Фундаментальные и прикладные проблемы информатики и информационных технологий

УДК 51.681.3

Ю.В. Бойко, Н.Н. Глибовец, С.В. Ершов, С.Л. Крывый, С.Д. Погорелый, А.И. Ролик, С.Ф. Теленик, А.И. Куляс, Ю.В. Крак, М.В. Ясочка

Методы исследования свойств высокопроизводительных инфраструктур. Обзор

Представлен аналитический обзор современных методов верификации программного обеспечения параллельных и распределенных систем. Описаны методы верификации на основе исследования свойств конечных автоматов, сетей Петри и транзиционных систем.

An analytical survey of the modern verification methods of reactive and distributed systems is presented. The verification methods founded based on investigation properties of the finite state automata, Petri nets and the transition systems are described.

Представлено аналітичний огляд сучасних метод1в верифікації програмного забезпечення паралельних та розподілених систем. Описано методи верифікації на основі дослідження властивостей скінченних автоматів, мереж Петрі та транзиційних систем.

Введение. Представлен обзор результатов, полученных как авторами, так и другими исследователями в области создания надежного программного и технического обеспечения для высокопроизводительных инфраструктур. В частности описаны методы анализа семантических свойств моделей систем реактивного и распределенного типов. Выбор этих направлений объясняется тем, что в настоящее время в области проектирования, разработки и внедрения развиваются и активно применяются именно такого типа системы [1–8].

Анализ семантических свойств систем состоит в разработке анализаторов, которые, получая на входе в качестве входных данных информацию о системе, выдают на выходе ответы на вопросы о ее свойствах. Сложность решения такой проблемы связана как с алгоритмической неразрешимостью общей проблемы анализа, так и со сложностью самого процесса анализа. Поэтому такие ответы неизбежно будут частичными, но и они всегда очень существенны. Одним из подходов к частичному решению проблемы такого анализа есть исследование свойств математических моделей реальных систем формальными методами. Основными математическими моделями служат транзиционные системы, автоматные, сетевые и логико-алгебраические модели.

В связи с алгоритмической неразрешимостью общей проблемы анализа, он возможен на разного рода аппроксимациях структур, входящих в семантические спецификации исследуемых объектов. На практике эта методология может быть применена в процессе разработки иерархии семантик и иерархии абстрактных алгебр, а также логических языков на разных уровнях абстракции. Цель такой методологии – создание средств такой семантической системы анализаторов, которая (по возможности) автоматически выполняла бы анализ спецификаций.

Транзиционные системы и их произведения

Одной из наиболее общих математических моделей разного рода систем являются транзиционные системы. Посредством этой модели исследуются многие свойства реальных параллельных и распределенных систем.

Определение 1. *Транзиционной системой* (ТС) *называется пятерка* $A=(S,T,\alpha,\beta,s_0)$, *где* S *– множество состояний*, T *– множество переходов между состояниями*, $\alpha: T \to S$ *– функция начала перехода*, $\beta: T \to S$ *– функция конца перехода*, $s_0 \in S$ *– начальное состояние* ТС.

Ключевые слова: верификация, конечные автоматы, транзиционные системы, сети Петри, верификация на моделях.

Графически ТС изображается в виде орграфа, вершины которого соответствуют состояниям, а дуги – переходам ТС.

Пример 1. Рассмотрим две ТС.

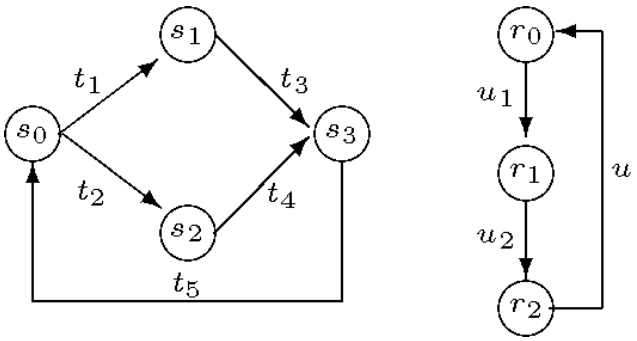

Рис. 1

В первой из приведенных ТС $A = (S = \{s_0, s_1, s_2, s_3\}, T = \{t_1, t_2, t_3, t_4\}, \alpha, \beta, s_0)$, где функции α и β задаются графом этой ТС. В частности, $\alpha(t_1) = s_0$, $\beta(t_1) = s_1$, $\alpha(t_5) = s_3$, $\beta(t_5) = s_0$.

Множество переходов T можно рассматривать как алфавит, а конечную или бесконечную последовательность переходов этого алфавита называть транзиционным словом или просто словом переходов. Множество всех слов в алфавите T будем обозначать $F(T)$.

Если $t \in T$, то тройка $(\alpha(t), t, \beta(t))$ называется *шагом вычислений* в ТС $A = (S, T, \alpha, \beta, s_0)$. Состояние $s \in S$ называется достижимым с помощью перехода $t \in T$, если существует $s' \in S$ такое, что (s, t, s') – шаг вычислений в ТС A. Слово $t_1 t_2 ... t_k \in F(T)$ называется *вычислением* в $A = (S, T, \alpha, \beta, s_0)$, если существует последовательность состояний $s_1, s_2, ..., s_k$ такая, что (s_{i-1}, t_i, s_i) есть шаг вычислений для каждого $i \in \{1, 2, \ldots, k\}$. Добавим к множеству переходов T пустой переход ε, который будет обозначать отсутствие перехода из состояния $s \in S$. Если $t_1, t_2, \ldots, t_k \in F(T)$ – вычисление в ТС, то говорят, что оно начинается в состоянии s_0 и ведет в состояние s_k. Вычисление называется *историей*, если оно стартует в начальном состоянии s_0. Слово $t_1 t_2 ... \in F(T)$ бесконечной длины называется *бесконечным вычислением* в ТС, если существует бесконечная последовательность состояний $s_{i_1} s_{i_2} ...$ такая, что $(s_{i_{j-1}}, t_j, s_{i_j})$ – шаг вычислений в ТС для каждого $i_j \geq 1$, и *бесконечной историей*, если $s_{i_1} = s_0$.

