Модел. и анализ информ. систем. Т. **19**, № **4** (2012) 128–143

УДК 519.876.5

Оптимизация расчёта инвариантов сети Петри в рамках задачи формирования сценариев интеграционного тестирования

Доррер М.Г., Курохтин В.В.

Сибирский Государственный Технологический университет

e-mail: mdorrer@mail.ru, vitaliy.kurokhtin@gmail.com

получена 25 июля 2012

Ключевые слова: инвариант, сеть Петри, оптимизация, интеграционное тестирование

Рассматривается метод нахождения инвариантов модели бизнес-процесса построенной в нотации eEPC. Метод основан на преобразовании исходной модели в сеть Петри, расчёте инвариантов и дальнейшем обратном преобразовании полученных инвариантов сети Петри в модель eEPC. Предлагается оптимизированный метод нахождения инвариантов сети Петри, основанный на определении возможных значений отдельных элементов (и их групп) векторов инвариантов и их дальнейшей комбинации. Полученные инварианты бизнес-процесса могут быть использованы для построения сценариев интеграционного тестирования внедряемой на предприятии АСУ.

1. Введение (необходимость)

Непосредственным побудительным мотивом к выполнению предлагаемой работы явилась высокая трудоемкость формирования сценариев интеграционного тестирования при внедрении АСУ предприятий. Данная задача решалась в консалтинговой фирме, где в это время работал один из авторов. Проблема состоит в необходимости предусмотреть все возможные сценарии функционирования АСУ при построении интеграционного теста. Выполнение данной работы «вручную» трудоемко и чревато ошибками и неполнотой в сформированных сценариях тестирования. Разработанный ранее алгоритм трансляции бизнес-модели в сеть Петри (изложен в работе [3]) позволяет автоматизировать работу по составлению сценариев интеграционного тестирования.

Понятно, что при определенных (изложенных также в работе [3]) ограничениях на правила перевода событийной модели (ARIS eEPC, IDEF3) эквивалентная ей сеть Петри может быть использована для исследования топологии модели. В частности, при использовании специализированных фишек сети Петри – авторы называли их процессными маркерами – последовательность срабатываний переходов такой сети соответствует последовательности выполнения операций бизнес-модели. Очевидно, что, решив задачу нахождения всех инвариантов сети, сформированной таким образом, мы сможем получить и все возможные варианты прохождения бизнес-процесса, описанного изначальной событийной моделью. А следовательно, – и полный набор сценариев интеграционного тестирования АСУ, внедряемой на базе данной бизнес-модели.

Для анализа в данной работе использовалась модель событийно-управляемых цепочек процессов ARIS eEPC [5].

2. Термины и определения

Алгебраически обыкновенная сеть Петри N может быть представлена следующим образом ([2], [4]):

$$N = \{\theta, P, T, F, M(0)\},$$

где θ – дискретное время, $\theta = 0, 1, 2, ...$

$P = \{p_1, p_2, ..., p_n\}$ – множество узлов, называемых позициями сети;

$T = \{t_1, t_2, ..., t_m\}$ – множество узлов, называемых переходами сети;

$F = F^p \cup F^t$ – функция инцидентности;

$F^p = \|f^p_{ij}\|$ – $n \times m$ матрица, $F^t = \|f^t_{ji}\|$ – $m \times n$ матрица, $f^p_{ij} \geq 0$ – кратность дуги от p_i к t_j, $f^t_{ji} \geq 0$ – кратность дуги от t_i к p_j, $t = 1, 2, ..., n$, $j = 1, 2, ..., m$.

Каждая позиция p_i в момент времени θ может содержать целочисленный ресурс $m_i(\theta)$, совокупность ресурсов в позициях в момент времени θ определяет вектор маркировки:

$$M(\theta) = [m_1(\theta), m_2(\theta), ..., m_n(\theta)].$$

Начальная маркировка $M(0)$ определяет наличие ресурсов в позициях в момент запуска системы.

