Российская академия наук
Сибирское отделение
Институт систем информатики
имени А. П. Ершова

Александр В. Быстров

СТРУКТУРНЫЙ АНАЛИЗ ПОВЕДЕНИЯ НЕПРЕРЫВНО-ВРЕМЕННЫХ СЕТЕЙ ПЕТРИ

Препринт
137

Новосибирск 2006

В данной работе выделяются подклассы непрерывно-временных сетей Петри, для которых разрабатываются эффективные структурные алгоритмы анализа таких поведенческих свойств, как живости, дивергентности, ограниченности, безопасности. Разработанные алгоритмы реализованы в рамках экспериментальной системы RT-MEC (Real Time Model and Equivalence Checker), поддерживающей спецификацию, анализ, валидацию и верификацию распределенных систем реального времени, представленных различными моделями временных сетей Петри.

Siberian Division of the Russian Academy of Sciences
A. P. Ershov Institute of Informatics Systems

Alexander V. Bystrov

STRUCTURAL ANALYSIS OF DENSE-TIME PETRI NETS BEHAVIOUR

Preprint
137

Novosibirsk 2006

The intention of the work is to develop structural algorithms to effectively analyze the behavioural properties (e.g., liveness, divergency, boundedness, safety) of different subclasses of dense-time Petri nets. The algorithms are implemented within the RT-MEC system (Real Time Model and Equivalence Checker) supporting specification, analysis, validation and verification of distributed and real-time systems represented by various timed PN-based models.

1. ВВЕДЕНИЕ

Последнее десятилетие резко возрос интерес к разработке и исследованию параллельных/распределенных систем, функционирующих в режиме реального времени. В литературе системы такого типа часто представляются временными автоматами, содержащими конечное множество часов [1], временными системами переходов [8] и алгебрами временных процессов (например, [12]). Однако все эти формализмы базируются на интерливинговой семантике и поэтому не позволяют моделировать параллелизм напрямую.

Особое место среди временных моделей с семантикой “истинного параллелизма” занимают временные сети Петри. Впервые временное расширение сетей Петри было предложено Рамхандани [10]. Существует ряд способов введения понятия времени в модель сетей Петри. Время может быть сопоставлено различным сетевым элементам: местам, переходам, дугам, фишкам, шагам (множествам параллельных переходов). При этом различают временные ограничения, сопоставленные некоторому элементу временной сети Петри, и часы, введенные в модель для контроля локального или глобального времени. Временная информация может быть представлена как одним числом (что соответствует дискретному представлению времени), так и интервалом (что соответствует непрерывному представлению времени). В модели либо разрешается только одиночное срабатывание перехода, либо предполагается принцип максимального шага срабатывания (т.е. одновременного срабатывания максимально возможного количества параллельных переходов). Согласно работам [2, 7] по сравнению с другими моделями временных сетей непрерывно-временные сети Петри (НВСП)[9] являются наиболее удобной и выразительной моделью для описания и изучения параллельных систем, функционирующих в режиме реального времени. В модели НВСП каждому переходу сопоставляется пара неотрицательных чисел, которые указывают относительно локальных часов наиболее раннее и наиболее позднее времена, когда может сработать переход, при этом, срабатывание перехода мгновенно. Однако, как было показано в работе [2], число состояний даже ограниченной НВСП бесконечно и проблемы анализа поведенческих свойств НВСП алгоритмически неразрешимы.

В данной работе делается попытка выделить подклассы НВСП, для которых можно разработать эффективные структурные алгоритмы анализа таких поведенческих свойств, как живости, дивергентности, ограниченности, безопасности. Разработанные алгоритмы реализованы в рамках экспериментальной системы RT-MEC (Real Time Model and Equivalence Checker), поддерживающей спецификацию, анализ, валидацию и верификацию распределенных систем реального времени, представленных различными моделями временных сетей Петри. RT-MEC функционирует в составе системы PEP (Programming Environment based on Petri nets)[3].

Работа организована следующим образом. В следующем разделе определяется ряд понятий, связанных со структурой и поведением НВСП. Далее, на основе полученных теоретических результатов разрабатываются структурные алгоритмы анализа некоторых поведенческих свойств НВСП, показывается корректность и дается оценка временной сложности данных алгоритмов. В приложении приводятся определения и классические результаты по анализу поведенческих свойств СП со свободным выбором.

2. НЕПРЕРЫВНО-ВРЕМЕННЫЕ СП

Рассмотрим ряд понятий, связанных со структурой и поведением НВСП. Под НВСП понимается сеть Петри, где с каждым переходом связан некоторый временной интервал задержки (относительно локальных часов) срабатывания перехода.

Пусть $\mathbf{N}$ — множество натуральных чисел, $\mathbf{R}_+$ — множество положительных действительных чисел и $\mathbf{R}$ — множество неотрицательных действительных чисел. Определим множество $Interv = \{[d_1, d_2] \subset \mathbf{R} \mid d_1, d_2 \in\ \mathbf{N},\ d_1 \leq d_2\}$.

Определение 2.1. НВСП *— это кортеж* $\mathcal{TN} = (P,\ T,\ F,\ M_0,\ D)$, *где* $\mathcal{N} = (P, T, F, M_0)$ *— ординарная СП;* $D : T \to Interv$ *— функция, сопоставляющая каждому переходу* $t \in T$ *интервал* $D(t)$ *из множества* $Interv$.

Через $Eft(t)$ будем обозначать нижнюю временную границу $\min D(t)$ перехода t, а через $Lft(t)$ его верхнюю границу $\max D(t)$.

Пример НВСП приведен на рис. 1, где с каждым переходом связан его временной интервал.

Переход $t \in T$ *может сработать (готов к срабатыванию) при разметке* M, если ${}^\bullet F(t) \leqslant M$. Обозначим через $En(M)$ множество переходов, готовых к срабатыванию при разметке M. Пусть $\mathcal{V}(\mathcal{TN}) = [T \to \mathbf{R}_+]$ — множество *наборов значений часов*, связанных с переходами из

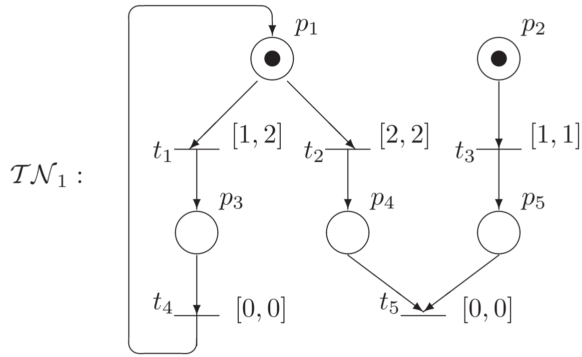

Рис. 1.

T. Для заданных $\nu \in \mathcal{V}(\mathcal{TN})$ и $\delta \in \mathbf{R}$ обозначим через $\nu + \delta$ набор значений часов, равных $\nu(t) + \delta$ для каждого перехода t из T.

Состоянием в $\mathcal{TN}$ будем называть пару $q = \langle M, \nu \rangle$, где M — разметка и $\nu \in \mathcal{V}(\mathcal{TN})$. Тогда *начальное состояние* в $\mathcal{TN}$ — это пара $q_0 = \langle M_0, \nu_0 \rangle$ такая, что $\nu_0(t) = 0$ для всех $t \in T$.

В НВСП смена одного состояния другим осуществляется либо по истечении некоторого времени, либо при срабатывании некоторого перехода сети.

Будем говорить, что в состоянии $q = \langle M, \nu \rangle$ *может пройти время* $\delta \in \mathbf{R}_+$, если существует $d' \geqslant d$ такое, что $\nu(t) + d' \in D(t)$ для всех $t \in En(M)$. Тогда состояние $q' = \langle M', \nu' \rangle$ получается *по истечении времени* δ из состояния q (обозначаем $q \xrightarrow{\delta} q'$), если $M' = M$ и $\nu'(t) = \nu(t) + \delta$ для всех $t \in T$.

Будем говорить, что в состоянии $q = \langle M, \nu \rangle$ переход $t \in T$ *может сработать*, если $t \in En(M)$ и $\nu(t) \in D(t)$. Тогда состояние $q' = \langle M', \nu' \rangle$ получается *при срабатывании перехода* t из состояния q (обозначаем $q \xrightarrow{0} q'$), если $M' = M -^{\bullet} F(t) + F(t)^{\bullet}$ и для каждого $t' \in T$ верно:

$$\nu'(t') = \begin{cases} 0, \text{ если } t' \in En(M') \setminus En(M), \\ \nu(t'), \text{ иначе.} \end{cases}$$

В случаях, когда существенно, что q' получается из q при срабатывании конкретного перехода t, будем писать $q \xrightarrow{t} q'$.