Если h – история, ведущая в состояние s, а c – вычисление, начинающееся в состоянии s, то конкатенация hc тоже есть история. В этом случае говорят, что hc – расширение истории h с помощью вычисления с.

Синхронное произведение ТС. Пусть $A_1, A_2, \ldots, A_n$ – ТС, где $A_i = (S_i, T_i, \alpha_i, \beta_i, s_0^i)$, $A_i \bigcap A_j = \varnothing$, если $i \neq j$, $i, j = 1, 2, ..., n$.

Определение 2. *Ограничением синхронизации называется подмножество* **T** *множества* $(T_1 \cup \{\varepsilon\}) \times (T_2 \cup \{\varepsilon\}) \times ... \times (T_n \cup \{\varepsilon\}) \setminus \{\varepsilon, \varepsilon, ..., \varepsilon\}$, *где* ε *– тождественный переход.*

Элементы множества **T** *называются глобальными переходами. Если* $\mathbf{t} = \{t_1, t_2, \ldots, t_n\} \in \mathbf{T}$ *и* $t_i \neq \varepsilon$, *то говорят, что* ТС A_i *принимает участие в переходе* **t**.

Кортеж $\mathbf{A} = (A_1, A_2, \ldots, A_n, \mathbf{T})$ *называется произведением* ТС $A_1, A_2, \ldots, A_n$ *над множеством* **T**, а ТС $A_1, A_2, \ldots, A_n$ *– компонентами* **A**.

На рис. 1 показано произведение TC_1 и TC_2, определяемое множеством глобальных переходов

$\mathbf{T} = \{(t_1, \varepsilon), (t_2, \varepsilon), (t_3, u_2), (t_4, u_2), (t_5, \varepsilon), (\varepsilon, u_1), (\varepsilon, u_3)\}$ (см. пример 1).

Глобальным состоянием $\mathbf{A} = (A_1, A_2, \ldots, A_n, \mathbf{T})$ называется n-ка $(s_1, s_2, ..., s_n)$, где $s_i \in S_i$, а состояние $(s_0^1, s_0^2, ..., s_0^n)$ – *начальным* состоянием **A**.

Шагом вычисления **A** называется тройка $(\mathbf{s}, \mathbf{t}, \mathbf{s}')$, где $\mathbf{s} = (s_1, s_2, ..., s_n)$ и $\mathbf{s}' = (s_1', s_2', ..., s_n')$ – глобальные состояния, а $\mathbf{t} = (t_1, t_2, ..., t_n)$ – гло-

бальный переход, удовлетворяющий следующим условиям: $\forall i \in \{1, 2, ..., n\}$

- если $t_i \neq \varepsilon$, то $s_i = \alpha(t_i)$ и $s_i' = \beta(t_i)$;
- если $t_i = \varepsilon$, то $s_i' = s_i$.

Глобальный переход **t** называется *допустимым* в глобальном состоянии **s**, если существует глобальное состояние **s′** такое, что (**s**, **t**, **s′**) – шаг вычисления.

Последовательность глобальных переходов $\mathbf{t_1}, \mathbf{t_2}, ..., \mathbf{t_k}, ...$ называется *глобальным вычислением,* если существует последовательность глобальных состояний $\mathbf{s_0}, \mathbf{s_1}, ..., \mathbf{s_k}$ такая, что $(\mathbf{s}_{i-1}, \mathbf{t}_i, \mathbf{s}_i)$ – шаг вычислений для каждого $i \in \{1,2,...,k\}$. В этом случае говорят, что глобальное вычисление начинается в глобальном состоянии $\mathbf{s_0}$ и ведет в глобальное состояние $\mathbf{s}_k$. Глобальное вычисление, которое начинается в состоянии $\mathbf{s}_0$, называется *глобальной историей вычисления.*

Пример 2. Рассмотрим произведение ТС на рис. 1. Начальное глобальное состояние – (s_0, r_0). Глобальным вычислением будет последовательность $(t_1, \varepsilon), (\varepsilon, u_1), (t_3, u_2)$, поскольку три шага вычислений $((s_0, r_0)(t_1, \varepsilon), (s_1, r_0))$, $((s_1, r_0)(\varepsilon, u_1), (s_1, r_1))$, $((s_1, r_1)(t_3, u_2), (s_3, r_2))$ составляет вычисление, ведущее из состояния (s_0, r_0) в состояние (s_3, r_2).