Функционирование сети Петри заключается в изменении маркировки $M(\theta)$ путём срабатывания переходов. Переход t_j может сработать (называется разрешённым) если:

$$m_i(\theta) \geq f^p_{ij}, i = 1, 2, ..., n.$$

При срабатывании перехода t_j происходит изменение маркировки по правилу:

$$m_i(\theta + 1) = m_i(\theta) - f^p_{ij} + f^t_{ji}, i = 1, 2, ..., n. \tag{1}$$

Пусть t_j – разрешённый переход в момент времени θ, Введём m-вектор $\tau_j(\theta) = [0, 0, ..., 1, ..., 0]$, в котором на j-м месте стоит единица, а все остальные элементы равны нулю. Тогда формула (1) может быть представлена в векторном виде:

$$M(\theta + 1) = M(\theta) + \tau_j(\theta)\Phi,$$

где $\Phi = \|f^t_{ji} - f^p_{ij}\| = (F^t)^T - F^p$ – $n \times m$ матрица разности ресурсов.

Для сети Петри определены понятия t- и p-инвариантов. t-инвариантом (инвариантом переходов) называется такое множество переходов $I^t \subset T$, которые, срабатывая последовательно, приводят маркировку сети к начальной. p-инвариантом (инвариантом позиций) называется такое множество позиций $I^p \subset P$, сумма ресурсов в которых неизменна на протяжении всего функционирования сети.

Известно, что инвариант переходов сети Петри определяется следующим образом [1]:

$$\Phi \times W = 0_m, \tag{2}$$

где W – ненулевой n-вектор, состоящий из нулей и единиц, единица в позиции j означает включение перехода t_j в инвариант, нуль – отсутствие оного в инварианте; 0_m – нулевой m-вектор столбец.

3. Алгоритм нахождения инвариантов

Для поиска инвариантов интеграционного тестирования преобразуем eEPC модель в сеть Петри при помощи метода описанного в [3], вычислим инварианты полученной сети и преобразуем их обратно в термины исходной модели.

Алгоритм трансляции модели eEPC в одноцветную сеть Петри описан в [3]. В данной статье используется его модификация, заключающаяся в замыкании исходной модели через два ИЛИ правила. Это позволяет рассчитать инварианты для линейных бизнес-процессов.

Необходимым условием наличия инвариантов сети Петри является положительное значение дефекта матрицы Φ [1], т.е. наличия в ней линейно зависимых строк и столбцов.

Алгоритм трансляции модели бизнес-процесса в одноцветную сеть Петри, описанный в [3], предполагает сопоставление перехода каждому активному элементу (событие и функция в нотации eEPC) и позиции – связям активных элементов. Что приводит к увеличению размера получаемой сети Петри (а соответственно и матрицы Φ) более чем в два раза по сравнению с размером исходной модели.

Первым шагом для расчета инвариантов является построение матрицы разности ресурсов – Φ. Для реальных моделей бизнес-процессов размерность матрицы может составлять многие десятки, а то и сотни строк и столбцов. Учитывая, что при трансляции бизнес-модели в сеть Петри количество составных элементов увеличивается, а также, что реальные модели процессов могут состоять из большого числа объектов с множеством логических операций, матрица Φ для реальных задач может иметь достаточно большую размерность. К примеру, для модели из 50 элементов и 25 правил матрица Φ будет иметь размерность близкую к 100×100.

В рамках поставленной задачи матрица Φ и вектор W обладают следующими свойствами:

1. Элементы матрицы Φ могут принимать только значения -1, 0 и 1. Данное свойство обусловлено алгоритмом трансляции, предполагающим создание ветвей с весом равным единице.

2. Матрица Ф является разреженной матрицей, ввиду того, что сеть Петри получена из модели бизнес-процесса, для которых свойственно малое число связей элементов модели по сравнению с количеством самих элементов в модели.

3. Элементы вектора W могут принимать только значения 0 и 1.

