СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ, 2020. Т. 30. № 4. С. 61–75

ПОСТРОЕНИЕ МОДЕЛЕЙ ПРОЦЕССА С ПОМОЩЬЮ ПРОСТЫХ СЕТЕЙ ПЕТРИ*

***И. Ю. Терёхина*[1], *А. А. Грушо*[2], *Е. Е. Тимонина*[3], *С. Я. Шоргин*[4]**

Аннотация: Цель данной работы — исследование класса простых сетей Петри для моделирования рабочего процесса. Показано, что при отсутствии ограничения на уникальность задач в логе построить корректную модель рабочего процесса не всегда удается. Также показано, что если в модели есть переходы, которые не соответствуют ни одной задаче лога рабочего процесса, то в модели рабочего процесса будут нарушены предположения о взаимосвязи между задачами в логе рабочего процесса и между переходами соответствующей сети Петри.

Ключевые слова: сеть Петри; обнаружение процесса; построение модели процесса

DOI: 10.14357/08696527200406

1 Введение

Под термином майнинг (обнаружение) процесса понимается метод построения модели процесса из множества примеров реализации процесса. Задача обнаружения процесса не нова, но все еще остается актуальной. В контексте рабочих процессов данная задача впервые изучалась в работе [1], после чего появилось множество различных подходов к ее решению (см., например, [2–6]).

Основные предположения, которые, как правило, накладываются на рабочий процесс и его лог:

– каждое событие относится к некоторой задаче (шаг рабочего процесса);
– каждое событие относится к некоторой реализации рабочего процесса;
– события упорядочены (притом что задачи могут исполняться параллельно);
– события атомарны.

*Работа частично поддержана РФФИ (проект 18-07-00274).

[1]Факультет вычислительной математики и кибернетики Московского государственного университета им. М. В. Ломоносова, iteryokhina@cs.msu.ru

[2]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, grusho@yandex.ru

[3]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, eltimon@yandex.ru

[4]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, sshorgin@ipiran.ru

В работе [6] описываются техники для обнаружения моделей рабочих процессов в терминах сетей Петри [7] и предложен алгоритм, который может обнаруживать рабочий процесс, представленный его логом. Считается [8], что моделирование в терминах сетей Петри помогает получить дополнительную информацию о структуре и динамическом поведении моделируемой системы.

Цель данной работы — исследование предложенного в работе [6] класса сетей Петри для моделирования рабочего процесса. Показано, что при отсутствии ограничения на уникальность задач в логе построить корректную модель рабочего процесса не всегда удается. Также показано, что если в модели есть переходы, которые не соответствуют никакой задаче в логе, в частности вспомогательные переходы AND-split, AND-join, OR-split, OR-join, то нарушаются теоремы работы [6] о взаимосвязи между задачами в логе рабочего процесса и между переходами соответствующей сети Петри. Тем самым показывая, что алгоритм обнаружения процесса по логу из работы [6] будет работать лишь для узкого класса рабочих процессов, в которых нет повторов задач и условий «ИЛИ».

2 Необходимые свойства модели для ее обнаружения

Под сетью Петри будем понимать тройку $N = (P, T, F)$, где P — конечное множество *мест*; T — конечное множество *переходов*, $P \cap T = \emptyset$; $F \subseteq (P \times T) \cup \cup (T \times P)$ — множество направленных некратных дуг, отношение инцидентности.

Для описания динамического поведения сети Петри вводится определение *разметки* над множеством мест P. Обычно под разметкой понимается мультимножество мест из P [6]. Так, запись $s = \{p_1, p_1, p_2\}$ означала бы, что в сети N есть 3 *метки*, 2 из них находятся на месте $p_1 \in P$, одна метка стоит на месте $p_2 \in P$. В рамках данной работы будут рассматриваться разметки, в которых на каждом месте сети N может находиться не более одной метки. Таким образом, разметка s — это множество мест из P, на которых есть метка.

Пусть $\mathcal{N}$ — множество всех размеченных сетей Петри, т. е. пар вида (N, s).

В работе [6] вводится ряд свойств корректности, накладываемых на сеть Петри для того, чтобы модель могла быть обнаружена по логу рабочего процесса. В данной работе будет проведен анализ совместимости свойств:

(1) надежности;

(2) запрета последовательного выполнения в сети конструкций синхронизации и выбора;

(3) полноты лога рабочего процесса для рассматриваемой сети;

(4) сохранения в сети отношения каузальности между переходами, если между соответствующими задачами в логе это отношение было выполнено.

Введем определения и сформулируем теорему в соответствии с вышеупомянутыми свойствами. Далее будут рассматриваться сети Петри $N = (P, T, F)$ с одним входным и выходным местом такие, что $\forall\, x, y \in P \cup T, xF^*y$, где F^* —

рефлексивное и транзитивное замыкание отношения F. Элементы из $P \cup T$ будем называть *узлами*. Введем обозначения входных и выходных узлов для некоторого узла x сети N: $(\cdot x) = \{y | (y, x) \in F\}$, $(x \cdot) = \{y | (x, y) \in F\}$.