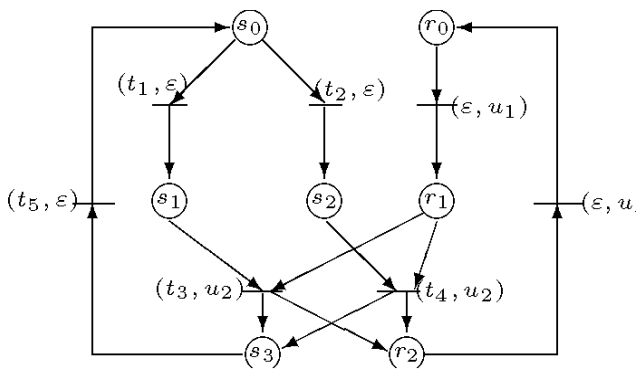

Рис. 2

Последовательность $(t_1, \varepsilon), (t_3, u_1)$ не будет глобальным вычислением, поскольку переход (t_3, u_1) не есть глобальным переходом из **T**.

Многие свойства произведения ТС можно исследовать путем его моделирования сетями Петри [8].

Сети Петри и представление ТС

Сети Петри (СП). *Сетью* называется тройка (P, T, F), где P и T непересекающиеся множества, элементы которых называются *местами* и *переходами* соответственно, а $F \subseteq (P \times T) \cup (T \times P)$ – отношение инцидентности. Элементы из F называются стрелками, а места – вершинами, подразумевая графическое представление сети. Если $(x, y) \in F$, то x называется *входной* вершиной y, а y – *выходной* вершиной x. Множества входных и выходных вершин для x обозначаются ${}^\bullet x$ и $x^\bullet$ соответственно.

Сеть (P, T, F) называется размеченной, если задана функция разметок $M : P \to N$, где N – множество натуральных чисел. Если $M(p) = m$, то это значит, что функция M ставит в вершину p ровно m фишек. Разметка сети в случае, когда $|P| = n$ и множество P упорядочено, представляется вектором $M = (m_1, m_2, ..., m_n)$, где $m_i = M(p_i)$, $p_i \in P$, $i = 1, 2, ..., n$.

Сетью Петри называется четверка (P, T, F, W, M_0), где (P, T, F) – сеть, M_0 – начальная разметка ее мест, а $W : F \to N \setminus \{0\}$ – функция кратности дуг СП.

Необходимость введения функции кратности дуг объясняется тем, что место и переход или переход и место могут быть связаны не одной, а несколькими дугами. Если в СП все дуги имеют кратность 1, то такая СП называется *ординарной*. Заметим, что имеется алгоритм, с помощью которого произвольная СП может быть преобразована в ординарную СП [9]. Такую СП будем обозначать четверкой (P, T, F, M_0) и далее, если не оговорено противное, под СП будем понимать ординарную сеть.

Переход $t \in T$ СП может сработать при разметке M, если она размечает каждое входное место этого перехода, т.е. ${}^\bullet t \subseteq M$. В этом

случае разметка M называется *допустимой* для перехода t.

Если M допустима для t, то этот переход может сработать и привести к новой разметке $M' = (M \setminus \{^\bullet t\}) \cup t^\bullet$. Разметка M' получается из разметки M путем удаления одной фишки из каждого входного места и добавления одной фишки в каждое выходное место перехода t. Переход от M к M' обозначается так: $M \xrightarrow{t} M'$. Говорят, что разметка M' достижима из разметки M, если существует последовательность переходов $t_1, t_2, \ldots, t_n$ из T таких, что $M \xrightarrow{t_1} M_1 \xrightarrow{t_2} M'_2 \xrightarrow{t_3} \ldots \xrightarrow{t_n} M_n$.

Разметка M' называется просто достижимой, если она достижима из начальной разметки M_0 СП.

СП как математическая модель вычислений хорошо изучены и поэтому полезно моделирование ТС с помощью СП.

Представление произведения ТС с помощью СП. СП (P, T, F, M_0) представляет произведение $\mathbf{A} = (\mathrm{A}_1, \mathrm{A}_2, \ldots, \mathrm{A}_n, \mathbf{T})$ ТС $\mathrm{A}_i = (S_i, T_i, \alpha_i, \beta_i, s_0^i)$, где $\mathrm{A}_i \bigcap \mathrm{A}_j = \varnothing$ при $i \neq j$, $i, j = 1, 2, \ldots, n$, если $P = S_1 \cup S_2 \cup \ldots \cup S_n$, $T = \mathbf{T}$, $F = \{(s, \mathbf{t}) \mid t_i \neq \varepsilon$ и $s = \alpha_i(t_i)\} \cup \{(\mathbf{t}, s) \mid t_i \neq \varepsilon$ и $s = \beta_i(t_i)\}$ для некоторого $i \in \{1, 2, \ldots, n\}$, где t_i обозначает i-ю компоненту $\mathbf{t} \in \mathbf{T}$, $M_0 = (s_0^1, s_0^2, \ldots, s_0^n)$.

Пример 3. СП, представляющая произведение ТС (см. рис. 1), имеет вид (рис. 2).

В этой СП имеем $^\bullet(t_2, \varepsilon) = \{s_1\}$, $(t_2, \varepsilon)^\bullet = \{s_2\}$, $^\bullet(t_4, u_2) = \{s_2, r_1\}$ и $(t_4, u_2)^\bullet = \{s_3, r_2\}$.

Далее будем использовать обозначения $^\bullet\mathbf{t} = \{\alpha_i(t_i) \mid t_i \neq \varepsilon\}$ и $\mathbf{t}^\bullet = \{\beta_i(t_i) \mid t_i \neq \varepsilon\}$.

Заметим, что семантика произведения ТС и семантика СП, представляющие его, согласуются в том смысле, что последовательность глобальных переходов $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_k$ будет глобальной историей произведения ТС $\mathbf{A}$ тогда и только тогда, когда она – допустимая последовательность срабатываний переходов в СП.