Однако, даже несмотря на эти ограничения, пространство возможных решений достаточно велико (2^n), что препятствует применению метода полного перебора. Ввиду этого для поиска инвариантов был применён алгоритм, приведённый на рис. 1.

Определение вариантов вектора W производится по признаку равенства нулю скалярного произведения данного вектора на анализируемую строку матрицы Ф. В силу специфики задачи матрица Ф сильно разрежена, соответственно в части элементов строки содержатся нули. Это означает, что на соответствующих позициях вектора W могут находиться любые значения. Таким образом, получим векторы, у которых часть значений определена, а часть – обозначена знаком вопроса. Полученные векторы назовем частичными.

Определение непротиворечивых сочетаний частичных векторов производится по следующему правилу: определённые значения одного частичного вектора равны или соответствуют неопределённым значениям другого.

Подобный подход не всегда позволяет получить конечное решение – получить из частичных векторов полные. После окончания обработки последней строки матрицы Ф в частичных векторах могут остаться неопределённые значения. Это говорит о том, что исходная матрица недоопределена. Однако, т.к. требуется найти только векторы из нулей и единиц, недостающие элементы можно определить методом полного перебора.

На основе полученных инвариантов определим соответствующие им наборы переходов, приводящих анализируемую сеть Петри к стартовой разметке. Проводя соответствие между каждым переходом и соответствующим ему символом, на модели бизнес-процесса легко построить соответствующий данному инварианту сценарий выполнения бизнес-процесса. Идентификация связей, входящих в инвариант бизнес-процесса, производится путём определения позиций, сопоставленных со связями бизнес-модели. Стоит заметить, что данный метод эффективен только на определённых входных данных, специфичных для рассматриваемого класса задач. Ввиду того, что количество связей в сети Петри, описывающей бизнес-процесс, сравнимо с количеством вершин, матрица приращения ресурсов имеет разреженный характер. Допустим, что в исходной матрице размерности $n \times m$ в каждой строке не более чем k ненулевых элементов, $k \leq m$.

Для каждой из n строк матрицы приращения ресурсов решается задача полного перебора по ненулевым элементам. Для каждого из 2^k вариантов выполняется k умножений и $k-1$ сложений:

$$T = n2^k(k + k - 1) = n2^k(2k - 1).$$

Для рассматриваемых задач k не зависит от n и m, таким образом, результирующая вычислительная сложность алгоритма имеет порядок $O(n)$. На более общих

Рис. 1. Алгоритм расчета инвариантов

входных данных, где значение k близко к n, вычислительная сложность алгоритма имеет порядок $O(n^2 2^n)$. К примеру, метод Гаусса имеет вычислительную сложность порядка $O(n^3)$ и более чем в два раза эффективен уже при $n = 3$.

4. Результат расчета

Рассмотрим алгоритм поиска инвариантов бизнес-процесса на примере. В качестве исходной модели возьмём диаграмму, описывающую простой процесс, которая приведена на рис. 2.

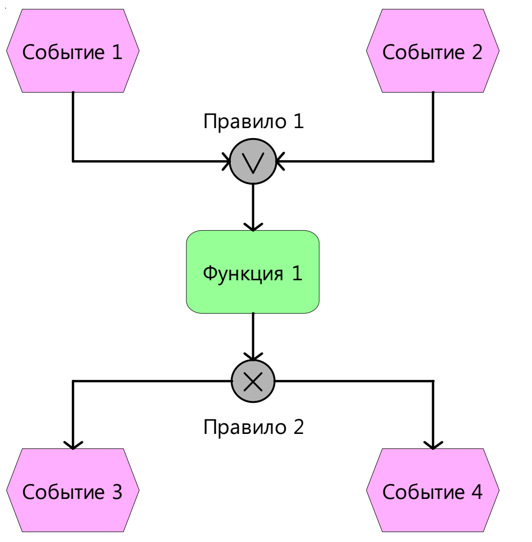

Рис. 2. Исходная диаграмма eEPC

В соответствии с этой модификацией, результирующая сеть Петри будет иметь следующий вид (рис. 3). Пунктиром обозначена область сети, создаваемая с целью замкнуть исходную модель бизнес-процесса. В рамках данной задачи прирост размерности, создаваемый подобным замыканием, выглядит значительным, однако на реальных данных он не оказывает особого влияния на сложность расчётов.

