 11-07-00549, 12-01-31508, 11-01-00737) и программой "Научные и научно-педагогические кадры инновационной России" (проект 14.B37.21.0392).

В потоках работ возможны циклы, также возможно распараллеливание и синхронизация. Однако есть и ограничения, в частности, существуют одно начальное и одно конечное состояния, запрещены тупики.

Для моделирования процессов потоков работ используется специальный подкласс сетей Петри — так называемые WF-сети [4, 1].

Одной из основных проблем при разработке workflow является обеспечение их правильной организованности — отсутствия тупиков, избыточных переходов, необоснованного использования ресурсов и т.п. Применяется формальный критерий *бездефектности* — говорят, что исполнение процесса завершается корректно, если от любой разметки, достижимой от начальной (одна фишка в начальной позиции), достижима финальная (одна фишка в финальной позиции), и при этом в сети по завершении работы не остаётся лишних управляющих фишек.

Бездефектность WF-сетей разрешима [4, 5]. Более того, известно несколько разрешимых модификаций классического понятия бездефектности, например, k-бездефектность [14], структурная бездефектность [21], бездефектность вложенных моделей [16] и структурированных сетей [12].

При разработке workflow-процессов большое внимание уделяется управлению ресурсами. Ресурсы в данном случае понимаются в обобщённом смысле — это могут быть исполнители (люди или устройства), сырьё, финансы и т.д. Это очень мощный инструмент моделирования, позволяющий отслеживать такие свойства системы, как избыточность, недостаточность, взаимная блокировка ресурсов и агентов [2], а также возможность эквивалентной замены одних ресурсов другими без нарушения наблюдаемого поведения процесса в целом [10, 9].

Для того, чтобы лучше учесть ресурсную составляющую систем, базовый формализм WF-сетей многими авторами был расширен до различных вариантов “сетей с ресурсами”, что повлекло необходимость соответствующих модификаций понятия бездефектности и, как правило, усложнило проверку этого свойства.

В [7, 8] был исследован специальный класс сетей с разделяемыми ресурсами (WFR-сетей), для которого также была установлена разрешимость бездефектности. В [15, 20] был введён более общий класс — так называемые ресурсно-ограниченные сети (Resource-Constrained Workflow Nets, RCWF-сети). В обоих случаях авторы накладывают на ресурсы два ограничения. Во-первых, количество доступных ресурсов на момент окончания работы процесса должно совпадать с начальным количеством. Во-вторых, при любой достижимой разметке количество доступных ресурсов не может превышать начальное количество.

В [15] было доказано, что для RCWF-сетей с одномерным ресурсом бездефектность разрешима за полиномиальное время. В [20] доказано, что бездефектность разрешима и в общем случае (сведением к задаче о домашней разметке).

В перечисленных статьях рассматриваются ресурсы, которые находятся в системе постоянно и в фиксированном количестве. Они не уничтожаются и не создаются, а всего лишь блокируются и освобождаются при срабатываниях переходов сети. Таким образом, множества состояний RCWF- и WFR-сетей конечны.

В работе [11] нами был представлен более общий класс сетей с произвольными трансформациями ресурсов, которые требуются, например, в случае открытых и/или адаптивных workflow-систем. Был определён формализм, названный ресурсными WF-сетями (RWF-сетями), в котором не накладывается никаких ограничений на работу с ресурсными позициями. Даже бездефектные RWF-сети могут обладать бесконечными множествами достижимых состояний, поэтому известные методы анализа бездефектности для них неприменимы.

Для ресурсов произвольной размерности вопрос разрешимости бездефектности остаётся открытым. В данной работе представлены алгоритмы проверки бездефектности и определения минимального необходимого ресурса для частного случая — одномерного ресурса (сетей с одной ресурсной позицией). Одномерный ресурс — интересное сужение RWF-сетей, имеющее достаточно много практических приложений (например, моделирование дискретного времени, объёма памяти, доступных финансов и т.п.). Заметим, что даже одномерный ресурс может накапливаться неограниченно, поэтому данный случай не может быть сведён к уже известным классам ресурсных сетей с конечными множествами состояний (RCWF- и WFR-сетям).

Используя теоретико-графовые свойства управляющего автомата одномерной RWF-сети, мы доказываем разрешимость бездефектности как для размеченной, так и для неразмеченной сети. Также представлен алгоритм вычисления наименьшего бездефектного ресурса для данной одномерной сети.

Статья организована следующим образом. Во втором разделе мы напоминаем основные определения и обозначения, касающиеся мультимножеств, сетей Петри и сетей потоков работ (WF-сетей). В разделе 3 определяются сети потоков работ с ресурсами (RWF-сети) и вводится понятие бездефектного ресурса. В разделе 4 рассматриваются RWF-сети с одномерным ресурсом. Доказывается разрешимость бездефектности, приводятся алгоритмы проверки бездефектности для размеченных и неразмеченных сетей, а также алгоритм вычисления наименьшего бездефектного ресурса. Раздел 5 содержит некоторые выводы.

2. Предварительные сведения

Пусть S — конечное множество. *Мультимножеством* M над множеством S называется отображение $M : S \to Nat$, где Nat — множество неотрицательных целых чисел. Обозначим через $\mathcal{M}(S)$ множество всех конечных мультимножеств над S.

Операции и отношения теории множеств естественно расширяются на конечные мультимножества. Пусть $M_1, M_2, M_3 \in \mathcal{M}(S)$. Полагаем:

- $M_1 \subseteq M_2 \quad \Leftrightarrow \quad \forall s \in S : M_1(s) \leq M_2(s)$;
- $M_1 = M_2 + M_3 \quad \Leftrightarrow \quad \forall s \in S : M_1(s) = M_1(s) + M_2(s)$;
- $M_1 = M_2 \cup M_3 \quad \Leftrightarrow \quad \forall s \in S : M_1(s) = max\{M_1(s), M_2(s)\}$.

Сетью Петри называется набор $N = (P, T, F)$, где P — конечное множество позиций; T — конечное множество переходов, $P \cap T = \varnothing$; $F : (P \times T) \cup (T \times P) \to Nat$ — функция инцидентности (мультимножество дуг).

Разметкой (состоянием) сети N называется функция вида $M : P \to Nat$, сопоставляющая каждой позиции сети некоторое натуральное число (или ноль). Разметка может рассматриваться как мультимножество над множеством позиций сети, то есть элемент множества $\mathcal{M}(P)$.