Представление произведения ТС в виде СП позволяет исследовать свойства этого произведения методами анализа хорошо развитых свойств СП.

Классы СП и их свойства. Рассмотрим классификацию СП и методы исследования их важнейших свойств. Для этого введем некоторые определения общего характера.

Определение 3. *Место p* СП *называется ограниченным, если существует такое число $k \in N$, что для произвольной достижимой разметки M в* СП *выполняется неравенство $M(p) \leq k$.* СП *называется ограниченной, если каждое ее место ограничено.*

Место p СП *называется безопасным, если для произвольной достижимой разметки M этой* СП $M(p) \leq 1$. СП *называется безопасной, если все ее места безопасны.*

Пара (p, t) $((t, p))$ – место p и переход t – в СП *называется местом–полуциклом* (*переходом–полуциклом*), *если место p* (*переход t*) *– как входное, так и выходное место* (*переход*) *для перехода t* (*места p*).

Пусть для некоторой СП (P, T, F, W, M_0) *имеется подмножество $Q \subseteq P$ и $Q \neq \varnothing$. Множество Q называют дедлоком* (*сифоном*) *тогда и только тогда, когда $^\bullet Q \subseteq Q^\bullet$, и ловушкой тогда и только тогда, когда $Q^\bullet \subseteq {}^\bullet Q$.*

Очевидно, что множество всех разметок в СП конечно тогда и только тогда, когда СП ограничена.

Матрица инцидентности и уравнение состояния. Для СП (S, W, M_0) с n переходами и m местами *матрицей инцидентности A* называется матрица с целочисленными коэффициентами размерности $n \times m$, элементы которой определяются уравнением

$$a_{ij} = I(t_j, p_i) - I(p_i, t_j),$$

где i – отношение инцидентности данной СП. Справедливость этого уравнения вытекает из правила сохранения фишек, поскольку коэффициент a_{ij} матрицы инцидентности представляет число фишек, которые перемещают-

ся, изменяются и добавляются в место p_i при срабатывании перехода t_j в СП.

Поскольку j-я строка в матрице инцидентности A означает смену разметки в результате срабатывания перехода t_j можно записать уравнение, называемое *уравнением состояния* СП:

$$A \cdot x = M_k - M_0 = d, \qquad (1)$$

где $k = 1, 2, ...$, а A – матрица инцидентности СП, x – вектор срабатывания переходов, M_k – финальная разметка, а M_0 – начальная разметка СП.

Система уравнений (1) называется также *необходимым условием достижимости* разметки M_k из начальной разметки M_0. Известно, что когда разметка M_k достижима из начальной разметки M_0, то система (1) всегда имеет решение. Обратное утверждение, к сожалению, не всегда корректно. Но из необходимого условия достижимости разметки вытекает достаточное условие недостижимости заданной разметки в СП. Это следует из правила контрапозиции. Заметим, что решения системы (1) ищут в множестве натуральных чисел N. Для решения такого типа систем разработано несколько алгоритмов [3, 6, 7, 16]. Все эти алгоритмы принадлежат классу временной сложности NP [11, 14] и проблема построения более эффективных алгоритмов для таких систем остается актуальной.

Анализ структурных свойств СП. Рассмотрим классы СП, в частности, так называемые чистые СП и методы анализа их структурных свойств. Последние – это свойства, не зависящие от начальной разметки СП в том смысле, что они выполняются для ее произвольной начальной разметки.

Определение 4. СП (S, W, M_0) *называется чистой, если из* $(p,t) \in F$ *вытекает* $(t,p) \notin F$ *или* ${}^\bullet t \cap t^\bullet = \varnothing$ (*т.е.* СП *не содержит мест–полуциклов*).

К структурным свойствам чистых СП относятся следующие свойства: *структурная живучесть*, *управляемость*, *структурная ограниченность*, *консервативность и частичная консервативность*, *повторяемость и частичная повторяемость*, *непротиворечивость и частичная непротиворечивость*.

Методы анализа этих свойств зависят от класса СП, для которых они исследуются. Классификация СП выполняется путем введения определенных ограничений на места и переходы СП.

Определение 5. *Пусть* (P, T, F, W, M_0) – СП, *удовлетворяющая условиям*, $\forall t \in T, p \in P$ $(I(p,t), I(t,p) \in \{0,1\})$. СП (P, T, F, W, M_0) *называется*:

машиной состояний (МС) $\Leftrightarrow \forall t \in T \left| t^\bullet \right| = \left| {}^\bullet t \right| \le 1$;

синхрографом (СГ) $\Leftrightarrow \forall p \in P \left| p^\bullet \right| = \left| {}^\bullet p \right| \le 1$;

СП *свободного выбора* (СВ) $\Leftrightarrow \forall p, p' \in$

$$\in P\,(p \ne p' \to (p^\bullet \cap p'^\bullet = \varnothing \vee \left| p^\bullet \right| = \left| p'^\bullet \right| \le 1));$$

расширенной СП *свободного выбора* (РСВ)

$$\Leftrightarrow \forall p, p' \in P\,(p \ne p' \to (p^\bullet \cap p'^\bullet = \varnothing \vee p^\bullet = p'^\bullet));$$

простой СП (ПСП) $\Leftrightarrow \forall p, p' \in P\,(p \ne p' \to$

$$\to (p^\bullet \cap p'^\bullet = \varnothing \vee \left| p^\bullet \right| \le 1 \vee \left| p'^\bullet \right| \le 1));$$

асимметричной СП (АСП) $\Leftrightarrow \forall p, p' \in$

$$\in P(p \ne p' \to (p^\bullet \cap p'^\bullet = \varnothing \vee p^\bullet \subseteq p'^\bullet \vee p'^\bullet \subseteq p^\bullet)).$$

Для этих классов СП справедливы следующие факты.