Матрицы инцидентности для приведённой сети имеют следующий вид:

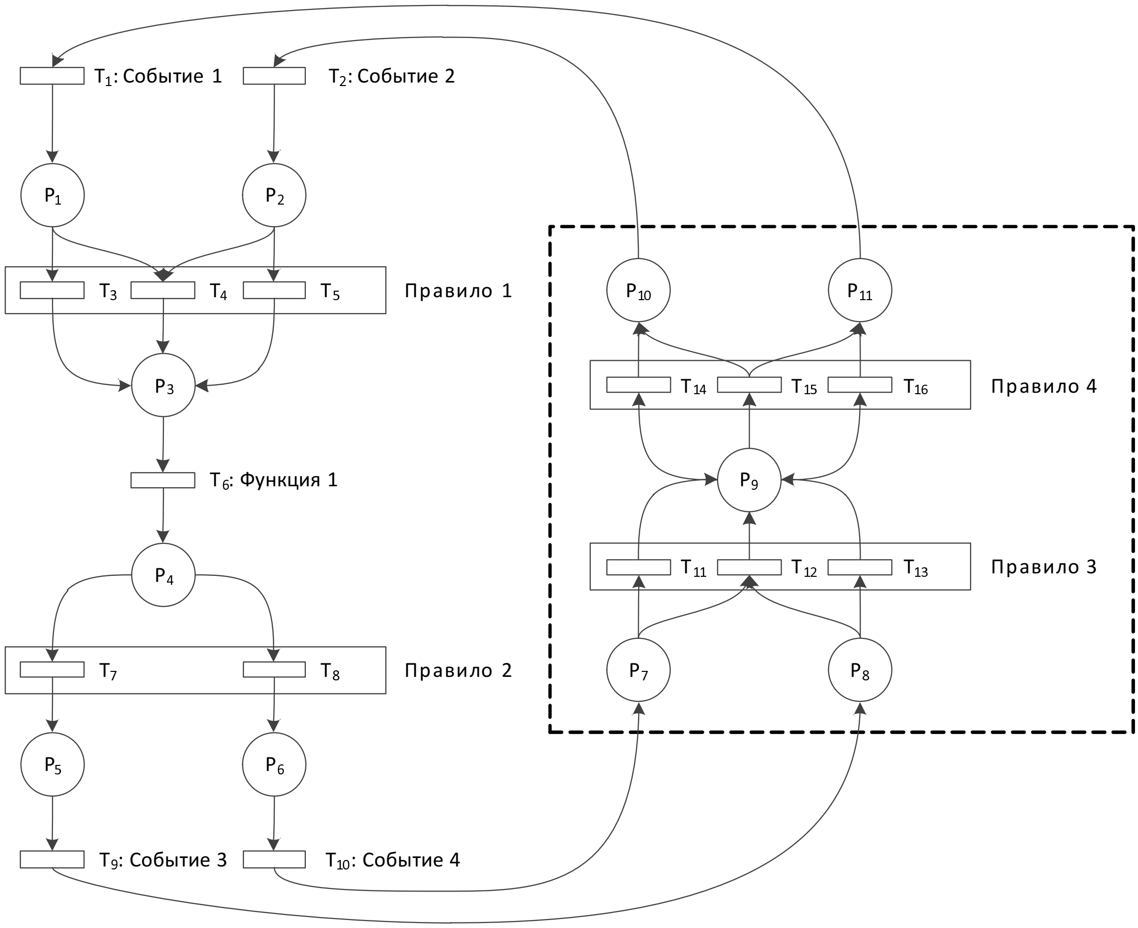

Рис. 3. Результирующая сеть Петри

$$\Phi^p = \begin{bmatrix}
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix},$$

$$\Phi^t = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}.$$

Вычислим матрицу разности ресурсов сети:

$$\Phi = \begin{bmatrix}
1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & -1 & -1 & -1 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1
\end{bmatrix}.$$