Состояние q является *достижимым*, если $q = q_0$ или существует достижимое состояние q' такое, что $q' \xrightarrow{\delta} q$ для некоторого $\delta \in \mathbf{R}$. Обозначим через $RS(\mathcal{TN})$ множество всех достижимых состояний в $\mathcal{TN}$. Пусть $RS(q)$ — множество состояний, достижимых из состояния q.

Назовем *q-путем* r в $\mathcal{TN}$ последовательность состояний, переходов и моментов времени вида: $q \xrightarrow{\delta_1} q_2 \xrightarrow{\delta_2} \dots q_{n-1} \xrightarrow{\delta_n} q_n$. В таком случае будем писать $q\ [\rho = \delta_1 \dots \delta_n > q_n$. Если ρ бесконечна, будем писать $q\ [\rho >$. Введем проекцию ρ на T (обозначается $trans(\rho)$), полученную вычеркиванием из ρ всех $\delta_i \in \mathbf{R}_+$. Проекция содержит только информацию о срабатывании переходов.

Рассмотрим в НВСП $\mathcal{TN} = (P, T, F, M_0, D)$ ситуацию, когда ${}^\bullet t_1 \cap {}^\bullet t_2 \neq \emptyset$. Очевидно, для того чтобы каждый из переходов t_1 и t_2 (в случае готовности к срабатыванию при некоторой разметке) имел возможность сработать, необходимо выполнение условия: $D(t_1) \cap D(t_2) \neq \emptyset$. Назовем это условие *корректным таймированием*.

При моделировании какой-либо реальной системы с помощью НВСП переходы сети можно разбить на два множества: внутренние и внешние. Внешние переходы соответствуют событиям в реальной системе, которые видны стороннему наблюдателю, а внутренние переходы — событиям, скрытым от стороннего наблюдателя. НВСП называется *дивергентной*, если существует достижимое состояние q такое, что $q\ [\rho >$, где $trans(\rho)$ — бесконечная последовательность срабатываний внутренних переходов. Дивергентная система является плохо сконструированной, так как в ней возможна бесконечная бесполезная работа.

Для НВСП свойства безопасности, ограниченности и живости определяются очевидным образом.

НВСП $\mathcal{TN} = (\mathcal{N}, M_0)$ называется

- *k-ограниченной* (*ограниченной*), если существует натуральное число k такое, что для любого достижимого состояния $\langle M, \nu \rangle$ и для любого места p верно $M(p) \leqslant k$. Если НВСП 1-ограничена, то ее называют *безопасной*;
- *живой*, если для любого перехода t и для любого достижимого состояния $\langle M, \nu \rangle$ существует достижимое из $\langle M, \nu \rangle$ состояние $\langle M', \nu' \rangle$, в котором переход t может сработать;
- *НВСП со свободным выбором* (*непрерывно-временная свободная СП*) (НВССП), если $\mathcal{N}$ — ССП (см. приложение).

3. СТРУКТУРНЫЙ АНАЛИЗ НВСП

3.1. Теоретическое обоснование

Сначала докажем ряд фактов, которые будут полезны в дальнейшем.

Лемма 3.1. *Пусть $\mathcal{N} = (P, T, F, M_0)$ — СП и $M_0\ [\tau >\ M$ в $\mathcal{N}$. Тогда $q_0 = (M_0, \nu_0)\ [\rho >\ q = (M, \nu)$ для некоторого $\nu \in \mathcal{V}(\mathcal{TN})$ и некоторой ρ такой, что $trans(\rho) = \tau$, в НВСП $\mathcal{TN} = (\mathcal{N}, D)$ при любом корректном таймировании D.*

Доказательство. Без потери общности предполагаем, что $\tau = t_1 \dots t_n$. Будем доказывать индукцией по n.

$n = 0$. Очевидно.

$n > 0$. По индукционному предположению, существует $\rho' = \delta_1 \dots \delta_m$ такая, что $q_0[\rho' > q' = \langle M_{n-1}, \nu' \rangle$ и $trans(\rho') = t_1 \dots t_{n-1}$. Это значит, что $t_n \in En(M_{n-1})$. Тогда $\nu'(t_n) \geqslant 0$. Исходя из правил смены состояний НВСП, только два случая возможны. Если $Eft(t_n) \leqslant \nu'(t_n) \leqslant Lft(t_n)$, то верно $\rho = \rho' t_n$. Если $\nu'(t_n) < Eft(t_n)$, то, согласно корректному таймированию, существует δ_n такое, что $\delta_n \in \bigcap_{t' \in En(M')} D(t')$, тогда верно $\rho = \rho' \delta_n \tau_n$. ♢

Лемма 3.2. *Пусть $\mathcal{TN} = (P, T, F, M_0, D)$ — НВСП и $q_0 = (M_0, \nu_0)$ $[\rho > q = (M, \nu)$. Тогда $M_0[trans(\rho) > M$ в СП $\mathcal{N} = (P, T, F, M_0)$.*

Доказательство. Пусть $q_0 = (M_0, \nu_0) \xrightarrow{\delta_1} \dots \xrightarrow{\delta_n} (M_n, \nu_n) = (M, \nu)$. Будем доказывать индукцией по n, что $M_0[trans(\delta_1 \dots \delta_n) > M_n$ в $\mathcal{N}$.

$n = 0$. Очевидно.

$n > 0$. Согласно индукционному предположению, имеем, что $M_0\ [trans(\delta_1\ \dots\ \delta_{n-1}) > M_{n-1}$ в $\mathcal{N}$. Возможны два случая:

$\delta_n \in \mathbf{R}_+$ тогда $M_{n-1} = M_n$, значит, $M_0[trans(\delta_1 \dots \delta_n) > M_n$ в $\mathcal{N}$;

$\delta_n \in T$ без потери общности предполагает $\delta_n = t$. Тогда $M' = M - {}^\bullet F(t) + F(t)^\bullet$. Значит, $M_0[trans(\delta_1 \dots \delta_n) > M_n$ в $\mathcal{N}$. ♢

Теорема 3.1. *Пусть $\mathcal{N} = (P, T, F, M_0)$ — ограниченная (безопасная) СП. Тогда $\mathcal{TN} = (\mathcal{N}, D)$ — ограниченная (безопасная) НВСП при любом таймировании D.*

Доказательство. Следует из леммы 3.2. ♢

Теорема 3.2. *Пусть $\mathcal{TN} = (P, T, F, M_0, D)$ — ограниченная (безопасная) НВСП с корректным таймированием. Тогда $\mathcal{N} = (P, T, F, M_0)$ — ограниченная (безопасная) СП.*

Доказательство. Следует из леммы 3.1. $\diamondsuit$

Теорема 3.3. *Пусть* $\mathcal{N} = (P, T, F, M_0)$ *— живая СП, удовлетворяющая следующим условиям:*

- $\forall p \in P \diamond |p^\bullet| \geqslant 1$,
- $\forall p, q \in P \diamond p^\bullet \cap q^\bullet \neq \emptyset \quad \Rightarrow \quad p^\bullet \subseteq q^\bullet \vee q^\bullet \subseteq p^\bullet$.

Тогда $\mathcal{TN} = (P, T, F, M_0, D)$ *— живая НВСП при любом корректном таймировании.*

Доказательство. Предположим, что существует корректное таймирование D такое, что $\mathcal{TN}$ — не живая НВСП, т.е. найдутся переход $t \in T$ и состояние $q \in RS(q_0)$ такие, что для всех состояний $q' = (M', \nu')$, достижимых из q, верно: $\exists\ p \in {}^\bullet t \ \diamond\ M'(p) < 1 \vee \nu'(t) \notin D(t)$.

Введем дополнительные обозначения. Пусть $q \in RS(q_0)$ и $Dead(q) = \{t \in T \mid \forall q' \in RS(q) \diamond \exists p \in {}^\bullet t$ такое, что $M'(p) < 1 \ \vee \ \nu'(t) \notin D(t)\}$. Очевидно, что для любого $q' \in RS(q)$ верно, что $Dead(q) \subseteq Dead(q')$. Определим множество $Max = \{q \in RS(q_0) \mid \forall q' \in RS(q) \diamond Dead(q) = Dead(q') \neq \emptyset\}$. Так как, по предположению, $\mathcal{TN}$ — не живая НВСП, то найдется $q \in RS(q_0)$ такое, что $Dead(q) \neq \emptyset$. Из сказанного выше следует, что $Max \neq \emptyset$.