Теорема 1. 1) СП *свободного выбора структурно жива тогда и только тогда, когда каждый ее дедлок имеет ловушку.*

2) Если СП (S, W, M_0) с m местами полностью управляема, то $rank(A^T) = m$, где A – матрица инцидентности СП.

3) СП (S, W, M_0) структурно ограничена тогда и только тогда, когда, система $A \cdot x \ge 0$, $x > 0$ несовместна над N или, что то же самое, система $A^T \cdot y \le 0$, $y > 0$, имеет хотя бы одно решение в N.

4) СП (S, W, M_0) (частично) консервативна тогда и только тогда, когда система $A \cdot x \ge 0$, $x > 0$ $(x \ge 0)$, несовместна над N или, что то же

самое, система $A^T \cdot y \le 0$ имеет хотя бы одно решение $y > 0$ ($y \ge 0$) над N.

5) СП (S, W, M_0) (частично) повторяема тогда и только тогда, когда система $A \cdot x \ge 0$ имеет хотя бы одно решение $x > 0$ ($x \ge 0$) над N.

6) СП (S, W, M_0) (частично) непротиворечива тогда и только тогда, когда система $A \cdot x = 0$ имеет хотя бы одно решение $x > 0$ ($x \ge 0$) над N.

***S*- и *T*-инварианты и ограниченность.** Введенное уравнение состояния СП, которое имело вид системы линейных диофантовых уравнений над множеством натуральных чисел N:

$$M_d = M_0 + A \cdot x, \qquad (2)$$

где M_0, M_d – начальная и конечная разметки СП соответственно, а A – матрица инцидентности этой СП, позволяет ввести понятия инвариантов СП.

Определение 6. *Решение x системы $A \cdot x = 0$ (т.е. когда в выражении (2) $M_0 = M_d$) называется Т-инвариантом* СП, *а решение системы $A^T \cdot y = 0$ называется S-инвариантом* СП (*или P-инвариантом*).

Инварианты СП – полезное средство при исследовании ее структурных свойств. Это вытекает из таких утверждений. Пусть дана некоторая СП с m местами и n переходами.

Теорема 2. 1) Вектор y размерности m есть S-инвариант СП тогда и только тогда, когда $M^T y = M_0^T y$ для произвольной фиксированной начальной разметки M_0 и произвольной достижимой разметки M.

2) Вектор x размерности n есть T-инвариант СП тогда и только тогда, когда существует разметка M_0 и последовательность срабатывания переходов σ, ведущая от M_0 к M_0, такие, что $\sigma = x$.

Из теоремы 1 п. 3 о структурной ограниченности чистых СП вытекает, что существует вектор y такой, что $A \cdot y < 0$. Но тогда существует $x \ge 0$ такой, что $M = M_0 + A \cdot x$ и $M^T y = M_0^T y + x^T A^T y$. Поскольку $A^T y \le 0$, то $M^T y \le M_0^T y$ и, следовательно,

$$M(p) \le (M_0^T y) / y(p), \qquad (3)$$

где $y(p)$ означает p-ю компоненту вектора y.

Полученное неравенство дает верхнюю оценку для числа фишек, которые укладываются в место p. Эта граница может быть улучшена, если в неравенстве (3) используются все S-инварианты из минимального порождающего множества S-инвариантов. Поскольку инварианты СП служат векторами, то минимальность векторов рассматривается над множеством натуральных чисел N, т.е. вектор y называется *минимальным*, если не существует другого вектора y_1 такого, что $y_1(p) \le y(p)$ для всех мест p.

Определение 7. *Порождающее множество* S-инвариантов (T-инвариантов) *называется минимальным порождающим множеством S-инвариантов* (*T-инвариантов*), *если не существует ни одного непустого его подмножества*, *которое тоже – порождающее*.

Из теории систем линейных уравнений известно, что *TSS*-алгоритм [12] находит минимальное порождающее множество решений такой системы и для ее нахождения можно применить этот алгоритм.

Таким образом, неравенство (3) редуцируется к виду

$$M(p) \le \min\left[M_0^T y_i / y_i(p) \right], \qquad (4)$$

где минимум ищем на всем множестве S-инвариантов из минимального порождающего множества. В работе [13] сказано, що данную оценку улучшить невозможно ни для каких других инвариантов.

Инвариантам СП отводится существенная роль в исследовании свойства ограниченности.

Определение 8. *Говорят*, *что* СП *покрывается позитивными S-(T)-инвариантами тогда и только тогда*, *когда для произвольного места p_i (перехода t_i) существует S-(T)-инвариант y такой, что $y_i > 0$ для всех $i = 1, 2, ..., |P|$.*

Связь между ограниченностью и инвариантами СП выражается такими утверждениями [2, 13].

Теорема 3. 1) Если все места СП покрываются позитивными *S*-инвариантами, то она ограничена.

2) Если существует *S*-инвариант $y \geq 0$, у которого компонента $y_i > 0$, то место p_i ограничено, т.е. существует такое натуральное число *k*, что для произвольной разметки *M*, достижимой из начальной разметки, справедливо неравенство $M(p_i) \leq k$.

3) Если СП ограничена и живая, то она покрывается позитивными *T*-инвариантами.