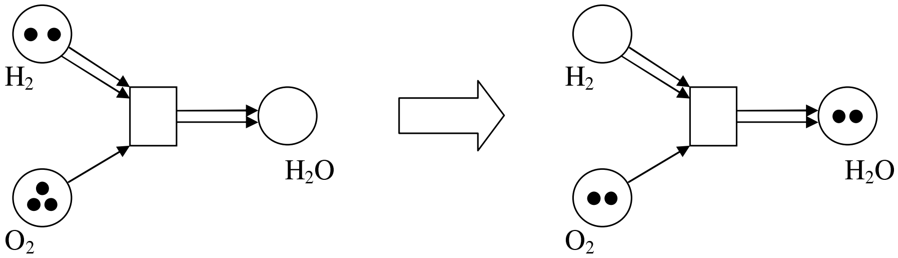

Рис. 1. Сеть Петри, моделирующая химическую реакцию

Размеченной сетью Петри называется пара (N, M_0), где $N = (P, T, F)$ — сеть Петри, $M_0 \in \mathcal{M}(P)$ — начальная разметка (количество ресурса в наличии при запуске сети).

Графически сеть Петри изображается как двудольный ориентированный граф. Вершины-позиции изображаются кружками и характеризуют локальные состояния сети, вершины-переходы изображаются прямоугольниками и соответствуют действиям. Дуги соответствуют элементам F. Позиции могут содержать маркеры (фишки), изображаемые черными точками. При разметке M в каждую позицию p помещается $M(p)$ фишек.

Для перехода $t \in T$ через ${}^\bullet t$ и $t^\bullet$ обозначим мультимножества его входных и выходных позиций, такие, что

$$\forall p \in P \quad {}^\bullet t(p) =_{def} F(p,t),\ t^\bullet(p) =_{def} F(t,p).$$

Переход $t \in T$ *активен (готов к срабатыванию)* при разметке M, если ${}^\bullet t \subseteq M$ (все входные позиции содержат достаточное количество фишек). Готовый к срабатыванию переход t может *сработать*, порождая новую разметку $M' =_{def} M - {}^\bullet t + t^\bullet$ (используется обозначение $M \xrightarrow{t} M'$). Множество всех разметок, достижимых от начальной разметки M за одно или несколько срабатываний переходов, обозначается как $\mathcal{R}(N, M)$.

Пример срабатывания переходов в обыкновенной сети Петри приведен на Рис. 1.

Для моделирования потоков работ (workflow) используется специальный подкласс сетей Петри — так называемые WF-сети [1].

Определение 1. *Пусть $N = (P, T, F)$ — обыкновенная сеть Петри. Сеть N называется* WF-сетью *(сетью потока работ), если*

1. *в множестве P имеются две специальные позиции i и o, такие, что ${}^\bullet i = o^\bullet = \emptyset$;*

2. *любой элемент множества $P \cup T$ лежит на пути из i в o.*

Позиция i называется *начальной*, а позиция o — *финальной* позицией сети N. Начальная разметка сети потока работ состоит из одной фишки в позиции i.

Правильное завершение процесса, моделируемого сетью, гарантируется выполнением следующего свойства:

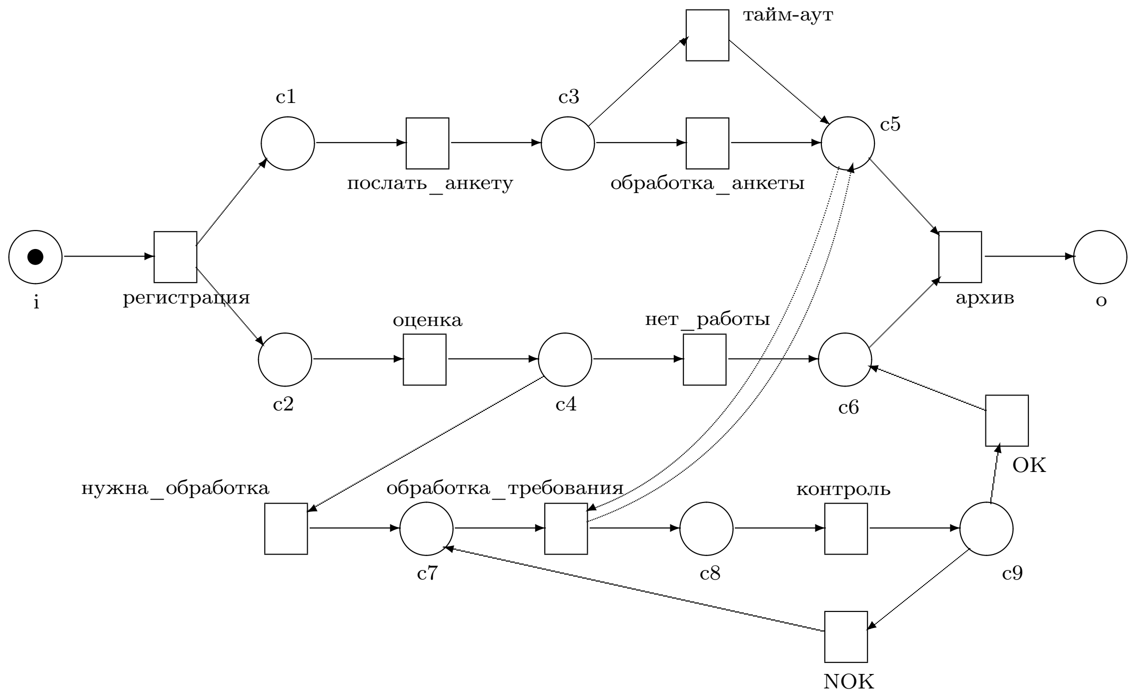

Рис. 2. Модель бизнес-процесса (workflow) обработки страховых требований

Определение 2. *WF-сеть N называется* бездефектной*, если для любой достижимой разметки $M \in R(N, i)$ выполняется*

1. $o \in R(N, M)$;
2. $o + M' \in R(N, M) \;\Rightarrow\; M' = \emptyset$.

Другими словами, из любого достижимого состояния бездефектной сети достижимо финальное состояние, при этом в финальном состоянии не может остаться никаких “лишних” фишек. Свойство бездефектности может быть эффективно проверено [4].