Лемма А. Если $q \in Max$ и $t \in Dead(q)$, то для каждого $q' \in RS(q)$ существует место $p \in {}^\bullet t$ такое, что $M'(p) < 1$.

Доказательство. Предположим обратное, т.е. $\exists\ q' \in RS(q)$ такое, что $\forall p \in {}^\bullet t \diamond M'(p) \geqslant 1$, что означает $t \in En(M')$. Но тогда, по определению, имеем $\nu'(t) \leqslant Lft(t)$. Из условия корректного таймирования следует, что существуют δ и q'' такие, что $q' \xrightarrow{\delta} q''$ и $\nu''(t) = \nu'(t) + \delta \in D(t)$. Следовательно, верно $t \notin Dead(q'') = Dead(q)$. Пришли к противоречию. $\diamondsuit$

Из Леммы А следует, что если $t \in Dead(q)$, то ${}^\bullet t = \{p_1, p_2, ..., p_n\}$, где $n \geqslant 1$. Согласно второму пункту условия теоремы, имеем, что $p_1^\bullet \subseteq ... \subseteq p_n^\bullet$.

Лемма Б. Пусть $q = (M, \nu) \in Max, q' = (M', \nu') \in RS(q)$ и $1 \leqslant i \leqslant n$. Тогда если $M'(p_i) < 1$ и $M'(p_j) \geqslant 1$ для всех $1 \leqslant j \leqslant i$, то $p_j^\bullet \subseteq Dead(q')$.

Доказательство. Предположим обратное, т.е. существует $t^* \in p_i^\bullet$ такой, что $t^* \notin Dead(q')$. Напомним, что $p_1^\bullet \subseteq ... \subseteq p_n^\bullet, Dead(q) = Dead(q') \neq \emptyset$. Тогда можно найти состояние $q^* = (M^*, \nu^*) \in RS(q')$ такое, что $M^*(p_i) \geqslant 1,\ \ldots,\ M^*(p_n) \geqslant 1$ и $\nu(t^*) \in D(t^*)$.

Рассмотрим последовательность $q' = q_0 \xrightarrow{\delta_1} q_1 ... \xrightarrow{\delta_k} q_k = q^*$ с наименьшим k. Покажем, что $M_l(p_j) \geqslant 1$ для всех $1 \leqslant j \leqslant i - 1$ и $0 \leqslant l \leqslant k$. Предположим обратное, т.е. существуют некоторые j и l такие, что

$M_l(p_j) < 1$. Это означает, что существует переход t^{**} такой, что верно: $p_j \in {}^\bullet t^{**}$, $\forall p \in^\bullet t^{**} \diamond M_r(p) \geqslant 1$ и $\nu_r(t^{**}) \in D(t^{**})$ для некоторого $0 \leqslant r \leqslant l$. (Должен сработать некоторый переход t^{**}, чтобы убрать фишки из места p_j). Так как $t^{**} \in p_j^\bullet \subseteq \ldots \subseteq p_i^\bullet \subseteq \ldots \subseteq p_n^\bullet$, то $M_r(p_j) \geqslant 1, \ldots, M_r(p_i) \geqslant 1, \ldots, M_r(p_n) \geqslant 1$. Получили противоречие с условием минимальности k. Таким образом, не существует $q^* \in RS(q')$ такого, что для всех $1 \leqslant i \leqslant n$ верно, что $M^*(p_1) \geqslant 1, \ldots, M^*(p_n) \geqslant 1$. Тогда, в силу Леммы А, имеем, что $t^* \in Dead(q')$. Пришли к противоречию. $\diamondsuit$

Лемма В. Пусть $q \in Max, t \in Dead(q)({}^\bullet t = \{p_1, \ldots, p_n\}, n \geqslant 1)$. Тогда существует j $(1 \leqslant j \leqslant n)$ такое, что $M'(p_j) < 1$ для всех $q' = (M', \nu') \in RS(q)$.

Доказательство. Предположим обратное, т.е. для всех $1 \leqslant j \leqslant n$ существует $q'_j \in RS(q)$ такое, что $M'_j(p_j) \geqslant 1$. Покажем индукцией по j, что для всех $1 \leqslant j \leqslant n$ существует $q_j \in RS(q)$ такое, что $M_j(p_j) \geqslant 1$. Для $j = 1$ полагаем $q_j = q'_j$. Для $j > 1$ существует $q_{j-1} = (M_{j-1}, \nu_{j-1}) \in RS(q)$ такое, что $M_{j-1}(p_1) \geqslant 1, \ldots, M_{j-1}(p_{j-1}) \geqslant 1$. Далее рассмотрим два случая. Если $M_{j-1}(p_j) \geqslant 1$, то полагаем $q_j = q_{j-1}$. Если $M_{j-1}(p_j) \geqslant 1$, то по Лемме Б, получаем, что $p_j^\bullet \subseteq Dead(q_{j-1})$. Тогда верно $p_1^\bullet \subseteq \ldots \subseteq p_j^\bullet \subseteq Dead(q_{j-1}) = Dead(q)$. Это означает, что ни один переход, уменьшающий количество фишек в местах $p_1, \ldots, p_j$ не может сработать в состояниях $q_1, \ldots, q_{j-1}$ и $M_{j-1}(p_j) < 1$. По предположению, существует q'_j такое, что $M'_j(p_j) \geqslant 1$. Очевидно, что $q'_j \in RS(q_{j-1}$ и $M'_j(p_k) \geqslant 1$ для всех $1 \leqslant k \leqslant j$.

Таким образом, получаем, что существует $q_n \in RS(q)$ такое, что $M_n(p_j) \geqslant 1$ для всех $1 \leqslant j \leqslant n$. Тогда, по Лемме А, имеем $t \notin Dead(q_n) = Dead(q)$. Пришли к противоречию. $\diamondsuit$

Определим множество $P_0(q) = \{p \mid \forall q' = (M', \nu') \in RS(q) \diamond M'(p) < 1\}$, где $q \in Max$. Возьмем $t \in Dead(q)$. В силу леммы Б, существует $p \in {}^\bullet t$ такое, что $M'(p) < 1$ для всех $q' = (M', \nu') \in RS(q)$. Следовательно, верно $p \in P_0(q)$, т.е. $P_0(q) \neq \emptyset$.

Рассуждая аналогичным образом для произвольного $t' \in Dead(q)$ получаем, что $Dead(q) \subseteq P_0^\bullet$. С другой стороны, возьмем произвольный $t' \in P_0^\bullet(q)$ (такой переход t' существует, в силу первого пункта условия теоремы). Тогда $M'(p) < 1$ для всех $q' = (M', \nu') \in RS(q)$, что означает $t \in Dead(q)$. Следовательно, верно $Dead(q) = P_0^\bullet(q)$. Если существует $t \in {}^\bullet P_0(q)$, то очевидно, что $t \in Dead(q)$ (иначе существует состояние, при котором t срабатывает и в $P_0(q)$ появляются фишки).

Значит, верно, что ${}^{\bullet}P_0(q) \subseteq Dead(q) = P_0^{\bullet}$, т.е. $P_0(q)$ — статический тупик в $\mathcal{N}$. ♢

Теорема 3.4. *Пусть $\mathcal{TN} = (P, T, F, M_0, D)$ — живая НВСП с корректным таймированием. Тогда $\mathcal{N} = (P, T, F, M_0)$ — живая СП.*

Доказательство. Следует из леммы 3.1. ♢

Теорема 3.5. *Пусть $\mathcal{TN} = (\mathcal{N}, D)$ — живая и ограниченная НВССП с корректном таймированием D, T_I и T_O — разбиение переходов сети на внутренние и внешние переходы соответственно. $\mathcal{TN}$ дивергентна по отношению к этому разбиению, если существует T-компонента (P_1, T_1, F_1) сети такая, что $T_1 \subseteq T_I$.*

Доказательство. Следует из леммы 3.2, 3.4 и теоремы А.10. ♢

3.2. Анализ живости НВСП

Алгоритм А1 осуществляет анализ живости ограниченных непрерывно-временных сетей со свободным выбором (НВССП).

ВХОД: Ограниченная НВССП.

1. Если ¬(НВССП корректно таймирована), то Печать “НВССП не корректно таймирована” и переход на ВЫХОД.
2. Если ¬(каждый минимальный тупик размечен), то Печать “Не живая НВССП” и переход на ВЫХОД.
3. Если ¬(каждый минимальный тупик является ловушкой), то Печать “Не живая НВССП” и переход на ВЫХОД, иначе Печать “Живая НВССП”.