4) Если СП ограничена и переход t_i в СП живой, то существует *T*-инвариант $x \geq 0$, у которого компонента $x_i > 0$.

Поиск дедлоков и ловушек в СП. Рассмотрим метод анализа живучести СП с помощью дедлоков и ловушек, который базируется на использовании диофантовых ограничений.

Напомним, что когда произвольный переход имеет выходное место, принадлежащее дедлоку *Q*, то в него следует включать и его выходное место. А для ловушки наоборот. Из этого определения вытекает, что когда дедлок *Q* есть пустое множество, то он всегда остается таковым в процессе функционирования СП. Для ловушки ситуация противоположная: если хотя бы одно ее место получило фишку, то ловушка постоянно остается непустым множеством в процессе функционирования СП.

Дедлоки и ловушки можно найти в СП путем решения системы логичных уравнений, описывающих эти свойства, или эквивалентной ей системы линейных уравнений над множеством {0,1} [13].

Ловушка называется *помеченной* начальной разметкой, если хотя бы одно место этой ловушки получает по крайней мере одну фишку.

Необходимость построения множеств дедлоков и ловушек для данной СП состоит в том, что посредством анализа полученных множеств можно исследовать свойство живучести СП. Это вытекает из такого утверждения.

Теорема 4. СП *свободного выбора как и расширенная* СП *свободного выбора живая тогда и только тогда*, *когда каждый дедлок этой* СП *содержит ловушку*, *помеченную начальной разметкой* M_0.

При использовании СП для верификации свойств систем существенным будет определение пути, ведущего к ошибке или к подозрительному месту, или переходу в СП. Этот путь скрыт в *T*-инвариантах, поскольку они содержат информацию только о том, какие переходы сработали, но не несут никакой информации о последовательности их срабатывания. Если установлено, что СП ограничена, то граф достижимых разметок такой СП есть конечным автоматом. Применяя к этому автомату алгоритм анализа, построим регулярное выражение, а по нему определим кратчайшие пути, ведущие в подозрительные места СП.

Первая группа свойств СП проверяется средствами линейной алгебры, в частности, средствами решения систем линейных диофантовых уравнений и неравенств в области натуральных чисел.

Эта группа средств базируется на оригинальном методе построения минимального порождающего множества решений систем линейных уравнений в области натуральных чисел. Этот метод решения назван *TSS*-методом [3]. С помощью этого алгоритма генерируются инварианты СП без какой-либо дополнительной обработки. Использование этого алгоритма позволяет решить проблему достижимости в подклассе СП свободного выбора; недостижимости разметки в общем случае; ограниченности СП для единственного ее места; вычисления ловушек и дедлоков СП; структурных свойств для данной СП.

Вторая группа средств базируется на хорошо разработанных алгоритмах построения, представления и обхода графов [10].

Третья группа тоже имеет хорошую алгоритмическую поддержку в виде алгоритмов

анализа и преобразования регулярных языков и регулярных выражений [5].

Перейдем к рассмотрению методов верификации свойств систем логико-алгебраическими средствами и средствами конечных автоматов.

Анализ свойств реактивных систем

Реактивной называется система, которая должна потенциально работать бесконечно. Методы верификации таких систем основаны на проверке на модели и некоторых ее разновидностях. Неформально суть этого метода состоит в следующем. Ожидаемые свойства реальной системы описываются в виде формул некоторого формального логического языка, а реальная система моделируется соответствующей ТС или произведением ТС. Верификация состоит в проверке выполнимости заданных формул на модели [1, 8]. Одним из популярных логических языков есть язык линейной темпоральной логики (*LTL – linear temporal logic*).

Язык линейной темпоральной логики. Множество *LTL*-формул над заданным множеством атомарных формул *AP* определяется индуктивно:

– каждая атомарная формула есть *LTL*-формулой,

– если φ *LTL*-формула, то $\neg\varphi$ и $\mathbf{X}\varphi$ *LTL*-формулы,

– если φ, ψ *LTL*-формулы, то $\varphi\vee\psi$ и $\varphi\mathbf{U}\psi$ есть *LTL*-формулы.

LTL-формулы интерпретируются над бесконечными словами, символами которых являются множества атомарных формул, т.е. бесконечными словами в алфавите $B(AP)$, где $B(AP)$ – булеан множества атомарных формул *AP*. Интуитивно это означает, что некоторое множество атомарных формул, соответствующих исследуемому базисному утверждению, выполняется в *i*-й момент времени.

Пусть φ – *LTL*-формула и бесконечное слово $\pi = x_0x_1x_2\ldots$, где $x_i \notin B(AP)$ для каждого $i \geq 0$. Обозначение $\pi \models \varphi$ означает, что слово π выполняет φ или удовлетворяет φ. Отношение выполнимости $\models$ определяется индуктивно: пусть $p \in AP$ и $\pi^i = x_ix_{i+1}\ldots$ – суффикс слова π, тогда

$\pi \models p$, если $p \in x_0$,

$\pi \models \neg\varphi$, если π не выполняет φ,

$\pi \models \neg\varphi\vee\psi$, если $\pi \models \varphi$ или $\pi \models \psi$,

$\pi \models \mathbf{X}\varphi$, если $\pi^1 \models \varphi$,

$\pi \models \varphi\mathbf{U}\psi$, если $\exists n \geq 0\ \pi^n \models \psi$ и $\forall i(0 \leq i < n)\ \pi^i \models \varphi$.