Динамическое поведение сети Петри описывается правилом срабатывания переходов.

Определение 1 (срабатывание переходов). Пусть $(N = (P, T, F), s)$ — сеть Петри с метками. Переход $t \in T$ называется *активированным*, $(N, s)[t\rangle$, если $(\cdot t) \subseteq s$. Пусть $t \in T$ — активированный переход в сети N. *Срабатывание перехода* t приводит сеть N с разметкой s к новой разметке $s \cup (t\cdot) \backslash (\cdot t)$ обозначение $(N, s)[t\rangle (N, s \cup (t\cdot) \backslash (\cdot t))$.

Определение 2 (достижимые разметки). Пусть (N, s_0) — размеченная сеть Петри из $\mathcal{N}$. Разметка s называется *достижимой* из разметки s_0, если существует последовательность активированных переходов, таких что их последовательное срабатывание приведет из разметки s_0 в разметку s. Множество достижимых разметок (N, s_0) обозначается $[N, s_0\rangle$.

Далее будут рассматриваться сети Петри с двумя выделенными местами: $i \in P$ — входное место сети и $o \in P$ — выходное место сети. Предполагается, что в сеть N новая метка может поступать только на входное место и покидать сеть только через выходное место сети.

Рассмотрим пример размеченной сети Петри $(N = (P, T, F), \{i\})$, где $P = \{i, o, p_1, p_2\}$, $T = \{i, o, t_1, t_2\}$, $F = \{(i, t_1), (t_1, p_1), (t_1, p_2), (p_1, t_2), (t_2, o)\}$ (рис. 1). Метка на месте i активирует переход t_1. Множество входящих мест в переход $t_1 : (\cdot t_1) = \{i\}$, множество выходящих мест $(t_1 \cdot) = \{p_1, p_2\}$

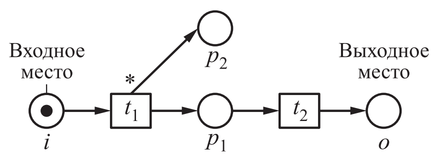

Рис. 1 Размеченная сеть $(N, \{i\})$. Активированный переход отмечен знаком «*»

Пусть срабатывает переход t_1. Стоит отметить, что если в сети Петри есть несколько активированных переходов, то в следующий момент времени может сработать любое подмножество из них [8]. В данной работе предполагается, что в один момент времени может сработать только один из активированных переходов. По правилу срабатывания переходов новая разметка для сети N после срабатывания перехода: $t_1 : s \cup (t\cdot) \backslash (\cdot t) = \{i\} \cup \{p_1, p_2\} \backslash \{i\} = \{p_1, p_2\}$, становится активированным переход t_2 (рис. 2).

Пусть срабатывает переход t_2. Множество входящих мест в переход t_2: $(\cdot t_2) = \{p_1\}$, множество выходящих мест $(t_2 \cdot) = \{o\}$. Новая разметка после срабатывания перехода t_2 по правилу срабатывания переходов: $\{p_1, p_2\} \cup \{o\} \backslash \{p_1\} = \{p_2, o\}$ (рис. 3).

Множество достижимых разметок для примера на рис. 1: $[N, \{i\}\rangle = \{\{i\}, \{p_1, p_2\}, \{p_2, o\}\}$.

Свойство 1 определяется следующим образом.

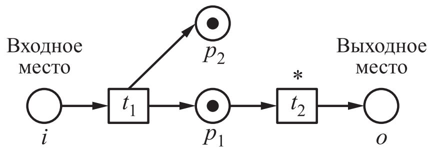

Рис. 2 Размеченная сеть $(N, \{p_1, p_2\})$

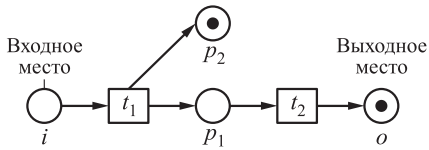

Рис. 3 Размеченная сеть $(N, \{p_2, o\})$

Определение 3 (свойство надежности). Пусть $N = (P, T, F)$ — сеть Петри с входным местом $i \in P$ и выходным местом $o \in P$. Сеть N называется *надежной*, если выполнены следующие условия:

– *корректное завершение*. Для всех достижимых разметок $\forall\, s \in [N, \{i\}\rangle$, если $o \in s$, то $s = \{o\}$;
– *отсутствие мертвых задач*. Каждый из переходов сети Nактивируется в какой-либо достижимой разметке $(N, \{i\})$.

Пример на рис. 1 не обладает свойством надежности, так как не выполнено условие корректного завершения.