ВЫХОД

На первом шаге проверяется корректное таймирование НВССП. Для каждого места такого, что $|p^{\bullet}| \geqslant 2$, осуществляется проверка на пересечение временных интервалов $D(t_i)$ и $D(t_j)$ для всех переходов $t_i, t_j \in |p^{\bullet}|$. В случае обнаружения, что для некоторых переходов $t_i, t_j \in |p^{\bullet}|$ пересечение их временных интервалов $D(t_i)$ и $D(t_j)$ пусто, работа алгоритма останавливается и печатается сообщение “НВССП не корректно таймирована”.

Для выполнения второго шага используется алгоритм из статьи [5] (рис. 3 в приложении), проверяющий размеченность каждого минимального тупика. Просматриваются все места НВССП, при этом производится маскирование размеченных мест. После того как все места просмотрены, сеть модифицируется в сеть N_d способом, описанным в приложении. Далее строится матрица инцидентности для модифицированной сети и демаскируются все места сети, при этом восстанавливается исходная сеть. С полученной матрицей инцидентности решается система однородных уравнений. Каждому ненулевому решению такой системы соответствует неразмеченный минимальный тупик. Таким образом, если найдено хотя бы одно ненулевое решение, работа алгоритма прекращается и печатается "Не живая НВССП".

Для выполнения третьего шага используется алгоритм из статьи [5] (рис. 4 в приложении), проверяющий, является ли каждый минимальный тупик ловушкой. Просматриваются все места НВССП, имеющие более одной выходной дуги. Для каждого такого места выполняется следующая процедура. Последовательно просматриваются выходные переходы текущего места и производится маскирование выходных мест каждого такого перехода. После маскирования НВССП модифицируется в сеть N_d способом, описанным в приложении, и одновременно вычисляется критерий для оценки решения — это номер компоненты вектора. Если соответствующая компонента решения больше нуля, то решение удовлетворяет критерию. Строится матрица инцидентности для модифицированной сети, решается однородная система линейных уравнений и проверяется соответствие решения критерию. Если решение не удовлетворяет критерию, то демаскируются все места сети и рассматривается следующий выходной переход текущего места. В противном случае работа алгоритма заканчивается и печатается "Не живая НВССП". Если были перебраны все места сети и ни одно решение не удовлетворило критерию, то печатается "Живая НВССП".

Корректность **Алгоритма А1** следует из теорем 3.2, 3.3 и теорем А.2, А.3. Сложность шага 1 **Алгоритма А1** составляет $O(|F|)$, сложность шагов 2 и 3 — $O(|F|^8)$.

3.3. Анализ на основе S-компонент

Алгоритм А2.а производит углубленный анализ ограниченности структурированных НВССП — определяет максимальное количество фишек для каждого места.

ВХОД: Живая и ограниченная НВССП

1. Если $\neg$(НВССП корректно таймирована), то Печать "НВССП не корректно таймирована" и переход на ВЫХОД.
2. Для каждого места p, не принадлежащего ни одной S-компоненте,

Построить S-компоненту, содержащую минимальный тупик с местом p и входные переходы мест тупика.

3. Для каждой S-компоненты S

   Вычислить сумму $M(S)$ фишек в местах S-компоненты.

4. Для каждого места p

   Построить множество $S(p)$ S-компонент, которым принадлежит место p.

5. Для каждого места p

   Вычислить максимальное количество фишек в p по следующей формуле:

$$min\{M_0(P_1)|(P_1, T_1, F_1) - \text{ } S\text{-компонента в } N, \text{ содержащая } p\}.$$

ВЫХОД

Первый шаг аналогичен первому шагу **Алгоритма А1**.

На втором шаге осуществляется построение S-компонент, основанное на алгоритмах поиска минимальных тупиков из статьи [5] (см. рис. 6, 7 и 8 в приложении). Произвольно выбирается место, с которого начинается построение первого минимального тупика. Это место добавляется в очередь Q, содержащую места, с которых предполагается начать построение очередного маршрута H. Затем вызывается процедура поиска дуги, с которой начинается построение H (рис. 7 в приложении). Дуга выбирается следующим образом: из очереди Q извлекается место p и перебираются все его входные переходы. Дуга составляется из первого перехода t, не принадлежащего тупику и места p. Если все входные переходы места p принадлежат тупику, то место p удаляется из очереди и извлекается следующее место. Процесс продолжается до тех пор, пока в очереди Q есть места или пока не будет найдена дуга (t, p). После нахождения дуги запускается рекурсивная процедура поиска в глубину с переходом t в качестве начального узла (рис. 8 в приложении). Поиск в глубину начинается с добавления начального узла к маршруту H и в стек S. Затем проверяется, принадлежит ли какой-либо из входных узлов y начального узла x тупику. Если это так, то построение H закончено, нужно лишь добавить y в стек и вернуть значение “истина”, иначе перебираются все входные узлы y начального узла x и производится проверка, принадлежит ли y маршруту H. Если узел y не принадлежит H, то для него рекурсивно запускается поиск в глубину с y в качестве начального узла, иначе берется следующий входной узел

y. Если вызванная рекурсивно процедура поиска в глубину возвращает значение "истина", то H построен, и можно произвести возврат со значением "истина". В противном случае берется следующий входной узел начального узла x. Когда будут перебраны все входные узлы, следует забрать из стека начальный узел x и произвести возврат со значением "ложь". Заметим, что в силу свойств исходной НВССП маршрут H всегда может быть построен, т.е. на начальном уровне рекурсии не может произойти возврата со значением "ложь". После того как маршрут H будет построен, производится добавление мест из H в очередь Q, и маршрут H добавляется к тупику. После этого снова ищется начальная дуга, и процесс продолжается. Алгоритм завершает свою работу, если очередь Q пуста. Поскольку в процессе построения тупика мы добавляем к нему не только места из H, но и переходы, то по утверждению А.3 построенное множество будет являться S-компонентой. Процесс построения S-компонент продолжается до тех пор, пока в сети есть места, не принадлежащие ни одной S-компоненте.

На третьем шаге для каждой построенной S-компоненты S подсчитывается сумма $M(S)$ фишек в ее местах.

На четвертом шаге для каждого места p строится множество $S(p)$ S-компонент, которым принадлежит данное место.

На пятом шаге для каждого места p вычисляется максимальное количество фишек в p по следующей формуле: $min\{M_0(P_1)|(P_1, T_1, F_1)$ — S-компонента в N, содержащая $p\}$ (теорема А.6).

Корректность **Алгоритма А2.а** следует из теорем 3.2, 3.4, утверждений А.1–А.4 и теорем А.2, А.4–А.6.

Оценим временную сложность **Алгоритма А2.а**. Сложность шага 1 составляет $O(|F|)$. Сложность остальных шагов складывается из сложности алгоритма, строящего S-компоненты, и алгоритма, определяющего максимальное количество фишек в каждом месте сети. Очевидно, что построение минимального тупика состоит в худшем случае из $|T|$ вызовов процедуры поиска в глубину. Эффективность алгоритма поиска в глубину основана на том факте, что узлы не проверяются более одного раза. Очевидно, что сложность поиска в глубину зависит от количества рекурсивных вызовов. Для любой вершины сети поиск в глубину вызывается ровно один раз, и любая дуга, ведущая в данную вершину, проверяется не более двух раз. Поскольку сеть содержит $(|P| + |T|)$ узлов и $|F|$ дуг, то худшее время работы алгоритма будет $O(|P|+|T|+|F|)$. Таким образом, временная сложность построения минимального тупика составляет $O(|T| \cdot (|P| + |T| + |F|))$. Так как процедура построения минимального тупика вызывается в худшем случае $|P|$ раз, то сложность построения S-покрытия составляет $O(|P| \cdot |T| \cdot (|P| + |T| + |F|))$. Очевидно, что подсчет максимального количества фишек для каждого места имеет линейную сложность.

Алгоритм А2.б предназначен для построения начального состояния, при котором исходная структурированная НВССП будет живой и безопасной.

ВХОД: Структурированная НВССП

1. Если ¬(НВССП корректно таймирована), то Печать "НВССП не корректно таймирована" и переход на ВЫХОД.
2. Для каждого места p, не принадлежащего ни одной S-компоненте, Построить S-компоненту, содержащую минимальный тупик с местом p и входные переходы мест тупика.
3. Для каждой S-компоненты S
   (a) Вычислить сумму $M(S)$ фишек в местах S-компоненты.
   (b) Если $M(S) > 1$, то удалить лишние фишки из мест S-компоненты и установить $M_0^{new}(S) = 1$, иначе установить $M_0^{new}(S) = M(S)$.

ВЫХОД Новая начальная разметка $M_0^{new}(S)$.

Первый и второй шаги аналогичны соответствующим шагам **Алгоритма А2.а**.