На рисунке 2 изображена сеть Петри, моделирующая процесс обработки страховых требований (пример взят из книги [1]). Прежде всего, требование регистрируется (задача *регистрация*), затем параллельно выполняются две задачи: клиенту посылается анкета (задача *послать_анкету*) и производится оценка требования (задача *оценка*). Если анкета возвращается в течение двух недель, то выполняется задача *обработка_анкеты*. Если в течение двух недель анкета не возвращена, то результат анкетирования игнорируется (задача *тайм-аут*). На основании результата оценки требование либо рассматривается, либо нет. Реальная работа с требованием (задача *обработка_требования*) откладывается до того момента, когда будет выполнено условие c5, т.е. либо анкета обработана, либо истекло время ее ожидания. Обработка требования контролируется с помощью задачи *контроль*. В конце концов, выполняется задача *архив*.

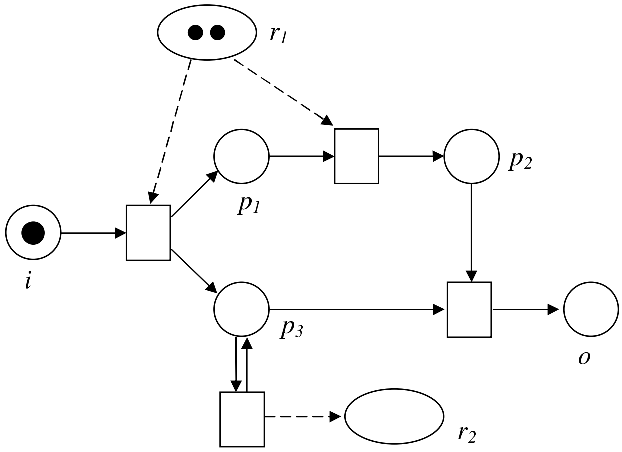

Рис. 3. WF-сеть с ресурсами

3. WF-сети с ресурсами

В сетях с ресурсами мы дополнительно вводим множество ресурсных позиций.

Определение 3. Сетью потоков работ с ресурсами *(RWF-сетью) называется набор* $N = (P_c, P_r, T, F_c, F_r)$*, где*

- $N_c = (P_c, T, F_c)$ *— обыкновенная WF-сеть (называемая* управляющей подсетью *сети* N*, при этом элементы множеств* P_c *и* F_c *называются управляющими позициями и дугами соответственно);*
- P_r *— конечное множество ресурсных позиций,* $P_c \cap P_r = \emptyset$*;*
- $F_r \ : \ (P_r \times T) \cup (T \times P_r) \to Nat$ *— конечное множество ресурсных дуг.*

Заметим, что из определений следует, в частности, что $\forall t \in T \ \exists p \in P_c \ :$ $F_c(p, t) + F_c(t, p) > 0$, то есть каждый переход инцидентен какой-нибудь управляющей позиции — этим гарантируется невозможность "неконтролируемых" модификаций ресурсов.

Разметка RWF-сети распадается на две составляющие — управляющую и ресурсную части. Мультимножество $c + r$, в котором $c \in \mathcal{M}(P_c)$ и $r \in \mathcal{M}(P_r)$, мы будем обозначать как $(c|r)$.

Определение 4. *Для сети* N ресурсом *называется мультимножество над* P_r*.* Управляемым ресурсом *называется мультимножество над* $P_c \cup P_r$*.*

На Рис. 3 приведен пример RWF-сети с ресурсными позициями r_1 и r_2. Ресурсные позиции обозначаются овалами, ресурсные дуги — пунктирными стрелками.

Введём определение бездефектности для сетей с ресурсами.

Определение 5. *Размеченная RWF-сеть* $(N, c|r)$ *называется* бездефектной, *если* $\forall s \in \mathcal{M}(P_r)$, $\forall M \in \mathcal{R}(N, c|r + s)$ *выполняется:*

1. $\exists s' \in \mathcal{M}(P_r) \ : \ o|s' \in \mathcal{R}(N, M)$;

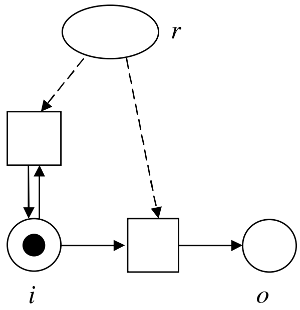

Рис. 4. RWF-сеть с бездефектной управляющей подсетью, для которой не существует бездефектных ресурсов

2. $c'|r' \in \mathcal{R}(N, M) \;\Rightarrow\; c' = o \;\vee\; c' \cap o = \emptyset$.

Таким образом, во-первых, процесс может правильно завершиться при любом развитии событий, и, дополнительно, увеличение начального ресурса не нарушает свойства правильной завершаемости.

Заметим, что наше определение существенно отличается от определения бездефектных RCWF-сетей [15]. Мы не запрещаем создание и уничтожение ресурсов — они могут производиться и непосредственно в ходе исполнения процесса. Из этого, в частности, следует возможная неограниченность бездефектных RWF-сетей.

Утверждение 1. *[11] Если размеченная RWF-сеть $(N, i|r)$ бездефектна, то размеченная WF-сеть (N_c, i) также бездефектна.*

Доказательство. Предположим противное. Пусть $(N, i|r)$ — бездефектная RWF-сеть, но при этом WF-сеть (N_c, i) не бездефектна. Тогда найдётся разметка $c \in \mathcal{R}(N_c, i)$, такая, что либо разметка o недостижима от c, либо $o \in c$ и $c \neq \{o\}$.

Поскольку в управляющей подсети разметка c достижима от начальной разметки i после некоторой конечной последовательности срабатываний переходов, мы всегда можем выбрать начальный ресурс s настолько большим, чтобы та же последовательность могла сработать в сети с ресурсами от разметки $(N, i|r+s)$, достигнув того же управляющего состояния c.

Если в управляющей подсети финальное состояние не было достижимо от c, то добавление ресурсных позиций никак не могло сделать его достижимым в сети с ресурсами, то есть для $(N, i|r+s)$ — что противоречит бездефектности $(N, i|r)$. Если же $o \in c$ и $c \neq \{o\}$, то мы также получаем противоречие с бездефектностью $(N, i|r)$, поскольку управляющее состояние c достижимо в $(N, i|r + s)$. $\square$

Обратное неверно: существуют не бездефектные RWF-сети, управляющие подсети которых бездефектны. Пример такой сети приведён на Рис. 4.

Пусть N — RWF-сеть. Через $C(N)$ обозначим *множество всех достижимых в N_c управляющих разметок*, т. е. $C(N) = \mathcal{R}(N_c, i)$.