Перейдем к рассмотрению свойства 2, касающегося запрещенных конструкций (рис. 4). Эти конструкции соответствуют ситуациям, когда происходят подряд выбор и синхронизация. Под *выбором* в контексте сетей Петри понимается конструкция, когда из одного места есть ребра в несколько переходов. Под *синхронизацией* — когда из нескольких мест есть ребра в один и тот же переход. Таким образом, рис. 4, *а* описывает ситуацию «синхронизация и выбор»: ситуация выбора задачи (место с многими выходными переходами) не должна происходить одновременно с синхронизацией задач (переходы с многими входными местами). Рисунок 4, *б* описывает ситуацию «выбор и синхронизация»: если происходит синхронизация задач (переход с многими входными местами), то все предшествующие ей задачи должны быть выполнены, т. е. синхронизация задач не должна происходить сразу после операции «ИЛИ».

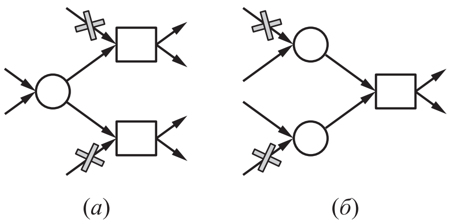

Рис. 4 Запрещенные конструкции согласно свойству 2

Определение 4 (структурированная сеть Петри). $N = (P, T, F)$ — *структурированная сеть Петри*, если $\forall\, p \in P, \forall\, t \in T$, где $(p, t) \in F$, выполнено:

– если $|(p\cdot)| > 1$, то $|(\cdot t)| = 1$;
– если $|(\cdot t)| > 1$, то $|(\cdot p)| = 1$;
– в сети нет неявных мест. Место $p \in P$ называется *неявным*, если для всех достижимых разметок $s' \in [N, s\rangle$ и переходов $t \in (p\cdot)$ выполнено: если $s' \subseteq (\cdot t) \backslash \{p\}$, то $s' \subseteq (\cdot t)$.

Для описания свойства 3 необходимо ввести дополнительные определения отношений порядка между задачами в логе и срабатывающей последовательности в сети Петри.

Пусть T — конечное множество задач, тогда $\sigma \in T^*$ — *трасса*, $W \subseteq T^*$ — *лог*.

Определение 5 (отношение порядка между задачами в логе). Пусть W — лог над T, и $a, b \in T$. Тогда:

- $a >_W b$ (a предшествует b), если $\exists\, \sigma = t_1, \ldots, t_{n-1}$ и $i \in \{1, \ldots, n-2\}$ такие, что $\sigma \in W$ и $t_i = a$, $t_{i+1} = b$;
- $a \rightarrow_W b$ (прямое каузальное отношение), если $a >_W b$ и $b \not>_W a$;
- $a \#_W b$ (задачи не встречаются вместе), если $a \not>_W b$ и $b \not>_W a$;
- $a \|_W b$ (потенциальный параллелизм), если $a >_W b$ и $b >_W a$.

Определение 6 (срабатывающая последовательность). Пусть дана размеченная сеть $(N = (P, T, F), s_0)$. Последовательность переходов $\sigma \in T^*$ называется *срабатывающей* для (N, s_0), если $\exists\, n \in \mathbb{N}$ такое, что существуют разметки $s_1, \ldots, s_n$ и переходы $t_1, \ldots, t_n \in T$ такие, что $\sigma = t_1, \ldots, t_n$ и для всех i, $0 \le i < n$ переход t_{i+1} активирован в размеченной сети (N, s_i) и $s_{i+1} = s_i \cup \cup (t_{i+1}\cdot) \backslash (\cdot t_{i+1})$. Обозначение: $(N, s_0)[\sigma\rangle$ Срабатывание последовательности σ ведет к разметке s_n, обозначение: $(N, s_0)[\sigma\rangle(N, s_n)$.

Так, в примере на рис. 1 срабатывающей будет последовательность $\sigma = \{t_1, t_2\}$ и после срабатывания разметка $\{i\}$ перейдет в разметку $\{p_2, o\}, (N, \{i\})[\delta\rangle(N, \{p_2, o\})$.

Определим свойство 3.

Определение 7 (полнота лога рабочего процесса). Пусть $N = (P, T, F)$ — надежная сеть Петри. W — *лог рабочего процесса* для сети N, если $W \subseteq T^*$ и $\forall\, \sigma \in W$ — срабатывающая последовательность в N, начиная с разметки $\{i\}$ и заканчивая разметкой $\{o\}$. W — *полный лог рабочего процесса* для сети N, если:

- W — лог рабочего процесса;
- для любого другого лога рабочего процесса W' сети N выполнено: $>'_W \subseteq >_W$;
- $\forall\, t \in T\; \exists\, \sigma \in W : t \in \sigma$ — все переходы покрываются некоторой срабатывающей последовательностью.

Свойство 4 формулируется теоремой в работе [6]. Из теоремы 1 следует, что если существует каузальное отношение между задачами согласно логу рабочего процесса, то в соответствующей сети Петри существует место, соединяющее переходы, соответствующие этим двум задачам.