На третьем шаге для каждой S-компоненты S вычисляется сумма $M(S)$ фишек в местах S-компоненты. Если $M(S)$=1, то разметка в местах текущей S-компоненты не меняется. В противном случае из мест S-компоненты, не принадлежащих другим S-компонентам, производится удаление фишек так, чтобы $M(S)$ стало равно 1. При этом свойство живости сети не нарушается, так как каждый ее минимальный тупик (места S-компоненты) остается размеченной ловушкой (теорема А.2).

Корректность **Алгоритма А2.б** следует из теорем 3.2, 3.4, утверждений А.1–А.4 и теорем А.2, А.4–А.6.

Оценим временную сложность **Алгоритма А2.б**. Очевидно, что построение живой и безопасной разметки при построенном S-покрытии имеет линейную временную сложность. Следовательно, временная сложность **Алгоритма А2.б** равна $O(|P| \cdot |T| \cdot (|P| + |T| + |F|))$.

3.4. Анализ на основе T-компонент

Алгоритм А3 предназначен для анализа свойства дивергентности при заданном множестве внутренних переходов.

ВХОД: Живая и ограниченная НВССП, T_{int} — множество внутренних переходов.

1. Если $\neg$(НВССП корректно таймирована), то Печать "НВССП не корректно таймирована" и переход на ВЫХОД.
2. Для каждого перехода t, не принадлежащего ни одной T-компоненте, построить T-компоненту.
3. $X := \emptyset$.
4. Для каждой T-компоненты T_k выполнить $X := X \cup \{k\}$, если все переходы T-компоненты T_k принадлежат множеству T_{int}.
5. Если $X \neq \emptyset$, то Печать "Дивергентная НВССП", иначе Печать "Недивергентная НВССП".

ВЫХОД

Первый шаг аналогичен первому шагу **Алгоритма А1**.

На втором шаге строятся T-компоненты по алгоритмам, являющимся модификацией алгоритмов поиска минимальных тупиков из статьи [5] (рис. 6, 7 и 8 в приложении). Произвольно выбирается переход, с которого начинается построение первой T-компоненты. Этот переход добавляется в очередь Q, содержащую переходы, с которых предполагается начать построение очередного маршрута H. Затем вызывается процедура поиска дуги, с которой начнется построение H. Дуга выбирается следующим образом: из очереди Q извлекается переход t и перебираются все его входные места, дуга составляется из первого места p, не принадлежащего T-компоненте, и перехода t. Если все входные места перехода t принадлежат T-компоненте, то переход t удаляется из очереди, и извлекается следующий переход. Процесс продолжается до тех пор, пока в очереди есть нерассмотренные переходы или пока не будет найдена дуга (p, t). После нахождения дуги запускается рекурсивная процедура поиска в глубину с местом p в качестве начального узла. Поиск в глубину начинается с добавления начальной вершины к маршруту H и в стек S. Затем проверяется, является ли начальный узел местом и принадлежит ли какой-либо из входных узлов y начального узла x T-компоненте. Если оба условия выполнены, то построение H закончено, нужно лишь добавить y в стек и вернуть значение "истина". Если нет, то перебираются все входные узлы y начального узла x и производится проверка, принадлежит ли y маршруту H. Если узел y не принадлежит H, то для него рекурсивно запускается поиск в глубину с y в качестве начального узла, иначе берется следующий входной узел y. Если вызванная рекурсивно процедура поиска в глубину возвращает значение "истина", то H построен, и можно произвести возврат со значением "истина". В противном случае берется следующий входной узел начального узла x. Когда будут перебраны все входные узлы, следует забрать из стека начальный узел x и произвести возврат со значением "ложь". Заметим, что в силу свойств исходной НВССП маршрут H всегда может быть построен, т.е. на начальном уровне рекурсии не может произойти возврата со значением "ложь". После того как маршрут H будет построен, производится добавление переходов H в очередь Q и маршрут H добавляется к T-компоненте. После этого снова ищется начальная дуга, и процесс продолжается. Алгоритм построения T-компонент завершает свою работу, если очередь Q пуста. После построения T-компоненты в сети ищется переход, не принадлежащий ни одной T-компоненте, и с него начинается построение очередной T-компоненты. Процесс продолжается до тех пор, пока в сети есть переходы, не принадлежащие ни одной T-компоненте.

На третьем шаге заводится пустое множество X.

На четвертом шаге проверяется, содержится ли во множестве внутренних переходов все множество переходов какой-либо T-компоненты T_k. Если это так, то печатается "Дивергентная НВССП", иначе печатается "Недивергентная НВССП".

Корректность **Алгоритма А3** следует из теорем 3.2–3.5, утверждения А.5 и теорем А.8, А.10.

Поскольку алгоритм поиска T-компонент является модификацией алгоритма поиска S-компонент, мы можем взять за основу оценку временной сложности, приведенную для **Алгоритма А2.а**. Отличие состоит в том, что построение T-компоненты состоит в худшем случае из $|P|$ вызовов процедуры поиска в глубину. Худшее время работы алгоритма поиска в глубину не изменится и будет составлять $O(|P| + |T| + |F|)$. Таким образом, временная сложность построения T-компоненты составляет $O(|P| \cdot (|P| + |T| + |F|))$. Так как процедура построения T-компоненты вызывается в худшем случае $|T|$ раз, то сложность построения T-покрытия составляет $O(|T| \cdot |P| \cdot (|P| + |T| + |F|))$ — так же, как и для S-покрытия. Очевидно, временная сложность проверки, содержится ли во множестве внутренних переходов все множество переходов какой-либо T-компоненты (таких T-компонент может быть несколько), равна $O(|T|)$. Общая временная сложность **Алгоритма А3** составляет $O(|T| \cdot |P| \cdot (|P| + |T| + |F|))$.

СПИСОК ЛИТЕРАТУРЫ

1. *Alur R., Dill D.* The theory of timed automata // Lect. Notes Comput. Sci. – 1991. – Vol. 600. – P. 45–73.
2. *Berthomieu B., Diaz M.* Modelling and verification of time dependent systems using time Petri nets // IEEE Transaction on Software Engineering. – 1991. – Vol. 17, N 3. – P. 259–273.
3. *Best E., Grahlmann B.* PEP — more than a Petri Net tool // Lect. Notes Comput. Sci. – 1996. – Vol. 1055. – P. 397–401.
4. *Commoner F., Holt A.W., Even S., Pnueli A.* Marked Directed Graphs // Journal of Computer and System Sciences. – 1990. – Vol. 5. – P. 511–527.
5. *Esparza J.,Silva M.* A polinomial time algorithm to decide liveness of bounded free choice nets // Theoretical Computer Science. – 1992. – Vol. 102(1).
6. *Desel J. Esparza J.* Free-choice Petri nets // Cambridge Tracts in Theoretical Computer Science. – 1995. – Vol. 40.
7. *C.Ghezzi, d. Mandrioli, S. Moraska, M. Pezze.* A general way to put time in Petri nets // Proc. 5th Internat. Workshop on Software Specification and Design, Pittsburg, Pennsylvania, May 1989. – P. 60–67.
8. *Henzinger T.A., Manna Z., Pnueli A.* Timed transition systems // Lect. Notes Comput. Sci. – 1991. – Vol. 600. – P. 226–251.
9. *Merlin P., Faber D. J.* Recoverability of communication protocols // IEEE Trans. of Communication. – 1976. – Vol. COM-24(9).
10. *Ramchandani C.* Analysis of asynchronous concurrent systems by timed Petri nets // PhD Thesis. MIT TR-120. Cambridge (Mass). – 1974
11. *Sifakis J.* Performance evaluation of systems using nets // Lect. Notes Comput. Sci. – 1981. – Vol. 84. – P. 307–320.
12. *Schneider S., Davies J., Jackson D.M., Reed G.M., Reed J.M., Roscoe A.W.* Timed CSP: theory and practice // Lect. Notes Comput. Sci. – 1991. – Vol. 600. – P. 640–675.
13. *Starke P.* Some properties of timed nets under the earliest firing time // Lect. Notes Comput. Sci. – 1990. – Vol. 424. – P. 418–432.

А. ПРИЛОЖЕНИЕ

В данном приложении приводятся определения и классические результаты по анализу поведенческих свойств СП со свободным выбором (ССП).

Определение ССП

Для эффективного анализа поведенческих свойств систем часто рассматриваются подклассы ординарных СП, в частности, ССП [6]. Достоинство этого класса сетей состоит в том, что, с одной стороны, ССП — это элегантный математический аппарат, для которого многие проблемы анализа являются разрешимыми, а с другой — ССП позволяют моделировать достаточно широкий спектр параллельных систем.