Утверждение 2. *[11] Если размеченная RWF-сеть $(N, i|r)$ бездефектна, то*

1. *для любой достижимой управляющей разметки $c \in C(N)$ найдётся ресурс r', такой, что $(N, c|r')$ бездефектна;*

2. *для любых $c_1, c_2 \in C(N)$ выполняется $c_1 \not\subset c_2$ и $c_2 \not\subset c_1$.*

Доказательство. (1) Как и в случае доказательства Утв. 1, мы можем взять достаточно большой начальный ресурс $r + s$.

(2) Предположим противное. Пусть для некоторых $c_1, c_2 \in C(N)$ выполняется $c_2 = c_1 + c'$, где $c' \neq \emptyset$. Из Утв. 2(1) следует, что найдутся ресурсы r_1 и r_2, такие, что RWF-сети $(N, c_1|r_1)$ и $(N, c_2|r_2)$ бездефектны. Тогда сети $(N, c_1|r_1+r_2)$ и $(N, c_2|r_1+r_2)$ также бездефектны. Следовательно, финальная разметка $o|r'$ достижима от разметки $c_1|r_1 + r_2$, то есть разметка $o + c'|r'$ достижима от бóльшей разметки $c_2|r_1 + r_2$ — противоречие со свойством бездефектности RWF-сети $(N, c_2|r_1 + r_2)$. □

Далее мы будем называть RWF-сеть N бездефектной (без указания конкретного ресурса), если существует такой ресурс r, при котором размеченная RWF-сеть $(N, i|r)$ бездефектна.

Из второй части Утв. 2 и известной леммы Диксона [13] получим, что

Следствие 1. *Для любой бездефектной RWF-сети N множество $C(N)$ достижимых управляющих состояний конечно.*

Замечание 1. *Поскольку управляющая подсеть бездефектной RWF-сети N ограничена, множество $C(N)$ может быть эффективно построено (например, при помощи покрывающего дерева [3]).*

Определение 6. *Пусть N — RWF-сеть, $c \in C(N)$. Определим:*

1. $res(c) =_{def} \{r \in \mathcal{M}(P_r) \mid (N, c|r)$ *бездефектна*$\}$ *— множество всех бездефектных ресурсов для c;*

2. $mres(c) =_{def} \{r \in res(c) \mid \nexists r' \in res(c) : r' \subset r\}$ *— множество всех минимальных бездефектных ресурсов для c.*

Из леммы Диксона немедленно получаем:

Утверждение 3. *[11] Для любой бездефектной RWF-сети N и для любой её управляющей разметки $c \in C(N)$ множество $mres(c)$ конечно.*

Вопросы о вычислимости $mres(c)$ и разрешимости бездефектности для RWF-сетей пока остаются открытыми. В следующей главе мы дадим положительные ответы на оба эти вопроса для случая одномерного ресурса.

4. RWF-сети с одномерным ресурсом

Пусть $N = (P_c, P_r, T, F_c, F_r)$ — RWF-сеть, в которой $P_r = \{p_r\}$ (то есть ресурсная позиция всего одна). Состояние сети мы будем обозначать как $(c|r)$, где $c \in C(N)$ — управляющее состояние, $r \in Nat$ — текущее количество ресурса (значение счётчика). Такие сети мы будем называть RWF-сетями с одномерным ресурсом или просто одномерными RWF-сетями. Пример сети приведён на Рис. 5.

Поскольку нас интересуют вопросы бездефектности ресурсов, в дальнейшем мы предполагаем, что в рассматриваемых сетях управляющая подсеть уже бездефектна (в противном случае в силу Утв. 1 сама сеть также автоматически была бы не бездефектной) и, следовательно, ограничена (Следствие 1).

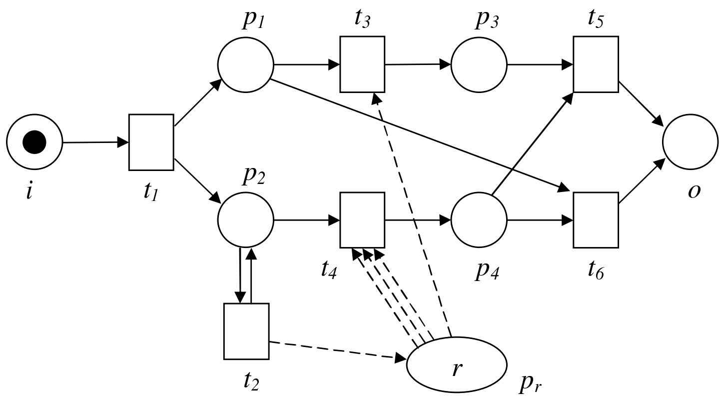

Рис. 5. Одномерная RWF-сеть N_1

4.1. Управляющий автомат

Ограниченная управляющая подсеть RWF-сети может быть представлена в виде конечного автомата (системы переходов). Состояниями автомата являются все управляющие состояния исходной сети (элементы $C(N)$), переходами — её переходы. Таким образом, диаграмма переходов представляет собой ориентированный граф с двумя выделенными вершинами — начальным состоянием i (без входящих дуг) и финальным состоянием o (без исходящих дуг). Заметим также, что из определения WF-сети следует, что в этом графе все прочие вершины лежат на путях из i в o.

Каждый переход t мы дополнительно пометим целым числом $\mathit{Eff}(t)$, показывающим "ресурсный эффект" данного перехода, то есть его воздействие на ресурс (увеличение или уменьшение значения счётчика):

$$\mathit{Eff}(t) =_{\text{def}} \begin{cases} -F_r(p_r, t) & \text{при } F_r(p_r, t) > 0; \\ F_r(t, p_r) & \text{при } F_r(t, p_r) > 0. \end{cases}$$

Заметим, что для простоты мы не рассматриваем переходы-петли (для которых одновременно выполняется $F_r(p_r, t) > 0$ и $F_r(t, p_r) > 0$) — они могут моделироваться цепочками из двух последовательных переходов.

Ресурс перехода определяется как:

$$\mathit{Supp}(t) =_{\text{def}} \begin{cases} 0, & \mathit{Eff}(t) \geq 0; \\ |\mathit{Eff}(t)|, & \mathit{Eff}(t) < 0. \end{cases}$$

Рассмотрим пример на Рис. 6. Здесь показан управляющий автомат одномерной RWF-сети, изображённой на Рис. 5. Состояния обозначены восьмиугольниками, помеченными соответствующими управляющими разметками сети. Переходы помечены соответствующими именами и ресурсными эффектами. Например, $t_2[+1]$ означает, что перед нами переход t_2 исходной сети, который увеличивает значение счётчика на единицу.