Теорема 1 [6]. *Пусть $N = (P, T, F)$ — надежная сеть Петри и W — полный лог рабочего процесса для сети N. Для всех $a, b \in T$ таких, что $a \rightarrow_W b$, следует*

$$(a\cdot) \cap (\cdot b) \neq \emptyset\,.$$

3 Описание рабочего процесса

Рассматривается задача построения модели рабочего процесса, который представлен логом W. Если такая модель рабочего процесса будет построена, то для нее будет применим алгоритм переобнаружения процесса, предложенный в работе [6]. В данной работе приведен пример рабочего процесса, состоящего из четырех задач, для которого в терминах предложенного в работе [6] математического аппарата не удается построить корректную модель, если под корректностью понимается наличие хотя бы четырех свойств, описанных в разд. 2.

Рассмотрим процесс, состоящий:

– из задачи A, которая всегда выполняется в начале;

– из блока «ИЛИ»: либо задача A, либо задача B, либо никакой задачи;

– из задач C и D, которые всегда выполняются в конце.

Рассмотрим полный лог для такого процесса: $W = \{ACD, AACD, ABCD\}$. Справедливые отношения для лога W:

– отношение предшествия между действиями (задачами, переходами в сети Петри): $A >_W A$, $A >_W B$, $A >_W C$, $B >_W C$ и $C >_W D$;

– прямое каузальное отношение: $A \to_W B$, $A \to_W C$, $B \to_W C$ и $C \to_W D$;

– отношение параллелизма: $A \|_W A$;

– задачи не встречаются вместе (симметричное отношение, для краткости не приведены записи вида bRa, если есть запись aRb): $A \#_W D$, $B \#_W B$, $B \#_W D$, $C \#_W C$ и $D \#_W D$.

4 Модели рабочего процесса

Рассмотрим возможные сети Петри для лога W с целью их исследования на наличие четырех свойств, введенных в разд. 2. Можно рассматривать сети Петри, которые представлены в данном разделе, как потенциальные модели процесса, так как трассы лога W могут быть порождены этими сетями. Или, другими словами, сети N_1, N_3, N_4 и N_5 *моделируют* лог W. Сеть N_2 представлена

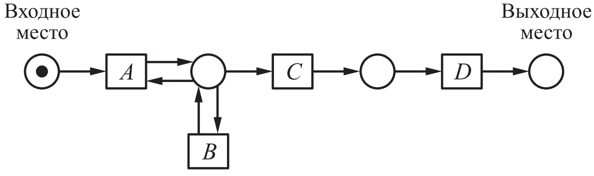

Рис. 5 Сеть N_1 для лога W

как пример неверной работы сложных блоков OR-split и OR-join в классе сетей Петри, предлагаемых авторами [6].

Рассмотрим сеть Петри N_1 (рис. 5) на наличие свойств, определенных в разд. 2.

Утверждение 1. *Сеть N_1 надежна.*

Доказательство. Проверка свойств по определению надежности.

1. *Корректное завершение*. Так как в сети N_1 нет переходов, увеличивающих число меток (переход с несколькими выходными местами), то если в разметке $\{i\}$ была 1 метка и эта метка дойдет до выходного места, то она будет единственной.

2. *Отсутствие мертвых переходов*. Трасса $ABCD$ активирует каждый из переходов сети N_1. Построенная сеть N_1 не является структурированной сетью (свойство 2), так как содержит обе запрещенные конструкции (см. рис. 4). На рис. 6 и 7 ребра сети, не участвующие в демонстрации нарушения, изображены пунктирными стрелками, а ребра, демонстрирующие нарушение условия, — толстыми:

   – нарушение условия $\forall(p,t) \in F$, если $|(p\cdot)| > 1$, то $|(\cdot t)| = 1$ (см. рис. 6);
   – нарушение условия $\forall(p,t) \in F$, если $|(\cdot t)| > 1$, то $|(\cdot p)| = 1$ (см. рис. 7).

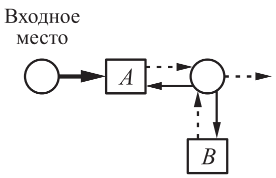

Рис. 6 Синхронизация и выбор в сети N_1

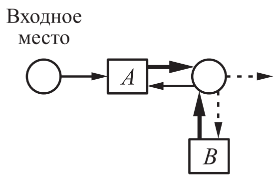

Рис. 7 Выбор и синхронизация в сети N_1

Легко видеть, что сеть моделирует большее число трасс, чем представлено в логе W. Например, срабатывающими последовательностями будут трассы $A\ldots ACD$, $AB\ldots BCD$, которых нет в логе W. Таким образом, сеть не обладает свойством 3 (полноты лога).

У сети N_1 налицо свойство 4, так как для каждой пары задач в логе W, между которыми есть каузальное отношение, между соответствующими переходами существует место.

Авторы работы [6] полагают, что структурированные сети поддерживают сложные блоки вида OR-split, OR-join, AND-split и AND-join. Так как в логе W есть конструкция «ИЛИ», построим сеть Петри N_2 с переходами OR-split и OR-join (рис. 8).

В рассматриваемом классе сетей Петри условия «И» и «ИЛИ» моделируются с помощью следующих конструкций [7] (рис. 9).