Начнем с определения вспомогательных понятий и обозначений.

Пусть $N = (P, T, F)$ — сеть и $P' \subseteq P$. *Тупиком* сети N называют множество мест $R = P'$, если $P' \neq \emptyset$ и ${}^\bullet R \subseteq R^\bullet$. *Ловушкой* сети N называют множество мест $Q = P'$, если $P' \neq \emptyset$ и $Q^\bullet \subseteq^\bullet Q$. Тупик (ловушка) называется *минимальным (минимальной)* тогда и только тогда, когда он(а) не содержит другого(ой) тупика (ловушки). Тупик R сильно связен тогда и только тогда, когда подсеть, образованная $R \cup {}^\bullet R$, сильно связна. Ловушка Q сильно связна тогда и только тогда, когда подсеть, образованная $Q \cup Q^\bullet$, сильно связна. Из определения тупика (ловушки) очевидно следует, что объединение тупиков (ловушек) также является тупиком (ловушкой).

Сеть N называется *S-сетью*, если каждый ее переход имеет ровно одну входную дугу и ровно одну выходную дугу.

Сеть N называется *T-сетью*, если каждое ее место имеет ровно одну входную дугу и ровно одну выходную дугу.

Пусть N' — подсеть сети N, образованная непустым множеством элементов X. Тогда N' — *S-компонента* N, если:

- ${}^\bullet p \cup p^\bullet \subseteq X$ для каждого места $p \in X$;
- N' — сильно связная S-сеть.

Пусть N' — подсеть сети N, образованная непустым множеством элементов X. Тогда N' — *T-компонента* N, если:

- ${}^\bullet t \cup t^\bullet \subseteq X$ для каждого перехода $t \in X$;
- N' — сильно связная T-сеть.

Пусть C — множество S-компонент сети. Тогда C называют *S-покрытием*, если любое место сети принадлежит какой-либо S-компоненте из C.

Пусть C — множество T-компонент сети. Тогда C называют *T-покрытием*, если любой переход сети принадлежит какой-либо T-компоненте из C.

Пусть $N = (P, T, F)$ — сеть с местами $P = \{p_1, \ldots, p_n\}$ и переходами $T = \{t_1, \ldots, t_m\}$. Матрица $C = \|c_{ij}\|(1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m)$, где $c_{ij} = F(t_j, p_i) - F(p_i, t_j)$, называется *матрицей инцидентности* сети N.

S-инвариантом сети N называется рациональное решение системы уравнений $X^T \cdot C = 0$. S-инвариант X сети N называется *неотрицательным*, если $X \geqslant 0$ и $X \neq 0$. *Поддержкой* неотрицательного S-инварианта X, обозначаемой как $\langle X \rangle$, называется множество мест p таких, что $X(p) > 0$.

T-инвариантом сети N называется рациональное решение системы уравнений $C \cdot X = 0$. Для T-инвариантов понятия неотрицательности и поддержки определяются так же, как и для S-инвариантов.

Сеть N называется

- *структурированной*, если существует разметка M_0, при которой (N, M_0) — живая и ограниченная сеть Петри;
- *сетью со свободным выбором* (*свободной сетью*) (СС), если $\forall p \in P,\ \forall t \in T \diamond (p, t) \in F \Rightarrow^\bullet t \times p^\bullet \subseteq F$;
- *сетью с асимметричным выбором* (*асимметричной сетью*) (АС), если $\forall p_1, p_2 \in P \diamond p_1^\bullet \ \cap\ p^\bullet \neq \emptyset \Rightarrow (p_1^\bullet \ \subseteq p_2^\bullet) \ \vee\ (p_2^\bullet \ \subseteq p_1^\bullet)$.

Ясно, что любая СС является АС, но не наоборот.

СП (N, M_0) называется

- *СП со свободным выбором* (*свободной СП*) (ССП), если N — СС;
- *СП с асимметричным выбором* (*асимметричной СП*) (АСП), если N — АСП;
- *k-ограниченной* (*ограниченной*), если существует натуральное число k такое, что для любой достижимой разметки и для любого места p верно $M(p) \leqslant k$. Если сеть Петри 1-ограничена, то ее называют *безопасной*;
- *живой*, если для любого перехода t и для любой достижимой разметки M существует достижимая из M разметка M', при которой переход t может сработать.

Классические результаты

Теорема А.1 (теорема Коммонера [4]). *ССП жива тогда и только тогда, когда каждый ее тупик содержит размеченную ловушку.*

Известно, что алгоритм, основанный на этой теореме, имеет экспоненциальную сложность, что затрудняет его практическое применение. Выходом из этой ситуации является разумное сужение класса анализируемых сетей. Испанскими учеными Эспарзой и Силвой [5] был получен следующий результат.

Теорема А.2. *Ограниченная ССП жива тогда и только тогда, когда каждый ее минимальный тупик является размеченной ловушкой.*

Для проверки, является ли каждый минимальный тупик сети размеченной ловушкой, в работе [5] был приведен алгоритм, имеющий полиномиальную сложность. Основой для алгоритма послужил следующий результат.

Теорема А.3 ([5]). *Пусть $N = (P, T, F)$ — сеть такая, что каждое место в ней имеет по крайней мере один входной переход, и $P' \subseteq P$. Тогда следующие утверждения эквивалентны:*

(а) P' — объединение множеств попарно различных сильно связных тупиков сети N;

(б) существует S-инвариант X сети N_d такой, что $\langle X \rangle = P'$.

Здесь сеть N_d строится по следующим правилам: места исходной сети N, имеющие не более одной входной и выходной дуги, остаются без изменений, а остальные места модифицируются способом, показанным на рис. 2.

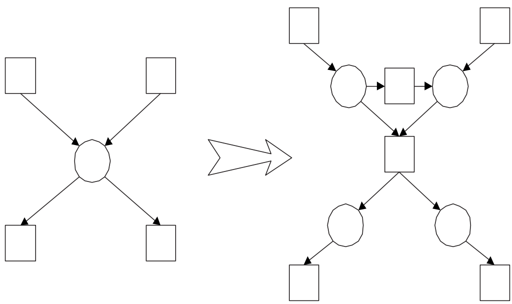

Рис. 2.

На рис. 3 показан алгоритм, проверяющий размеченность каждого минимального тупика, а на рис. 4 — алгоритм, проверяющий, является ли каждый минимальный тупик ловушкой.

Для структурированных сетей справедливы следующие факты.

Утверждение A.1 ([6]). *Структурированные сети сильно связны.*

Утверждение A.2 ([6]). *Каждое место структурированной СС содержится в минимальном тупике.*

Ряд интересных свойств СС связан с декомпозицией их на подсети специального вида, называемые S-компонентами.

Теорема A.4 ([6]). *Структурированные СС покрываются S-компонентами (рис. 5).*

Теорема A.5 ([6]). *Пусть N — структурированная СС. Тогда верно:*

(1) N содержит положительный S-инвариант,

(2) каждая ССП (N, M) ограничена,

(3) ССП (N, M) жива тогда и только тогда, когда каждая S-компонента в N размечена при M.

Строить S-компоненты можно с помощью следующего утверждения.

Утверждение A.3 ([6]). *Пусть R — минимальный тупик структурированной СС. Тогда*

(1) R — ловушка,

(2) подсеть, образованная $R \cup {}^{\bullet}R$, является S-компонентой.

Таким образом, для построения S-компонент достаточно найти все минимальные тупики. Алгоритмы построения минимального тупика, содержащего заданное место, приведены на рис. 6, 7 и 8.

Следующее утверждение иллюстрирует несколько простых свойств S-компонент.

Утверждение A.4 ([6]). *Пусть $N_1 = (P_1, T_1, F_1)$ — S-компонента СС N. Тогда верно:*

(1) $|{}^{\bullet}t \cap P_1| = 1 = |t^{\bullet} \cap P_1|$ для всех $t \in T_1$,

(2) если M и M' — разметки СС N такие, что M' достижима из M, то $M'(P_1) = M(P_1)$,

(3) P_1 является минимальным тупиком и минимальной ловушкой СС N.