Итак, RWF-сеть с одномерным ресурсом представима в виде управляющего автомата с одним счётчиком без проверки на ноль. Этот класс систем также называют односчётчиковыми сетями (например, [6]) или системами сложения одномерных

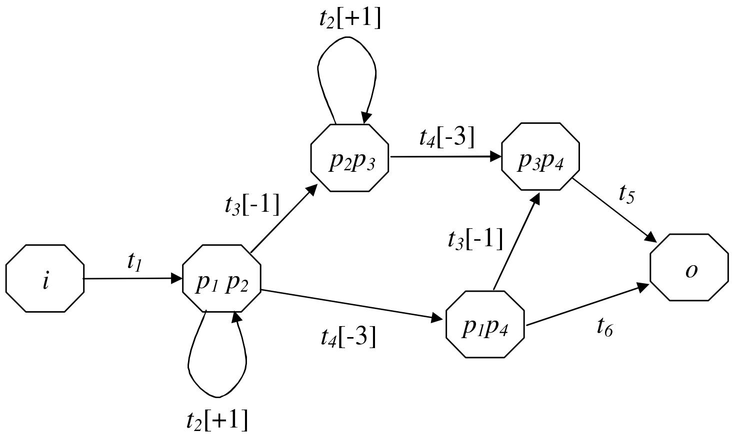

Рис. 6. Управляющий автомат для N_1

векторов с состояниями (1-dim Vector Addition System with States — VASS [17]). Однако заметим, что из-за особенностей потоков работ дополнительно накладывается ограничение: в системе есть ровно по одному начальному и финальному состоянию.

Диаграмма переходов односчетчиковой сети представляет собой взвешенный ориентированный граф. Напомним несколько основных понятий теории графов.

Маршрут в графе — это чередующаяся последовательность вершин и рёбер, начинающаяся и заканчивающаяся вершинами, в которой вершины, идущие до и после дуги, являются её началом и концом соответственно.

Далее мы рассматриваем только непустые маршруты, содержащие по крайней мере одну дугу.

Маршрут *замкнут*, если его первая и последняя вершины совпадают.

Цепью (ориентированной цепью) называется маршрут без повторяющихся дуг (вершины могут повторяться).

Простая цепь — это цепь, в которой вершины не повторяются.

Циклом называется замкнутая цепь.

Простой цикл — это цикл, в котором вершины не повторяются (кроме первой и последней).

Эффект и ресурс маршрута определяются индуктивно. Пусть t — переход, σ – маршрут, такой, что конец перехода t является началом первого перехода из σ. Обозначив как $t\sigma$ маршрут, полученный сцеплением t и σ, мы получим:

$$Eff(t\sigma) =_{\text{def}} Eff(t) + Eff(\sigma); \quad Supp(t\sigma) =_{\text{def}} Supp(t) + (Supp(\sigma) \ominus Eff(t)).$$

Здесь $\ominus$ обозначает усеченное вычитание: для $x, y \in Nat$ положим $x \ominus y =_{\text{def}} \max\{0, x - y\}$.

Положительным (*отрицательным*) называется маршрут с положительным (соответственно, отрицательным) эффектом. Очевидно, что эффект цикла не зависит от выбора начального/конечного узла.

Узел q называется *положительным генератором*, если существует положительный маршрут от q до q (образующий положительный цикл) с нулевым ресурсом.

Лемма 1. *Любой положительный цикл содержит хотя бы один генератор.*

Доказательство. Индукция по длине цикла. Заметим также, что без ограничения общности мы можем рассматривать только циклы четной длины с чередующимися положительными и отрицательными дугами. □

Узел q называется *отрицательным генератором*, если существует отрицательный маршрут θ от q до q (образующий отрицательный цикл), такой, что $Supp(\theta) = -Eff(\theta)$.

Лемма 2. *Любой отрицательный цикл содержит хотя бы один генератор.*

Доказательство. Аналогично предыдущей лемме. □

4.2. Разрешимость бездефектности для размеченных сетей

Пусть $(N, i|r_0)$ — одномерная RWF-сеть с заданной начальной разметкой. Допуская вольность обозначений, будем тем же символом N обозначать её управляющий автомат, представленный в виде односчётчиковой сети.

Существует два типа нежелательного поведения сети, приводящего к нарушению бездефектности — тупик (deadlock) и динамический тупик (livelock), называемый также динамической блокировкой. Для краткости далее мы везде будем использовать термины "тупик" и "блокировка".

Определение 7. *Состояние сети* $(c|r) \in C(N) \times Nat$ *называется* тупиком*, если* $c \neq o$ *и не существует перехода* $t \in T$*, такого, что* $(c|r) \xrightarrow{t} (c'|r')$.

Конечное подмножество $L \subset C(N) \times Nat$ *множества состояний сети называется* блокировкой*, если*

1. $|L| > 1$;
2. *для любых* $(c|r), (c'|r') \in L$ *существует конечная последовательность переходов* $\sigma \in T^*$*, такая, что* $(c|r) \xrightarrow{\sigma} (c'|r')$;
3. *для любых* $(c|r) \in L$ *и* $t \in T$*, таких, что* $(c|r) \xrightarrow{t} (c''|r'')$*, выполнено* $(c''|r'') \in L$.

Блокирующим состоянием *называется состояние, принадлежащее блокировке.*

Заметим, что по определению $(o|r) \notin L$ для любого r.

Утверждение 4. *Если* $(c|r)$ *— тупик, то для любого* $t \in T$*, такого, что* $c \xrightarrow{t} c'$*, выполняется* $Supp(t) > r$.

Доказательство. Очевидно. □

Заметим, что из Утв. 4 следует, что $Eff(t) < 0$, то есть мы можем его переформулировать:

Следствие 2. *Если* $(c|r)$ *— тупик, то:*

1. $\forall t \in T$*, такого, что* $c \xrightarrow{t} c'$ *для некоторого* c'*, выполняется* $Eff(t) < 0$;
2. $r < min\{|Eff(t)| \ : \ c \xrightarrow{t} c' \ \text{ для некоторого } \ c'\}$.

Таким образом, тупики могут возникать (1) только в управляющих состояниях, у которых все исходящие переходы — отрицательные; (2) только для конечного числа различных значений счётчика — когда ресурса недостаточно ни для одного из исходящих переходов.