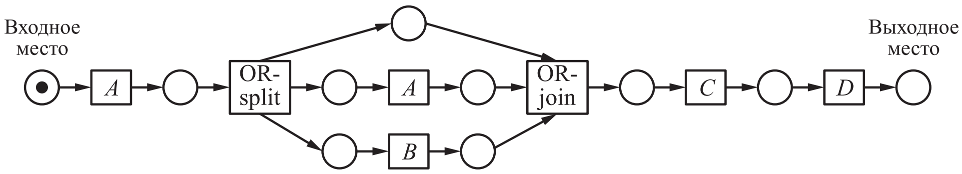

Рис. 8 Сеть N_2 для лога W

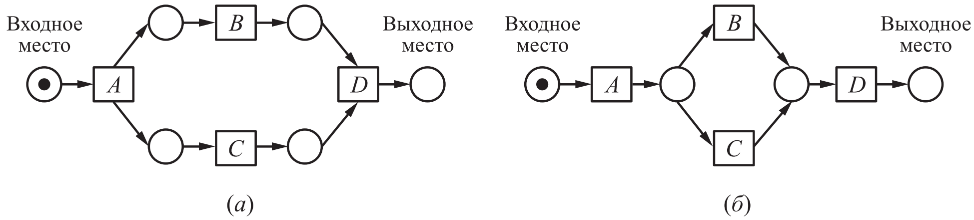

Рис. 9 Конструкции параллельного исполнения (условие «И») (*а*) и «race condition» (условие «ИЛИ») (*б*)

Таким образом, без наложения дополнительных условий на переходы OR-split и OR-join в сети N_2 эти переходы будут моделировать условие «И», а не условие «ИЛИ», т. е. сеть N_2 будет моделировать не лог W, а лог $W' = \{AABCD, ABACD\}$ Решение данной проблемы может заключаться в рассмотрении более сложного класса сетей Петри с наложением условий на переходы, например класс ингибиторных сетей Петри.

Рассмотрим полный лог W' для сети N_2. В сети N_2 есть сложные переходы OR-split и OR-join (которые работают в точности, как AND-split и AND-join), и такая сеть все равно не будет корректной, так как не обладает свойством 4:

– лог W' не подразумевает наличия дополнительных переходов, кроме указанных в логике процесса задач (в частности, задач OR-split и OR-join), откуда не может быть определено более полное каузальное отношение $\to$ (чтобы появились конструкции вида $A \to$ OR-join);

– в логе W' справедливо каузальное отношение $B \to_{W'} C$, но при этом в сети N_2, в срабатывающих разметках, метка может находиться либо между переходами B и OR-join (рис. 10), в таком случае $(\cdot C) = \emptyset$, откуда $(B\cdot) \cap (\cdot C) = \emptyset$, либо метка может находиться между переходами OR-join и C (рис. 11), в таком случае $(B\cdot) = \emptyset$, откуда $(B\cdot) \cap (\cdot C) = \emptyset$.

Используем конструкцию «ИЛИ» с рис. 9, *б* и построим новую сеть Петри N_3 без переходов OR-split и OR-join (рис. 12). Основной проблемой остается моделирование пустого перехода в условии «ИЛИ», так как места в сетях Петри не могут быть соединены с местами.

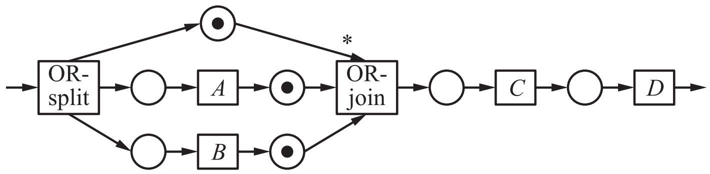

Рис. 10 Часть сети N_2 после срабатывания последовательности AAB или ABA

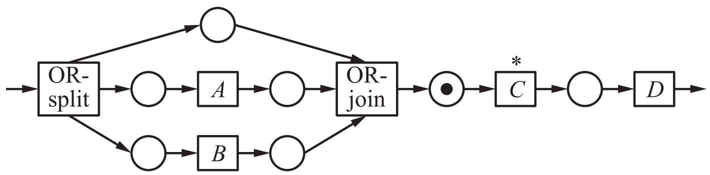

Рис. 11 Часть сети N_2 после срабатывания последовательности $AABC$ или $ABAC$

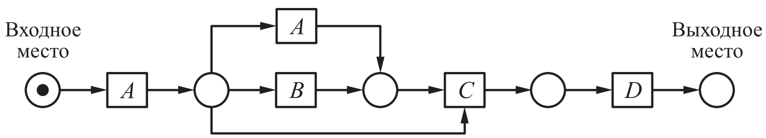

Рис. 12 Сеть N_3 для лога W

Утверждение 2. *Сеть N_3 надежна.*

Доказательство. Проверка свойств по определению надежности.