Определим дефект матрицы Φ. Для этого произведем расчет матрицы $\Phi^T \times \Phi$ и вычислим её дефект.

$$\Phi^T\times\Phi=\begin{bmatrix}
2&0&-1&-1&0&0&0&0&0&0&0&0&0&0&-1&-1\\
0&2&0&-1&-1&0&0&0&0&0&0&0&0&-1&-1&0\\
-1&0&2&2&1&-1&0&0&0&0&0&0&0&0&0&0\\
-1&-1&2&3&2&-1&0&0&0&0&0&0&0&0&0&0\\
0&-1&1&2&2&-1&0&0&0&0&0&0&0&0&0&0\\
0&0&-1&-1&-1&2&-1&-1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&-1&2&1&-1&0&0&0&0&0&0&0\\
0&0&0&0&0&-1&1&2&0&-1&0&0&0&0&0&0\\
0&0&0&0&0&0&-1&0&2&0&0&-1&-1&0&0&0\\
0&0&0&0&0&0&0&-1&0&2&-1&-1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&-1&2&2&1&-1&-1&-1\\
0&0&0&0&0&0&0&0&-1&-1&2&3&2&-1&-1&-1\\
0&0&0&0&0&0&0&0&-1&0&1&2&2&-1&-1&-1\\
0&-1&0&0&0&0&0&0&0&0&-1&-1&-1&2&2&1\\
-1&-1&0&0&0&0&0&0&0&0&-1&-1&-1&2&3&2\\
-1&0&0&0&0&0&0&0&0&0&-1&-1&-1&1&2&2
\end{bmatrix}.$$

Размерность полученной матрицы 16 на 16, при этом её ранг равен 11, из чего следует, что дефект матрицы отличен от нуля и равен 5, а значит, сеть, описываемая данной матрицей, имеет инварианты.

Определим возможные варианты вектора W. Для первой строки матрицы Φ определяются следующие частичные векторы:

$$W_0^{11}=\begin{bmatrix}0\\?\\0\\0\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\end{bmatrix},W_0^{12}=\begin{bmatrix}1\\?\\1\\0\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\end{bmatrix},W_0^{13}=\begin{bmatrix}1\\?\\0\\1\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\end{bmatrix}.$$

Повторим операцию для второй строки:

$$W_0^{21} = \begin{bmatrix} ? \\ 0 \\ ? \\ 0 \\ 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix}, W_0^{22} = \begin{bmatrix} ? \\ 1 \\ ? \\ 1 \\ 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix}, W_0^{23} = \begin{bmatrix} ? \\ 1 \\ ? \\ 0 \\ 1 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix}.$$

Сопоставляя каждый вектор из набора W_0^{11}, W_0^{12}, W_0^{13} с каждым вектором из набора W_0^{21}, W_0^{22}, W_0^{23}, определим непротиворечивые сочетания частичных векторов. Так, например, сочетание W_0^{11} и W_0^{21} непротиворечиво: определённые значения одного частичного вектора равны или соответствуют неопределённым значениям другого, в результате образуется более полный частичный вектор A_0:

$$W_0^{11} = \begin{bmatrix} 0 \\ ? \\ 0 \\ 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix}, W_0^{21} = \begin{bmatrix} ? \\ 0 \\ ? \\ 0 \\ 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix} \Rightarrow A_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix}.$$

Сочетание W_0^{11} и W_0^{22}, напротив, противоречиво, т.к. вектор W_0^{11} в позиции 4 содержит значение 0, в то время как вектор W_0^{22} содержит значение 1:

$$W_0^{11} = \begin{bmatrix} 0 \\ ? \\ 0 \\ \underline{0} \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix}, W_0^{22} = \begin{bmatrix} ? \\ 1 \\ ? \\ \underline{1} \\ 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix}.$$