Утверждение 5. *Множество тупиков конечно.*

Доказательство. Множество “потенциальных тупиков” — управляющих состояний с исходящими отрицательными переходами — конечно. Для данного “потенциального тупика” множество “опасных” значений счётчика (натуральных чисел, меньших, чем наименьший из ресурсов, требуемых для срабатывания исходящего перехода) также конечно. □

Следовательно, все тупики могут быть обнаружены перебором управляющих состояний (число которых конечно).

Теперь рассмотрим блокировки.

Утверждение 6. *Если $L \subset C(N) \times Nat$ — блокировка, то найдутся состояние $(c|r) \in L$ и отрицательный переход $t \in T$, такие, что $c \xrightarrow{t} c'$ для некоторого c', и при этом $Eff(t) < -r$.*

Доказательство. Очевидно, поскольку управляющая подсеть RWF-сети N бездефектна. □

Утверждение 7. *Множество блокировок конечно.*

Доказательство. Во-первых, заметим, что если $(c|r), (c|r + x) \in L$, то L не является тупиком. Действительно, в этом случае последовательность переходов $(c|r) \xrightarrow{\sigma} (c|r + x)$ является положительным циклом, который порождает бесконечное множество состояний — а это противоречит конечности числа состояний, входящих в блокировку. Следовательно, каждое управляющее состояние может встретиться в блокировке не более одного раза.

Предположим противное: существует бесконечно много блокировок. Тогда среди них бесконечно много блокировок с одним и тем же набором управляющих состояний. Эти блокировки отличаются друг от друга лишь значением счётчика. Следовательно, в этом множестве блокировок существуют блокировки со сколь угодно большим значением счётчика, и мы можем выбрать такую, из которой будет достижимо управляющее состояние o — а это противоречит определению блокировки. □

Следовательно, все блокировки могут быть найдены перебором конечных подмножеств множества $C(N)$, замкнутых относительно срабатывания переходов (сильно связных компонент) и удовлетворяющих свойству, сформулированному в Утв. 6.

Теорема 1. *Бездефектность разрешима для размеченных одномерных RWF-сетей.*

Доказательство. Далее используется приём, подобный использованному в [21] для доказательства разрешимости структурной бездефектности WF-сетей.

Для данной одномерной RWF-сети N построим модифицированную RWF-сеть $\overline{N}$, добавив новую начальную позицию $\overline{i}$ и два новых перехода так, как показано

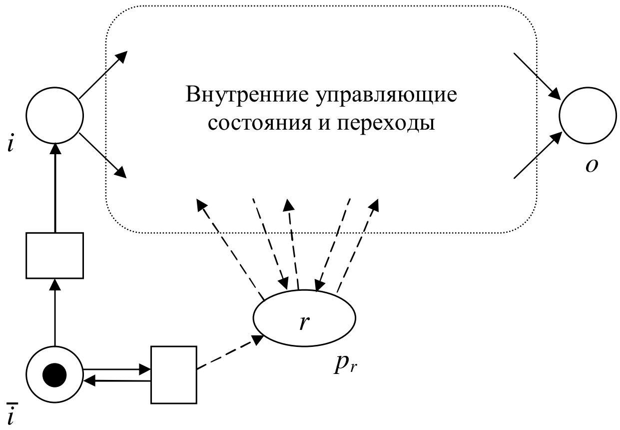

Рис. 7. Модифицированная RWF-сеть $\overline{N}$

на Рис. 7. Очевидно, что исходная сеть $(N, i|r)$ бездефектна тогда и только тогда, когда ни один из тупиков и ни одна из блокировок не достижимы в сети $(\overline{N}, \bar{i}|r)$.

Поскольку множества тупиков и блокировок конечны и вычислимы, проблема бездефектности размеченной одномерной RWF-сети сводится к проблеме достижимости в односчётчиковой сети Петри, которая является разрешимой (в частности, мы можем использовать для её решения алгоритм построения однопериодической свёртки пространства состояний). $\square$

4.3. Разрешимость бездефектности для неразмеченных сетей

Определение 8. *(Неразмеченная) RWF-сеть N называется* бездефектной, *если существует ресурс r, такой, что размеченная сеть $(N, i|r)$ бездефектна.*

Теорема 1 предоставляет только полуразрешающую процедуру для проблемы бездефектности сети. С её помощью можно проверить бездефектность при данной начальной разметке, но если ответ будет отрицательным, то мы не будем знать, существует ли бóльшая начальная разметка, при которой сеть станет бездефектной.

Следствие 2 даёт нам необходимое условие для тупика, достижимого от *некоторой* начальной разметки. Далее мы докажем более сильную теорему, предоставляющую необходимое и достаточное условие существования нарушающего бездефектность тупика (то есть тупика, достижимого от бесконечного числа различных начальных разметок).

Теорема 2. *Неразмеченная одномерная RWF-сеть не бездефектна из-за тупиков тогда и только тогда, когда в ней существуют тупиковое состояние $(c|r)$, отрицательный генератор q и простой маршрут $q \xrightarrow{\sigma} c$, такие, что*

$$Eff(\sigma) \ominus Supp(\sigma) \leq r.$$

Доказательство. ($\Leftarrow$) Достаточно показать, что для любого (достаточно большого) начального ресурса r_0 найдётся ещё бóльший начальный ресурс $r_0 + x$, такой, что от $(i|r_0 + x)$ будет достижим тупик.

Рассмотрим произвольный достаточно большой начальный ресурс r_0, такой, что выполнено

$$(i|r_0) \xrightarrow{\tau} (q|s)$$

для некоторого маршрута τ и ресурса s (всегда можно подобрать такой ресурс, поскольку управляющая подсеть бездефектна, и, следовательно, любое управляющее состояние достижимо при достаточно большом начальном количестве ресурса). Пусть $\theta = qc_1 \dots c_j q$ — простой отрицательный цикл с генератором q, то есть $Supp(\theta) = -Eff(\theta)$. Обозначим $z = s \ \mathbf{mod} \ Supp(\theta)$ и рассмотрим бо́льший начальный ресурс $r_0 + z + Supp(\sigma)$. Мы имеем:

$$\begin{array}{c} \big(i|r_0 + z + Supp(\sigma)\big) \\ \downarrow \ \tau \\ \big(q|s + z + Supp(\sigma)\big) \\ \downarrow \ \theta^{\left((s+z)/Supp(\theta)\right)} \\ \big(q|Supp(\sigma)\big) \\ \downarrow \ \sigma \\ \big(c|Eff(\sigma) \ominus Supp(\sigma)\big) \end{array}$$

и, следовательно, тупик.