1. *Корректное завершение*. В сети N_3 нет переходов, у которых есть несколько выходных мест, поэтому число меток не может увеличиться. Если в сеть Петри не может поступить новая метка, пока находящаяся в ней метка не дойдет до выходного места, то если эта метка окажется в выходном месте, она будет единственной.

2. *Отсутствие мертвых переходов*. Трасса ACD активирует каждый из переходов сети N_3.

Построенная сеть N_3 не является структурированной сетью (свойство 2), так как содержит обе запрещенные конструкции из рис. 4. На рис. 13 и 14 ребра сети, не участвующие в демонстрации нарушения, изображены пунктирными стрелками, а ребра, демонстрирующие нарушение условия, — жирными стрелками:

– нарушение условия $\forall (p,t) \in F$, если $|(p\cdot)| > 1$, то $|(\cdot t)| = 1$ (см. рис. 13);

– нарушение условия $\forall (p,t) \in F$, если $|(\cdot t)| > 1$, то $|(\cdot p)| = 1$ (см. рис. 14).

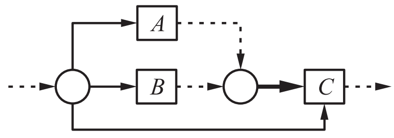

Рис. 13 Синхронизация и выбор в сети N_3

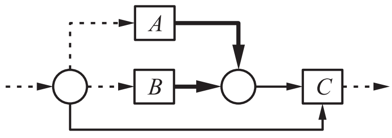

Рис. 14 Выбор и синхронизация в сети N_3

W является полным логом рабочего процесса для сети N_3 (наличие свойства 3), так как содержит все срабатывающие последовательности для этой сети. Различие в срабатывающих последовательностях для сети N_3 моделируется условием «ИЛИ»: префиксы трасс AAC, ABC или AC. Число различных трасс равно 3, и эти трассы в точности совпадают с логом W.

Лог W и сеть N_3 не обладают свойством 4, касающимся наличия общего места в сети, если в логе между соответствующими задачами справедливо каузальное отношение, и можно доказать более общую теорему.

Теорема 2. *Теорема 1 не выполняется, когда в полном логе W для сети N содержатся повторы задач.*

Доказательство. В доказательство теоремы рассмотрим построенный пример с сетью Петри N_3 и полным логом рабочего процесса W.

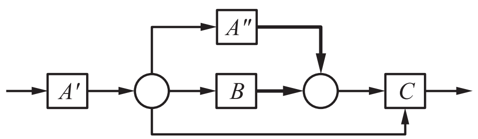

Рис. 15 Часть сети N_3 для иллюстрации противоречия теореме 1 (одинаковые переходы с задачей A размечены дополнительно)

Для лога W справедливо отношение $A \rightarrow_W B$. Согласно теореме 1 из этого отношения должно следовать, что существует общее место для переходов с задачами A и B в сети Петри N_3. Рассмотрим часть сети N_3 (рис. 15). Можно видеть, что между переходами A'' и B нет общего места.

Проблемы с сетью N_3 связаны с тем, что происходит попытка моделирования условия «ИЛИ», где одна из возможных веток не содержит ни одной задачи; таким образом, выбор и синхронизация должны происходить подряд. Поменяем конструирование пустого перехода в условии «ИЛИ» и построим новую сеть N_4 (рис. 16). Таким образом, у сети N_4 будет заведомо присутство-

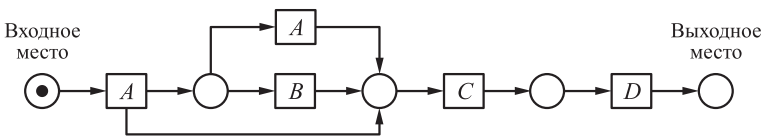

Рис. 16 Сеть N_4 для лога W

вать свойство 2, касающееся того, что сеть не должна содержать запрещенных конструкций. Свойство 3, связанное с полнотой лога W для N_4, также заведомо верно, по построению.

Утверждение 3. *Сеть N_4 некорректно завершается.*

Доказательство. Для иллюстрации противоречия рассмотрим срабатывающую последовательность переходов $\sigma = ACD$, переводящую метку со входного места на выходное место — $(N_4, \{i\})[\sigma\rangle(N_4, \{o\})$:

– разметка $\{i\}$ активирует переход A (рис. 17);

– срабатывает переход A, появляются 2 метки, которые активируют 3 перехода (рис. 18);

– срабатывает переход C, меняет местоположение одна метка из двух, активированы 3 перехода (рис. 19);

– срабатывает переход D, метка перемещается на выходное место, в сети еще остается одна метка между переходами A, A и B (рис. 20).

Таким образом, обнаружена достижимая разметка из разметки $\{i\}$ такая, что в этой разметке есть метка, находящаяся на выходном месте, и она не единственная.