Формула $\mathbf{X}\varphi$ читается как φ *в следующий момент*, а $\varphi\mathbf{U}\psi$ – как φ *пока не* ψ. Другими словами, $\mathbf{X}\varphi$ выполняется в данный момент времени, если в следующий момент будет выполняться φ, а $\varphi\mathbf{U}\psi$ выполняется в данный момент времени, если формула ψ выполняется в данный момент или она будет выполнена в будущий момент времени, а в каждый момент времени до этого момента, выполняется формула φ.

Остальные логические связки вводятся обычным путем: $true = p\vee\neg p$ для любого $p\in AP$, $false = \neg true$, $\varphi\wedge\psi = \neg\varphi\vee\psi$, $\varphi\mathbf{R}\psi = \neg(\neg\varphi\mathbf{U}\neg\psi)$, $\mathbf{F}\varphi = true\,\mathbf{U}\varphi$ и $\mathbf{G}\varphi = false\,\mathbf{R}\varphi$.

Пример 4. Пусть $AP = \{p\}$ и даны *LTL*-формулы $\mathbf{G}(p \to \mathbf{X}\neg p)$ и $\mathbf{F}(p\wedge\mathbf{X}p)$ над алфавитом $AP = \{p\}$ атомарных формул. Первая формула читается как *всегда, если p выполняется в данный момент времени*, то в следующий момент *она не выполняется*, а вторая формула – *существует два последовательных момента времени, когда формула p выполняется*.

Пусть $(p0)^\omega$ и $00pp(0)^\omega$ обозначают два бесконечных слова вида:
$p0p0p0\ldots$ и $00pp000\ldots$ соответственно. На этих словах получаем:

$(p0)^\omega \models \mathbf{G}(p \to \mathbf{X}\neg p)$, $00pp(0)^\omega$ не выполняет $\mathbf{G}(p \to \mathbf{X}\neg p)$,

$(p0)^\omega$ не выполняет $\mathbf{F}(p\wedge\mathbf{X}p)$, $00pp(0)^\omega \models \mathbf{F}(p\wedge\mathbf{X}p)$.

Интерпретация *LTL*-формул на произведении ТС. Пусть имеется произведение ТС

$\mathbf{A} = (A_1, A_2, \ldots, A_n, \mathbf{T})$, где $A_i = (S_i, T_i, \alpha_i, \beta_i, a_0^i)$, $i = 0,1,...,n$. Базисными утверждениями будут такие *текущим локальным состоянием i-й компоненты является* a_i. Следовательно, множеством атомарных формул выбирается множество $AP = \bigcup_{i=1}^{n} S_i$.

Пусть задана бесконечная глобальная история $\mathbf{h} = \mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3, \ldots$ произведения **A**, тогда существует единственная последовательность глобальных состояний $\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2\ldots$ такая, что $\mathbf{a}_0$ – начальное состояние, а $(\mathbf{a}_i, \mathbf{t}_{i+1}, \mathbf{a}_{i+1})$ – шаг вычислений в **A** для каждого $i \geq 0$. Иными словами, $\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2\ldots$ – последовательность состояний, которые проходят во время выполнения **h**. Бесконечная последовательность $\pi(\mathbf{h})$ множеств атомарных формул определяется так: для каждого $i \geq 0$ i-й элемент $\pi(\mathbf{h})$ представляет собой множество локальных состояний глобального состояния $\mathbf{a}_i$ (т.е. множество локальных состояний компонент произведения ТС в i-й момент времени). Из определения шага вычисления следует, что если S_i и S_{i+1} соответственно i-й и $(i + 1)$-й элемент $\pi(\mathbf{h})$, то $S_{i+1} = (S_i \setminus^{\bullet} t_i) \cup t_i^{\bullet}$.

Пример 5. Рассмотрим произведение ТС, показанных на рис. 3 [8]. *LTL*-формулы строятся над алфавитом $AP = \{t_0, t_1, u_0, u_1\}$ и пусть $\mathbf{T} = \{\mathbf{a} = (\varepsilon, a), \mathbf{b} = (\varepsilon, b), \mathbf{c} = (c, \varepsilon)\}$.

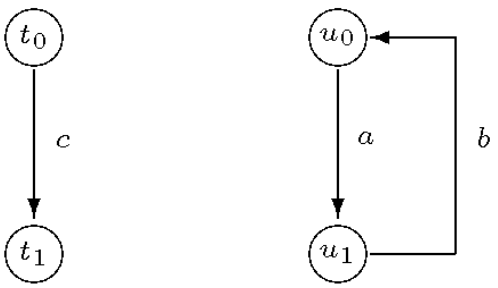

Рис. 3

Последовательность $\mathbf{h} = \mathbf{abc}(\mathbf{ab})^{\omega}$ – бесконечная история. Последовательность глобальных состояний, появляющихся во время выполнения, будет такой:

$$(t_0,u_0)(t_0,u_1)(t_0,u_0)((t_1,u_0)(t_1,u_1))^{\omega},$$

т.е. $\pi(\mathbf{h}) = (t_0,u_0)(t_0,u_1)(t_0,u_0)((t_1,u_0)(t_1,u_1))^{\omega}$.

Теперь можно определить интерпретацию *LTL*-формулы φ на $\pi(\mathbf{h})$. Будем говорить, что произведение **A** выполняет формулу φ (обозначение $\mathbf{A} \models \varphi$), если каждая бесконечная история произведения **A** выполняет φ. Другими словами, произведение **A** выполняет формулу φ, если все ее бесконечные истории выполняют формулу φ.