Проверяя таким образом все возможные сочетания, определим, что W_0^{11} и W_0^{23}, W_0^{12} и W_0^{21}, W_0^{12} и W_0^{23}, W_0^{13} и W_0^{22} приводят к непротиворечивым частичным векторам A_1, A_2, A_3, A_4, а комбинации W_0^{12} и W_0^{22}, W_0^{13} и W_0^{21}, W_0^{13} и W_0^{23} противоречивы:

$$W_0^{11} = \begin{bmatrix} 0 \\ ? \\ 0 \\ 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix}, W_0^{23} = \begin{bmatrix} ? \\ 1 \\ ? \\ 0 \\ 1 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix} \Rightarrow A_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix}; \quad W_0^{12} = \begin{bmatrix} 1 \\ ? \\ 1 \\ 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix}, W_0^{21} = \begin{bmatrix} ? \\ 0 \\ ? \\ 0 \\ 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix} \Rightarrow A_2 = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix};$$

$$W_0^{12} = \begin{bmatrix} 1\\?\\1\\0\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\\? \end{bmatrix}, W_0^{23} = \begin{bmatrix} ?\\1\\?\\0\\1\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\\? \end{bmatrix} \Rightarrow A_3 = \begin{bmatrix} 1\\1\\1\\0\\1\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\\? \end{bmatrix}; \quad W_0^{13} = \begin{bmatrix} 1\\?\\0\\1\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\\? \end{bmatrix}, W_0^{22} = \begin{bmatrix} ?\\1\\?\\1\\0\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\\? \end{bmatrix} \Rightarrow A_4 = \begin{bmatrix} 1\\1\\1\\1\\0\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\\? \end{bmatrix};$$

$$W_0^{12} = \begin{bmatrix} 1\\?\\1\\\underline{0}\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\\? \end{bmatrix}, W_0^{22} = \begin{bmatrix} ?\\1\\?\\\underline{1}\\0\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\\? \end{bmatrix}; \quad W_0^{13} = \begin{bmatrix} 1\\?\\0\\\underline{1}\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\\? \end{bmatrix}, W_0^{21} = \begin{bmatrix} ?\\0\\?\\\underline{0}\\0\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\\? \end{bmatrix}; \quad W_0^{13} = \begin{bmatrix} 1\\?\\0\\\underline{1}\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\\? \end{bmatrix}, W_0^{23} = \begin{bmatrix} ?\\1\\?\\\underline{0}\\1\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\\? \end{bmatrix}.$$

В результате анализа первых двух строк матрицы R определяем векторы A_0, A_1, A_2, A_3 и A_4:

$$A_0 = \begin{bmatrix} 0\\0\\0\\0\\0\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\\? \end{bmatrix}, \quad A_1 = \begin{bmatrix} 0\\1\\0\\0\\1\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\\? \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1\\0\\1\\0\\0\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\\? \end{bmatrix}, \quad A_3 = \begin{bmatrix} 1\\1\\1\\0\\1\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\\? \end{bmatrix}, \quad A_4 = \begin{bmatrix} 1\\1\\1\\1\\0\\?\\?\\?\\?\\?\\?\\?\\?\\?\\?\\? \end{bmatrix}.$$

Далее процесс продолжается с определения возможных вариантов вектора W на основе третьей строки и поиска непротиворечивых сочетаний с векторами A_0, A_1, A_2, A_3 и A_4. В результате, после обработки последней строки матрицы Φ, получим следующие векторы инварианты:

$$W_0 = \begin{bmatrix} 1\\0\\1\\0\\0\\1\\1\\0\\1\\0\\0\\0\\1\\0\\0\\1 \end{bmatrix}, \quad W_1 = \begin{bmatrix} 1\\0\\1\\0\\0\\1\\0\\1\\0\\1\\1\\0\\0\\0\\0\\1 \end{bmatrix}, \quad W_2 = \begin{bmatrix} 1\\1\\0\\1\\0\\1\\1\\0\\1\\0\\0\\0\\1\\0\\1\\0 \end{bmatrix}, \quad W_3 = \begin{bmatrix} 1\\1\\0\\1\\0\\1\\0\\1\\0\\1\\1\\0\\0\\0\\1\\0 \end{bmatrix}, \quad W_4 = \begin{bmatrix} 0\\1\\0\\0\\1\\1\\1\\0\\1\\0\\0\\0\\1\\1\\0\\0 \end{bmatrix}, \quad W_5 = \begin{bmatrix} 0\\1\\0\\0\\1\\1\\0\\1\\0\\1\\1\\0\\0\\1\\0\\0 \end{bmatrix}.$$

На основе полученных инвариантов определим наборы переходов (единица в значении элемента означает, что переход включается в инвариант, нуль – не включается):

$$\begin{aligned}
W_0 &\Rightarrow (T_1, T_3, T_6, T_7, T_9, T_{13}, T_{16}),\\
W_1 &\Rightarrow (T_1, T_3, T_6, T_8, T_{10}, T_{11}, T_{16}),\\
W_2 &\Rightarrow (T_1, T_2, T_4, T_6, T_7, T_9, T_{13}, T_{15}),\\
W_3 &\Rightarrow (T_1, T_2, T_4, T_6, T_8, T_{10}, T_{11}, T_{15}),\\
W_4 &\Rightarrow (T_2, T_5, T_6, T_7, T_9, T_{13}, T_{14}),\\
W_5 &\Rightarrow (T_2, T_5, T_6, T_8, T_{10}, T_{11}, T_{14}).
\end{aligned}$$

Анализ связей показывает, что для инварианта W_0 включёнными позициями будут P_1, P_3, P_4, P_5, P_8, P_9 и P_{11}. Аналогично определяются позиции и для других инвариантов.

Результирующие инварианты исследуемого бизнес-процесса будут иметь следующий вид (рис. 4):