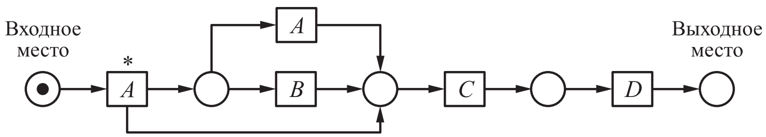

Рис. 17 Часть сети N_4 в начальной разметке. Активированный переход отмечен знаком «*»

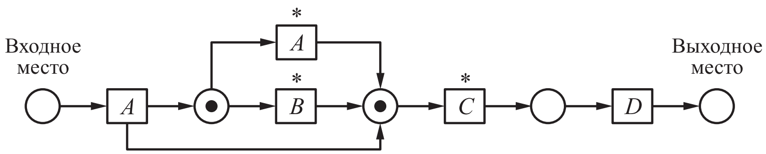

Рис. 18 Часть сети N_4 после срабатывания перехода A

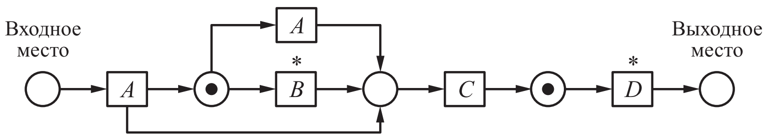

Рис. 19 Часть сети N_4 после срабатывания перехода C

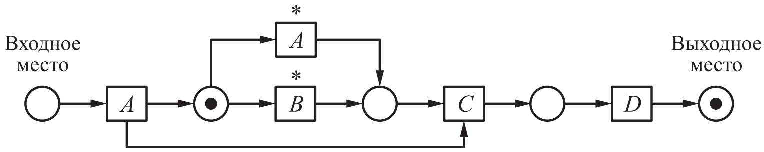

Рис. 20 Часть сети N_4 после срабатывания перехода D

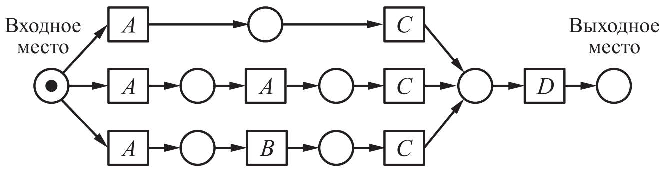

Рис. 21 Сеть N_5 для лога W

Так как сеть N_4 не завершается корректно, она не может быть надежной. Свойство 1 отсутствует. Свойство 4 также отсутствует, так как для сети N_4 справедлива теорема 2.

Увеличим число переходов в сети, для того чтобы получилась каноническая конструкция условия «ИЛИ», как на рис. 9, *б*. Сеть N_5 представлена на рис. 21.

Сеть N_5 будет обладать свойствами полноты лога и отсутствия запрещенных конструкций по построению. В то же время будет отсутствовать свойство, обеспечивающее выполнение теоремы 1, так как, например, не между каждой парой переходов с задачамиA и B существует общее место.

Утверждение 4. *Сеть N_5 надежна.*

Доказательство. Проверка свойств по определению надежности.

1. *Корректное завершение*. В сети N_5 есть только тип переходов, который не увеличивает число меток. Если в сеть Петри не может поступить новая метка, пока находящаяся в ней метка не дойдет до выходного места, то если эта метка окажется на выходном месте, она будет единственной.

2. *Отсутствие мертвых переходов*. Трассы ACD, $AACD$ и $ABCD$ активируют каждый из переходов сети N_5.

5 Заключение

В статье проведено исследование предложенного в работе [6] класса сетей Петри для моделирования рабочего процесса. Показано, что при отсутствии ограничения на уникальность задач в логе и наличии в модели переходов, которые не соответствуют никакой задаче в логе, построение модели, которая бы обладала хотя бы рассматриваемыми четырьмя свойствами, является сложной

Сравнение сетей Петри для одного процесса

Сеть	Надежность	Нет запрещенных конструкций	Полнота лога	Сохранение отношения каузальности
N_1	+	–	–	+
N_2	–*	–*	–*	–*
N_3	+	–	+	–
N_4	–	+	+	–
N_5	+	+	+	–

задачей, где выполнение одних требований часто противоречит выполнению других. Сравнительные характеристики рассмотренных сетей по свойствам представлены в таблице.

Также в работе показано, что переходы OR-split и OR-join без наложения дополнительных условий на рассматриваемый класс сетей Петри будут работать в точности, как переходы AND-split и AND-join, и будут моделировать другой процесс. Поэтому в таблице результаты сравнений по требованиям для сети N_2 помечены знаком «*».

Таким образом, можно говорить о том, что алгоритм обнаружения процесса по логу из статьи [6] справедлив для еще более узкого класса рабочих процессов, чем предполагается в [6]. Более точно, в логе рабочего процесса задачи должны быть уникальными, не должно быть переходов по пустому условию, не должно быть моделирования условий «ИЛИ».