На основе S-покрытия СС мы можем провести более детальный анализ свойства ограниченности. Следующая теорема позволяет разрабо-

Рис. 3. Алгоритм, проверяющий размеченность каждого минимального тупика

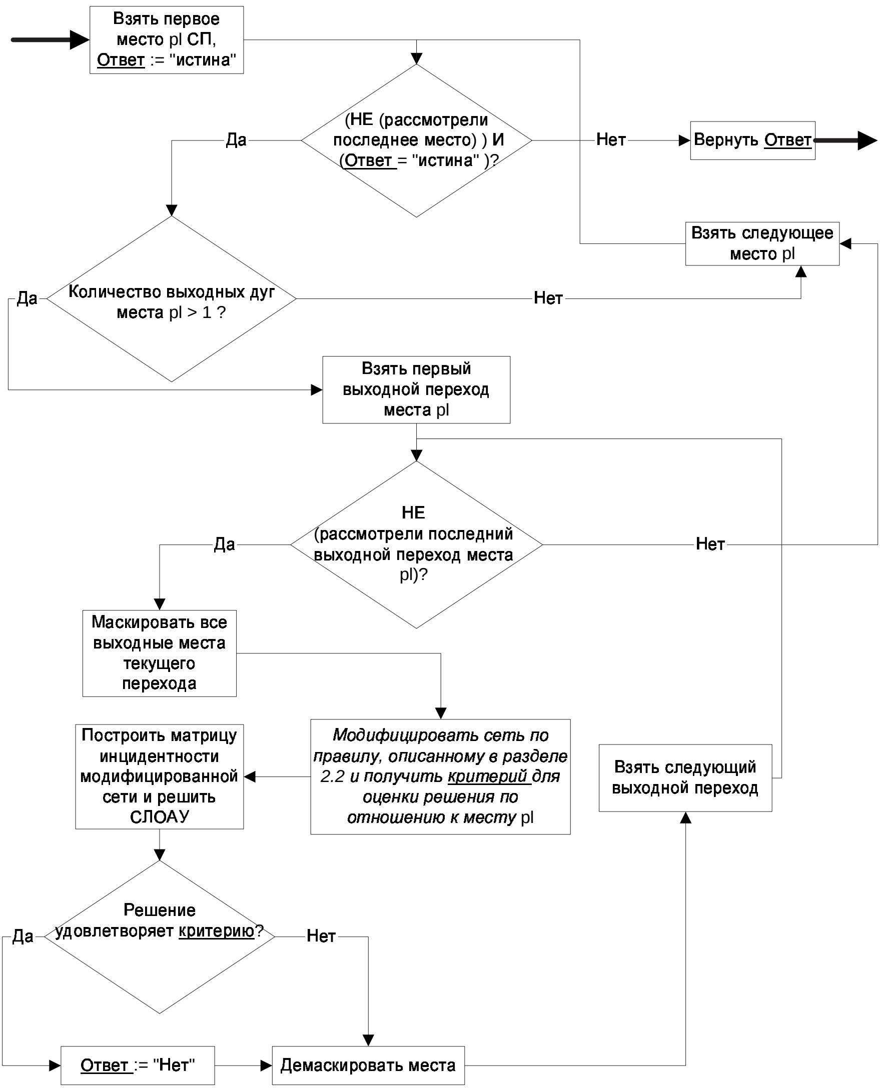

Рис. 4. Алгоритм, проверяющий, является ли каждый минимальный тупик ловушкой

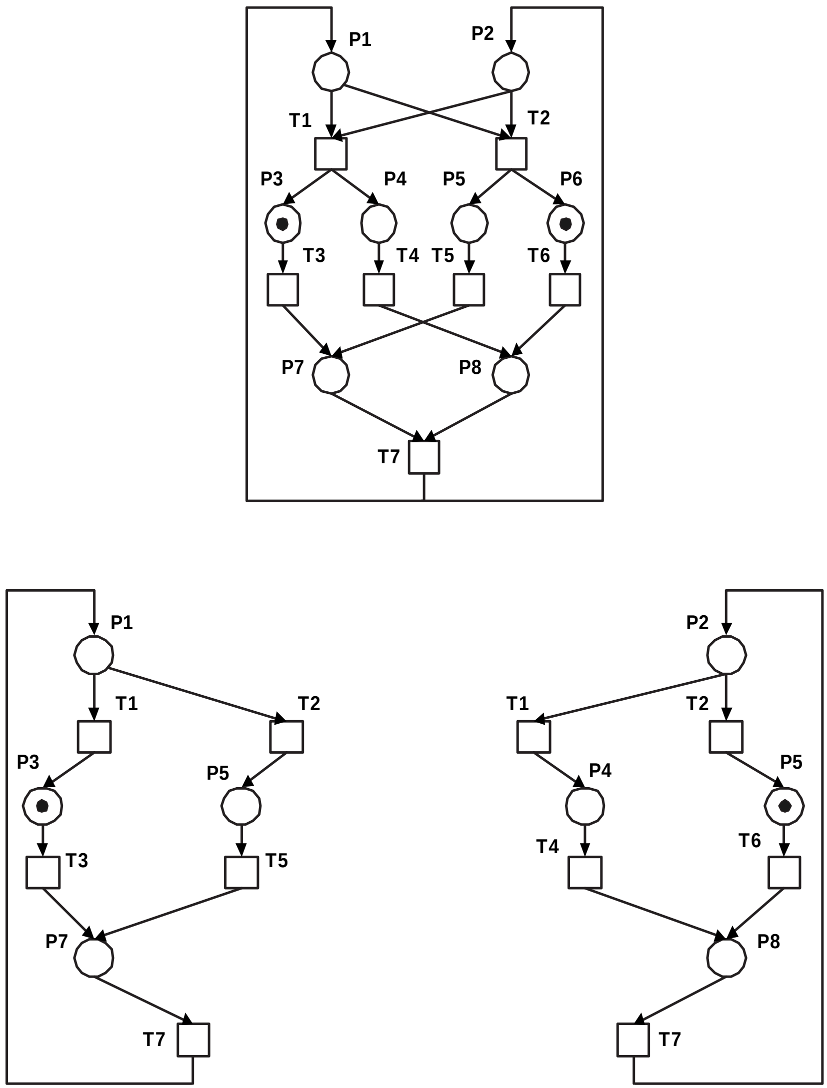

Рис. 5. Структурированная ССП и ее декомпозиция на S-компоненты

Рис. 6. Схема работы алгоритма, строящего минимальный тупик (Q — очередь мест (кандидатов для очередного маршрута H), S — стек для хранения H)

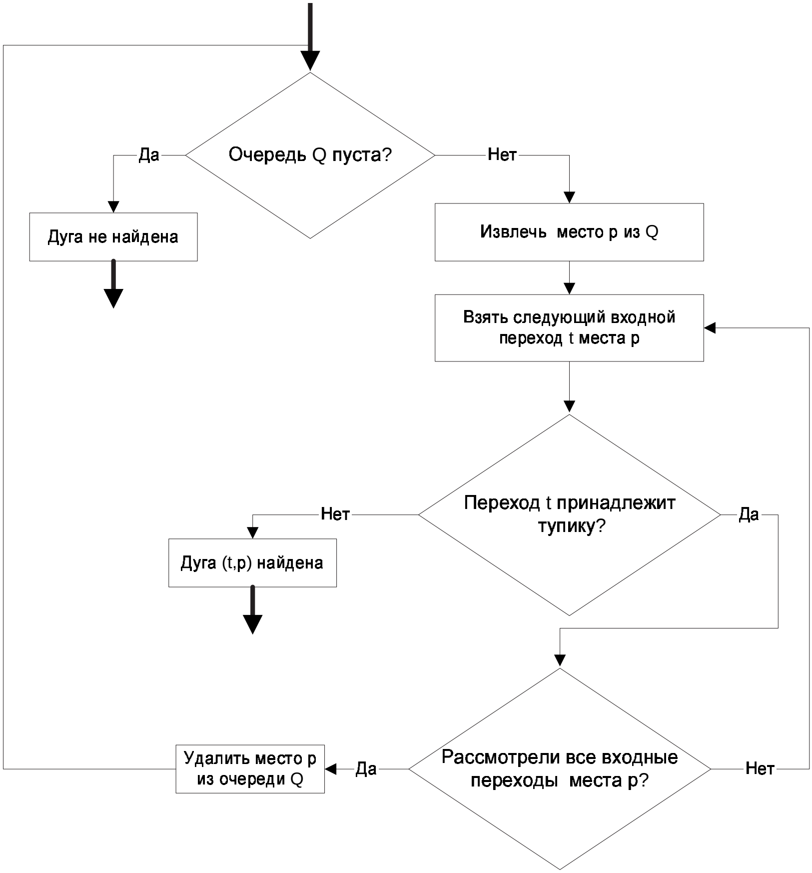

Рис. 7. Схема работы алгоритма, выбирающего дугу (t, p) для построения маршрута H