($\Rightarrow$) Предположим противное: сеть не бездефектна из-за наличия тупиков, однако ни для одного из тупиковых состояний не выполнено свойство существования связанного отрицательного генератора.

Количество тупиковых состояний конечно, следовательно, некоторое тупиковое состояние $(c|r)$ достижимо от бесконечного числа различных начальных состояний (различных начальных значений счётчика).

Каждая последовательность переходов $\sigma = t_1.t_2.\dots.t_n$ от $(i|r_0)$ до $(c|r)$ соответствует маршруту σ в графе управляющего автомата. Поскольку существует бесконечно много порождающих данный тупик начальных состояний, множество соответствующих маршрутов также бесконечно. Каждый из этих маршрутов может быть разбит на последовательность чередующихся простых циклов и ациклических маршрутов:

$$\sigma = \tau_1(\theta_1)^{k_1}\tau_2(\theta_2)^{k_2} \dots \tau_{n-1}(\theta_{n-1})^{k_{n-1}}\tau_n.$$

Заметим, что такое разбиение не всегда единственно: например, $ababa$ может быть представлено и как $(ab)^2a$, и как $a(ba)^2$. Для определённости мы всегда рассматриваем “декомпозицию справа налево”, то есть в данном случае выбираем $a(ba)^2$.

Покажем, что среди всех этих маршрутов найдётся хотя бы один с отрицательным последним циклом θ_{n-1}. Действительно, если последний цикл положителен (или нейтрален) с эффектом x, то мы можем перейти к рассмотрению увеличенного начального ресурса $r_0 + x * k_{n-1}$ и укороченного маршрута

$$\sigma' = \tau_1(\theta_1)^{k_1}\tau_2(\theta_2)^{k_2} \dots \tau_{n-1}\tau_n,$$

имеющего то же самое завершение — тупик. Далее, новый маршрут σ' также может быть разбит на простые циклы и маршруты, затем его последний положительный цикл (если он опять окажется положительным) также может быть удален за счёт увеличения начального ресурса, и т.д. В конце этого процесса мы получим либо маршрут с отрицательным “последним циклом”, либо полностью ациклический маршрут (простой маршрут из i в c). В графе существует конечное число ациклических маршрутов, но у нас в наличии бесконечно много различных приводящих к тупику начальных разметок, так что второй вариант не может повторяться бесконечно много раз — и, следовательно, мы неизбежно построим маршрут с отрицательным циклом.

Рассмотрим такой “тупиковый” маршрут σ'' с окончанием $\theta^k\tau$, где θ — отрицательный цикл, а τ — ациклический маршрут. Пусть $\theta = c_1c_2\dots c_i\dots c_mc_1$, где c_i — отрицательный генератор (по Лемме 2 такое c_i всегда существует). Маршрут $\big((c_i\dots c_mc_1)\tau\big)$ является простым маршрутом (напомним, что мы раскладывали последовательность “справа налево” и, следовательно, $\theta\tau$ не содержит циклов, отличных от θ). Поскольку финальным состоянием всего маршрута σ'' является $(c|r)$, для любого окончания ϕ маршрута σ'' выполняется

$$\mathit{Eff}(\phi) \ominus \mathit{Supp}(\phi) \leq r.$$

То же самое выполняется и для $\big((c_i\dots c_nc_1)\tau\big)$. Но это простой маршрут, ведущий от отрицательного генератора к тупиковому управляющему состоянию — что и требовалось найти. □

Теорема 3. *Неразмеченная одномерная RWF-сеть не бездефектна из-за блокировок тогда и только тогда, когда в ней существуют блокирующее состояние $(c|r)$, отрицательный генератор q и простой маршрут $q \xrightarrow{\sigma} c$, такие, что*

$$\mathit{Eff}(\sigma) \ominus \mathit{Supp}(\sigma) \leq r.$$

Доказательство. Аналогично доказательству Теоремы 2. □

Следствие 3. *Бездефектность разрешима для неразмеченных одномерных сетей.*

Доказательство. Все простые (отрицательные) циклы могут быть найдены при помощи алгоритма Тарьяна, тупики и блокировки — при помощи перебора управляющих состояний и проверки свойств, указанных в Утв. 4 и Утв. 6 соответственно. Множество простых маршрутов конечно и может быть легко построено. □

4.4. Вычислимость минимального бездефектного ресурса

Рассмотрим “наивный” (и, возможно, не самый эффективный) алгоритм поиска минимального ресурса r, при котором сеть $(N, i|r)$ бездефектна. Этот алгоритм применим только к бездефектным сетям (бездефектность может быть проверена предварительно при помощи алгоритма, описанного в доказательстве Следствия 3).

Алгоритм 1. *(вычисление минимального бездефектного ресурса)*
Ввод: Бездефектная одномерная RWF-сеть N.
Вывод: Наименьшее r, такое, что сеть $(N, i|r)$ бездефектна.
Шаг 0: Пусть $r := 0$.
Шаг 1: Проверим, бездефектна ли $(N, i|r)$: будем искать тупики и блокировки, достижимые в размеченной служебной сети $(\overline{N}, \bar{i}|r)$(см. Теорему 1). Если “да”, то возвращаем r, иначе $r := r + 1$ и возвращаемся на Шаг 1.

Поскольку сеть бездефектна, число итераций конечно.

5. Заключение

В данной работе исследовано свойство бездефектности для сетей потоков работ с одной (неограниченной) ресурсной позицией. Доказано, что бездефектность разрешима для размеченных и неразмеченных сетей, представлен алгоритм вычисления минимального бездефектного ресурса.

В дальнейшем мы предполагаем рассмотреть разрешимость для общего случая RWF-сетей. Также планируется применить другие известные варианты бездефектности WF-сетей к нашему классу сетей потоков работ с бесконечными множествами состояний. В частности, достаточно интересной представляется так называемая ослабленная бездефектность (relaxed soundness). В ослабленно бездефектной WF-сети корректно завершаться должны только те процессы, в ходе которых хотя бы по одному разу срабатывает каждый переход. На остальные варианты исполнения никаких ограничений не накладывается.

Другими интересными вариантами бездефектности являются k-бездефектность, а также обобщённая и структурная бездефектность [5]. Кроме того, понятие достаточного (бездефектного) ресурса может оказаться важным и при разработке адаптивных процессов потоков работ [18].