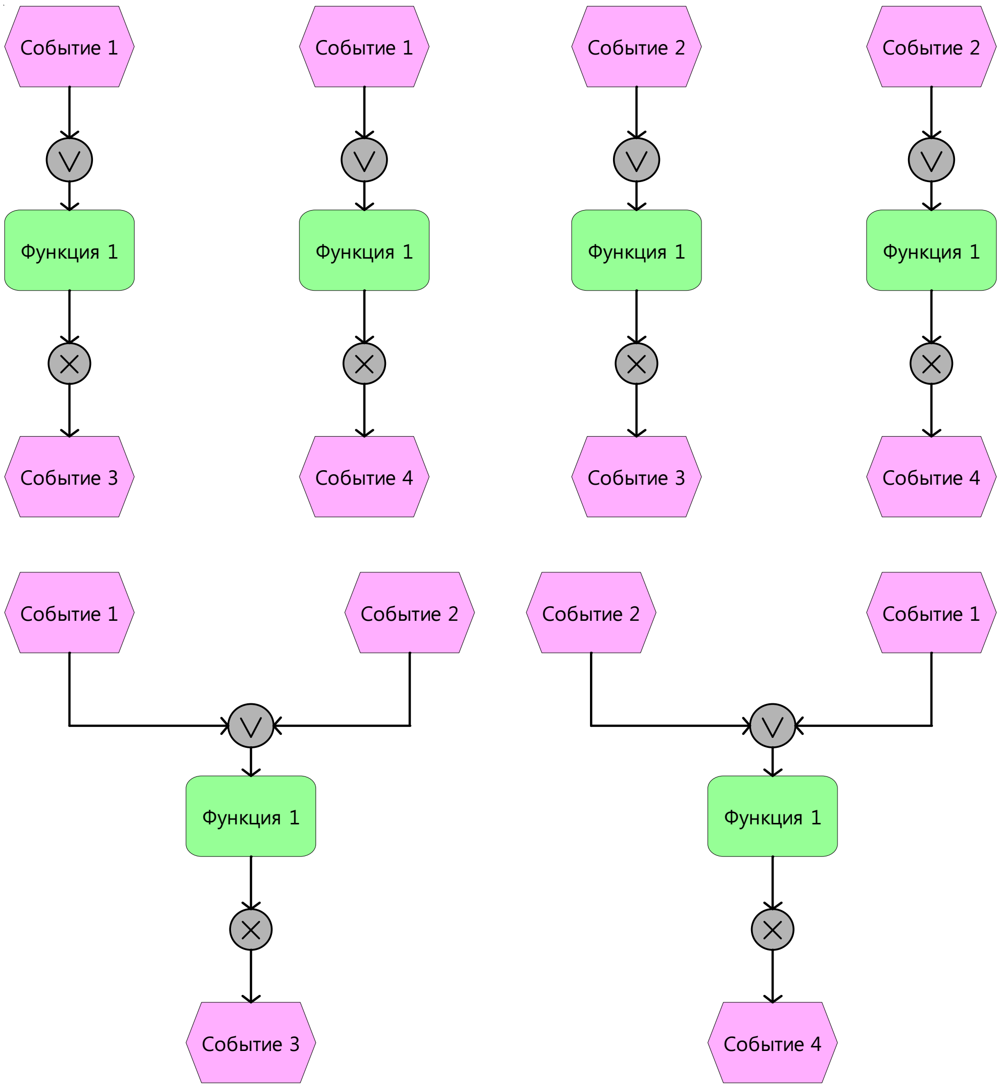

Рис. 4. Инварианты исходного бизнес-процесса

5. Заключение

Таким образом, предложен алгоритм расчёта инвариантов сети Петри, эффективный на задачах больших размерностей. Алгоритм реализован в виде программного средства, обеспечивающего генерацию и запись сценариев интеграционного тестирования АСУ на базе исходной бизнес-модели в нотации ARIS eEPC. Данный алгоритм показал высокую скорость работы – сценарии для моделей реально применяющихся на практике размерностей (20 – 30 элементов на модели) рассчитываются за время, ограничивающееся единицами минут. Таким образом, помимо научной значимости результат обладает и чисто практической значимостью, снижая трудоемкость разработки сценариев интеграционного тестирования и исключая риск неполноты полученных тестов.

Список литературы

1. Jensen Kurt. Coloured Petri Nets – Basic Concepts, Analysis Methods and Practical Use. Vol. 2: Analysis Methods. Berlin; Heidelberg; New York: Springler, 1997.

2. Доррер Г.А. Моделирование вычислительных систем: Учебное пособие для студентов направлений 552800 и 654600. Красноярск: СибГТУ, 2003. 188 с.

3. Доррер М.Г. Алгоритм преобразования моделей бизнесс-процессов в одноцветные сети Петри // Моделирование и анализ информационных систем. 2010. Т. 17, №2. С. 5—16.

4. Котов В.Е. Сети Петри. М.:Наука,1984.

5. Шеер А.-В. Бизнес-процессы. Основные понятия. Теория. Методы. М.: Весть-МетаТехнология, 1999. 182 с.

Optimization of Calculating Invariants in Petri Nets to Support the Creation of Integration Testing Scenarios

Dorrer M.G., Kurokhtin V.V.

Keywords: invariant, Petri net, optimization, integration tesing

The article describes a method of finding business process invariants basing on a given model in eEPC notation. The method uses an original translation process to build a Petri net corresponding to the source eEPC model, to find its invariants and to translate them back into the eEPC notation. An optimized method of finding Petri net invariants is also offered, based on estimating possible values for separate vector elements (and a group of elements) and combining these values with each other to receive a Petri net invariant. The resulting business process invariants may be used to create integration testing scenarios for an implemented automation system.

Сведения об авторах:
Доррер Михаил Георгиевич,
Сибирский Государственный Технологический университет,
канд. техн. наук, доцент каф. системотехники;
Курохтин Виталий Валерьевич,
Сибирский Государственный Технологический университет,
аспирант каф. системотехники