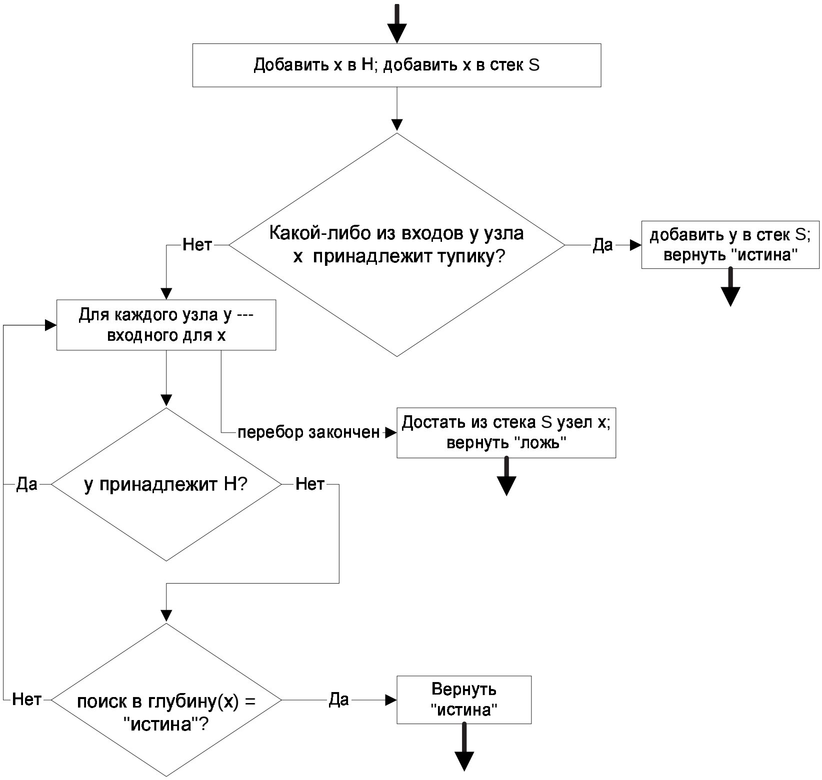

Рис. 8. Схема работы алгоритма поиска в глубину узла x для построения маршрута H

тать алгоритм, определяющий максимально возможное количество фишек для каждого места сети.

Теорема А.6 ([6]). *Пусть p — место живой и ограниченной ССП (N, M_0). Количество фишек в p ограничено числом*
$min\{M_0(P_1) | (P_1, T_1, F_1)$ *— S-компонента СС N, содержащая p*$\}$.

Существует большое количество алгоритмов анализа поведенческих свойств, которые основаны на свойстве безопасности.

Теорема А.7 ([6]). *Пусть N — структурированная СС. Тогда существует разметка M_0, при которой (N, M_0) является живой и безопасной ССП.*

Ряд интересных свойств СС связан с декомпозицией их на подсети специального вида, называемые T-компонентами.

Теорема А.8 ([6]). *Структурированные СС покрываются T-компонентами (рис. 9).*

T-компоненты обладают рядом свойств, которые могут быть применены непосредственно при анализе сетевого поведения.

Утверждение А.5 ([6]). *Пусть $N_1 = (P_1, T_1, F_1)$ — T-компонента СС N. Тогда верно:*

(1) $|\bullet p \cap T_1| = 1 = |\ p\bullet \cap T_1|$ *для каждого места* $p \in P_1$;

(2) пусть M_0 — разметка N и σ — последовательность срабатываний переходов из T_1. Тогда $M_0 \to M$ в сети N тогда и только тогда, когда $M_0|_{P_1} \to M|_{P_1}$ в СС N_1. Иначе говоря, поведение T-компоненты не зависит от поведения всей системы.

Для построения T-компонент можно легко адаптировать алгоритмы построения минимальных тупиков, рассмотренные выше.

Пусть $N_1 = (P_1, T_1, F_1)$ — T-компонента сети N. Говорят, что разметка M сети N *активирует* N_1, если сеть Петри $(N, M|_{S_1})$ жива. Следующая теорема позволяет более детально проанализировать свойство живости.

Теорема А.9 ([6]). *Пусть N_1 — T-компонента структурированной ССП (N, M_0). Тогда существует последовательность срабатываний τ такая, что M достижима из M_0 посредством τ, и M активирует N_1, причем τ не содержит переходов из N_1.*

При рассмотрении сетевой модели какой-либо реальной системы переходы сети можно разделить на *внутренние* и *внешние*. Внешние переходы моделируют действия, которые могут восприниматься наблюдателем извне системы, а внутренние переходы — действия, не имеющие заметного эффекта на поведение системы. Будем говорить, что

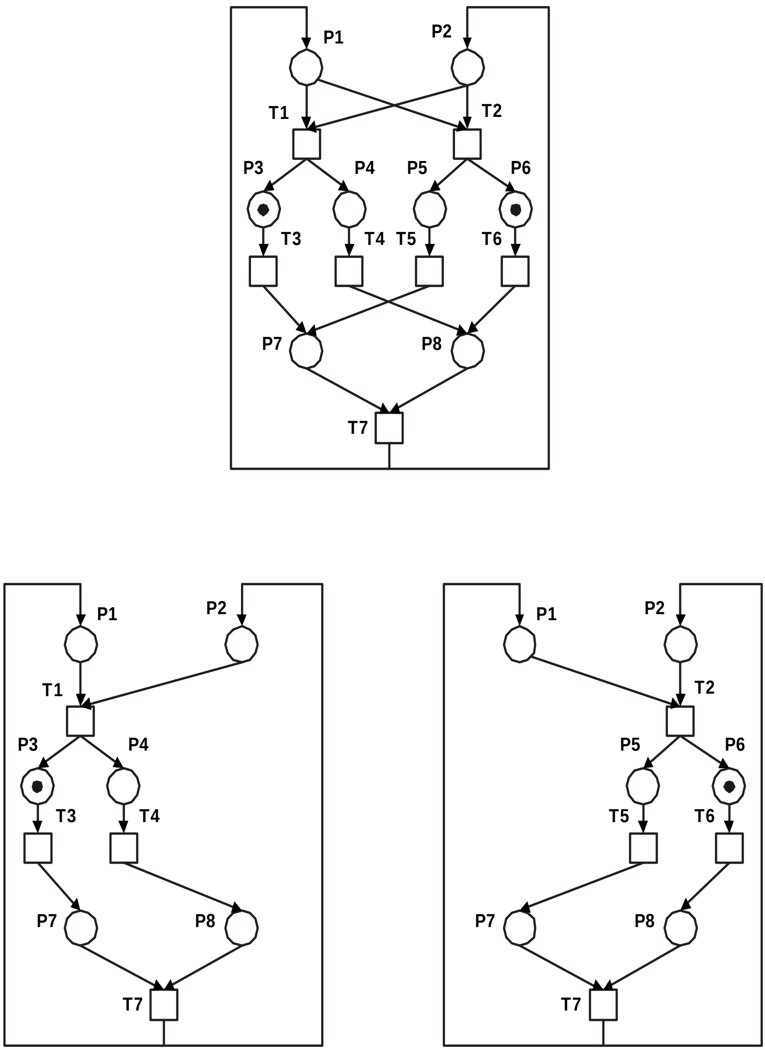

Рис. 9. Структурированная ССП и ее декомпозиция на T-компоненты

СП *дивергентна* для некоторого разбиения ее переходов на внутренние и внешние, если существует достижимая разметка, при которой возможна бесконечная последовательность срабатываний внутренних переходов. Дивергентная СП является плохо спроектированной, так как в ней возможна ситуация, при которой система будет занята бесконечной бесполезной работой.

Теорема A.10 ([6]). *Пусть* (N, M_0) *— живая и ограниченная ССП,* T_I *и* T_O *— разбиение переходов сети* N *на внутренние и внешние переходы соответственно. ССП* (N, M_0) *дивергентна по отношению к этому разбиению, если существует* T*-компонента* (P_1, T_1, F_1) *сети* N *такая, что* $T_1 \subseteq T_I$*.*

Александр В. Быстров

СТРУКТУРНЫЙ АНАЛИЗ ПОВЕДЕНИЯ НЕПРЕРЫВНО-ВРЕМЕННЫХ СЕТЕЙ ПЕТРИ

Препринт
137

Рукопись поступила в редакцию 11.12.2006
Рецензент Н.С. Грибовская
Редактор З. В. Скок

Подписано в печать 28.12.2006
Формат бумаги 60×84 1/16 Объем 1,7 уч.-изд.л., 2 п.л.
Тираж 60 экз.

Центр оперативной печати “Оригинал 2”
г.Бердск, ул. Островского, 55, оф. 02, тел. (383) 214 45 35