Список литературы

1. *ван дер Аалст В., ван Хей К.* Управление потоками работ: модели, методы и системы. М.: Научный мир, 2007. (English: *van der Aalst W. M. P., van Hee K. M.* Workflow Management: Models, Methods and Systems. MIT Press, 2002.)

2. *Башкин В. А.* Формализация семантики систем с ненадежными агентами // Программирование. 2010. Т. 36. №4. С. 3–15. (English transl.: *Bashkin V. A.* Formalization of semantics of systems with unreliable agents by means of nets of active resources // Programming and Computer Software. 2010. **36**(4). P. 187–196.)

3. *Котов В. Е.* Сети Петри. М.: Наука, 1984. (*Kotov V. E.* Seti Petri. Moskva: Nauka, 1984 [in Russian].)

4. *van der Aalst W. M. P.* The Application of Petri Nets to Workflow Management // The Journal of Circuits, Systems and Computers. 1998. **8**(1). P. 21–66.

5. *van der Aalst W. M. P., van Hee K. M., Hofstede A. H. M., Sidorova N., Verbeek H. M. W., Voorhoeve M., Wynn M. T.* Soundness of workflow nets: classification, decidability, and analysis // Formal Aspects of Computing. 2011. **23**(3). P. 333–363.

6. *Abdulla P. A., Čerans K.* Simulation is decidable for one-counter nets // Proc. of CONCUR'98. Lecture Notes in Computer Science. 1998. Vol. 1466. P. 253–268.

7. *Barkaoui K., Petrucci L.* Structural Analysis of Workflow Nets with Shared Resources // Proc. of Workflow Management: Net-based Concepts, Models, Techniques and Tools (WFM98). Computing Science Reports. Eindhoven University of Technology. 1998. Vol. 98/7. P. 82–95.

8. *Barkaoui K., Ben Ayed R., Sbaï Z.* Workflow Soundness Verification based on Structure Theory of Petri Nets // International Journal of Computing and Information Sciences. 2007. **5**(1). P. 51–61.

9. *Bashkin V. A., Lomazova I. A.* Resource Similarities in Petri Net Models of Distributed Systems // Proc. of PaCT 2003. Lecture Notes in Computer Science. 2003. Vol. 2763. P. 35–48.

10. *Bashkin V. A., Lomazova I. A.* Petri Nets and resource bisimulation // Fundamenta Informaticae. 2003. Vol. 55, No 2, P. 101–114,

11. *Bashkin V. A., Lomazova I. A.* Resource equivalence in workflow nets // Proc. of Concurrency, Specification and Programming (CS&P'2006). Humboldt Universitat zu Berlin, 2006. Vol. 1. P. 80–91.

12. *Chrząstowski-Wachtel P.* Sound Markings in Structured Nets // Proc. of Concurrency, Specification and Programming (CS&P'2005). Warsaw University, 2005. P. 71–85.

13. *Dickson L. E.* Finiteness of the Odd Perfect and Primitive Abundant Numbers with n Distinct Prime Factors // American Journal of Mathematics. 1913. Vol. 35, No 4. P. 413–422.

14. *van Hee K., Sidorova N., Voorhoeve M.* Generalized Soundness of Workflow Nets is Decidable // Proc. of ICATPN 2004. Lecture Notes in Computer Science. 2004. Vol. 3099. P. 197–216.

15. *van Hee K., Serebrenik A., Sidorova N., Voorhoeve M.* Soundness of Resource-Constrained Workflow Nets // Proc. of ICATPN 2005. Lecture Notes in Computer Science. 2005. Vol. 3536. P. 250–267.

16. *van Hee K., Oanea O., Serebrenik A., Sidorova N., Voorhoeve M., Lomazova I. A.* Checking Properties of Adaptive Workflow Nets // Fundamenta Informaticae. 2007. Vol. 79, No 3. P. 347–362.

17. *Hopcroft J., Pansiot J.-J.* On the reachability problem for 5-dimensional vector addition systems // Theoretical Computer Science. 1979. **8**(2). P. 135–159.

18. *Lomazova I.A.* Nested Petri Nets for Adaptive Process Modeling // Lecture Notes in Computer Science. 2008. Vol. 4800. P. 460–474.

19. *Petri C. A.* Kommunikation mit Automaten. Phd thesis. Bonn, Institute für Instrumentelle Mathematik. 1962.

20. *Sidorova N., Stahl C.* Soundness for Resource-Contrained Workflow Nets is Decidable // BPM Center Report BPM-12-09. BPMcenter.org. 2012.

21. *Tiplea F. L., Marinescu D. C.* Structural soundness of workflow nets is decidable // Information Processing Letters. 2005. Vol. 96. P. 54–58.

On the Decidability of Soundness of Workflow Nets with an Unbounded Resource

Bashkin V.A., Lomazova I.A.

P.G. Demidov Yaroslavl State University,
Sovetskaya str., 14, Yaroslavl, 150000, Russia
National research university "Higher school of economics"
ul. Myasnitskaya, 20, Moskva, 101000 Russia
Program Systems Institute of RAS
Pereslavl-Zalessky, Yaroslavl Region, 152020 Russia

Keywords: Petri nets, workflow, resource, soundness, verification, modeling

In this work, we consider the modeling of workflow systems with Petri nets. A resource workflow net (RWF-net) is a workflow net supplied with an additional set of initially marked resource places. Resources can be consumed and/or produced by transitions. We constrain neither the intermediate nor final resource markings, hence a net can have an infinite number of different reachable states. An initially marked RWF-net is called sound if it properly terminates its work and, moreover, an increase of the initial resource does not violate its proper termination. An unmarked RWF-net is sound if it is sound for some initial resource. In this paper, we prove the decidability of both marked and unmarked soundness for a restricted class of RWF-nets with a single unbounded resource place (1-dim RWF-nets). We present an algorithm for computing the minimal sound resource for a given sound 1-dim RWF-net.

Сведения об авторах:
Башкин Владимир Анатольевич,
Ярославский государственный университет им. П. Г. Демидова,
канд. физ.-мат. наук, доцент;
Ломазова Ирина Александровна,
Национальный исследовательский университет “Высшая школа экономики”,
д-р физ.-мат. наук, профессор,
Институт программных систем РАН, зав. международной научно-учебной
лабораторией процессно-ориентированных информационных систем,
гл. науч. сотр.