Литература

1. *Agrawal R., Gunopulos D., Leymann F.* Mining process models from workflow logs // Advances in database technology / Eds. H. J. Schek, G. Alonso, F. Saltor, I. Ramos. — Lecture notes in computer science ser. — Springer, 1998. Vol. 1337. P. 467–483.
2. *Cook J. E., Wolf A. L.* Discovering models of software processes from event-based data // ACM T. Softw. Eng. Meth., 1998. Vol. 7. No. 3. P. 215–249.
3. *Mannila H., Rusakov D.* Decomposition of event sequences into independent components // SIAM Conference (International) on Data Mining Proceedings. — Society for Industrial and Applied Mathematics, 2001. P. 1–17.
4. *Schimm G.* Process miner — a tool for mining process schemes from event-based data // Logics in artificial intelligence / Eds. S. Flesca, S. Greco, G. Ianni, N. Leone. — Lecture notes in computer science ser. — Springer, 2002. Vol. 2424. P. 525–528.
5. *Herbst J.* Ein induktiver Ansatz zur Akquisition und Adaption von Workflow-Modellen. — Berlin: Tenea Verlag Ltd., 2004. 284 p.
6. *Van der Aalst W., Weijters T., Maruster L.* Workflow mining: Discovering process models from event logs // IEEE T. Knowl. Data En., 2004. Vol. 16. No. 9. P. 1128–1142.
7. Lectures on Petri nets I: Basic models: Advances in Petri nets / Eds. W. Reisig, G. Rozenberg. — Berlin–Heidelberg: Springer-Verlag, 1998. 691 p.

8. *Питерсон Д.* Теория сетей Петри и моделирование систем / Пер. с англ. — М.: Мир, 1984. 264 с. (*Peterson J. L.* Petri net theory and the modeling of systems. — Prentice-Hall, 1981. 290 p.)

Поступила в редакцию 11.09.20

CONSTRUCTING PROCESS MODELS REPRESENTED BY SIMPLE PETRI NETS

***I. Yu. Teryokhina*[1], *A. A. Grusho*[2], *E. E. Timonina*[2], *and S. Ya. Shorgin*[2]**

[1]Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University, 1-52 Leninskie Gory, GSP-1, Moscow 119991, Russian Federation

[2]Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119133, Russian Federation

Abstract: The paper deals with the problem of "workflow mining." Workflow mining is numerous techniques for discovering processes' models represented by their workflow log. The paper considers process models in terms of simple Petri nets. It is shown that constructing a correct model when a process contains equal tasks is not always an attainable goal. Moreover, it was revealed that in the case when a model has transitions with no correspondence to any process task, the relation between the causal relations detected in the log and the presence of places connecting transitions in the Petri net is violated.

Keywords: Petri nets; workflow mining; process modeling

DOI: 10.14357/08696527200406

Acknowledgments

The paper was partially supported by RFBR, project No. 18-07-00274.

References

1. Agrawal, R., D. Gunopulos, and F. Leymann. 1998. Mining process models from workflow logs. *Advances in database technology*. Eds. H. J. Schek, G. Alonso, F. Saltor, and I. Ramos. Lecture notes in computer science ser. Springer. 1337:467–483.
2. Cook, J. E., and A. L. Wolf. 1998. Discovering models of software processes from event-based data. *ACM T. Softw. Eng. Meth.* 7(3):215–249.
3. Mannila, H., and D. Rusakov. 2001. Decomposition of event sequences into independent components. *SIAM Conference (International) on Data Mining Proceedings.* Society for Industrial and Applied Mathematics. 1–17.

4. Schimm, G. 2002. Process miner — a tool for mining process schemes from event-based data. *Logics in artificial intelligence*. Eds. S. Flesca, S. Greco, G. Ianni, and N. Leone. Lecture notes in computer science ser. Springer. 2424:525–528.
5. Herbst, J. 2004. *Ein induktiver ansatz zur akquisition und adaption von workflow-modellen*. Berlin: Tenea Verlag Ltd. 284 p.
6. Van der Aalst, W., T. Weijters, and L. Maruster. 2004. Workflow mining: Discovering process models from event logs. *IEEE T. Knowl. Data En.* 16(9):1128–1142.
7. Reisig, W., and G. Rozenberg, eds. 1998. *Lectures on Petri nets I: Basic models: Advances in Petri nets*. Berlin–Heidelberg: Springer-Verlag. 691 p.
8. Peterson, J. L. 1981. *Petri net theory and the modeling of systems*. Prentice-Hall. 290 p.

Received September 11, 2020

Contributors

Teryokhina Irina Yu. (b. 1994) — PhD student, Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University, 1-52 Leninskie Gory, GSP-1, Moscow 119991, Russian Federation; irina_teryokhina@mail.ru

Grusho Alexander A. (b. 1946) — Doctor of Science in physics and mathematics, professor, principal scientist, Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences; 44-2 Vavilov Str., Moscow 119133, Russian Federation; grusho@yandex.ru

Timonina Elena E. (b. 1952) — Doctor of Science in technology, professor, leading scientist, Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences; 44-2 Vavilov Str., Moscow 119133, Russian Federation; eltimon@yandex.ru

Shorgin Sergey Ya. (b. 1952) — Doctor of Science in physics and mathematics, professor, principal scientist, Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119133, Russian Federation; sshorgin@ipiran.ru