Проблема проверки выполнимости на модели состоит в определении для заданных произведения **A** и *LTL*-формулы φ, выполнима или нет формула φ на **A**.

Как и в случае историй, видим, что для данной бесконечной ψ-истории $\sigma = \boldsymbol{t}_1, \boldsymbol{t}_2, \ldots$ существует единственная последовательность $\boldsymbol{r}_0, \boldsymbol{r}_1, \ldots$ ψ-состояний такая, что $(\mathbf{r}_{i-1}, \mathbf{t}_i, \mathbf{r}_i)$ есть ψ-шагом произведения **A** для каждого $i \geq 1$.

Обозначим эту последовательность $\pi_{\psi}(\sigma)$ и назовем ее ψ-последовательностью σ. В этом случае говорят, что σ выполняет ψ и обозначают это $\sigma \models \psi$, если $\pi_{\psi}(\sigma) \models \psi$.

Проверка *LTL*-свойств. Эта проблема имеет несколько эквивалентных формулировок. Проверка того, что все бесконечные истории произведения **A** выполняют *LTL*-формулу ψ эквивалентна тому, что существует некоторая бесконечная история, которая не выполняет ψ, а это, в свою очередь, эквивалентно тому, что существует некоторая история, которая выполняет $\neg\psi$. Таким образом, при рассмотрении бесконечных историй, выполняющих некоторое свойство, как язык слов бесконечной длины в алфавите AP_{ψ}, проблема выполнимости сводится к проблеме проверки пустоты некоторых такого типа языков.

В данном случае тестер свойств произведения **A** рассматривает **A** как механизм, распознающий язык L слов бесконечной длины соответствующих ψ-историям, и бесконечным историям произведения **A**. Следовательно, по-

лучаем такую процедуру проверки выполнимости формулы ψ:

а) построить тестер, распознающий язык L_1 всех ψ-историй, выполняющих $\neg\psi$;

б) используя этот тестер и его произведение с тестером языка L всех ψ-историй, построить тестер, допускающий язык $L \cap L_1$;

в) проверить равенство $L \cap L_1 = \varnothing$; если оно выполняется, то формула ψ выполняется на всех бесконечных ψ-историях, иначе на некоторой выполняется $\neg\psi$ (и эта история дает контрпример).

Такого типа тестеры известны – это автоматы Бюхи и обобщенные автоматы Бюхи, называемые также автоматами Мюллера [5, 16]. Тестер для *LTL*-формул называют *Бюхи-тестером* или просто *тестером* для произведения *A*. Тестер – это тройка вида BT = (B, *F*, λ), где $\mathrm{B} = (S, T, \alpha, \beta, a_0)$ – транзитивная система, $F \subseteq S$ – множество заключительных состояний и $\lambda : T \to \mathbf{T}$ – функция отметок, сопоставляющая каждому переходу B некоторый глобальный переход из **T** произведения **A**. Тестер BT воспринимает бесконечные последовательности $\mathbf{t}_1\mathbf{t}_2\mathbf{t}_3... \in \mathbf{T}^\omega$, если существует бесконечная история $h = u_1u_2u_3...$ в B и заключительное состояние $a \in F$ такое, что $\mathbf{t}_i = \lambda(u_i)$ для каждого $i \geq 1$ и на истории *h* состояние *a* появляется бесконечно часто. Это значит, что последовательность $\pi(h)$ содержит бесконечно много вхождений состояния *a.* Язык, который воспринимается BT, состоит из слов $\mathbf{T}^\omega$.

Пусть ψ – *LTL*-формула. Тестер BT проверяет выполнимость формулы ψ или служит тестером свойства ψ, если он воспринимает все бесконечные ψ-истории **A**, которые выполняют ψ.

Существует огромная литература по проблеме проверки на модели и большое число различных конструкций для такого типа проверки. Здесь будет рассмотрена одна из самых простых конструкций.

Построение Бюхи-тестера часто выполняется в два этапа: сначала строится обобщенный Бюхи-тестер для заданной *LTL*-формулы ψ, а затем этот тестер преобразуется в Бюхи-тестер для этой формулы.

Обобщенным Бюхи-тестером для **A** называется кортеж BT = (B, $\{F_0, F_1, ..., F_{k-1}\}$, λ), где B, λ такие же, как и в определении Бюхи-тестера, а F_0, $F_1, ...$, F_{k-1} – множество подмножеств заключительных состояний. B воспринимает бесконечную последовательность $\mathbf{t}_1\mathbf{t}_2\mathbf{t}_3... \in \mathbf{T}^\omega$, если существует бесконечная история $\mathbf{h} = a_1, a_2, a_3, \ldots$ и заключительные состояния $a_1 \in F_0, a_2 \in F_1, \ldots \ldots, a_k \in F_{k-1}$ такие, что $\mathbf{t}_i = \lambda(u_i)$ для каждого $i \geq 1$ и каждое состояние $a_1, \ldots, a_k$ входит в **h** бесконечное число раз. Язык называется распознаваемым или допускаемым тестером BT, если каждое слово этого языка допускается этим тестером.

Сравнение с логическим подходом к анализу свойств систем показывает, что:

– некоторые свойства могут быть выражены ТС и СП более естественно и проще, чем в логических формализмах;

– некоторые свойства (ограниченность, возможность возврата к начальному состоянию, определение дедлоков и ловушек) вовсе не могут быть выражены или очень трудно выражаются средствами логических формализмов;

– проверка соответствующих свойств средствами ТС и СП более проста.

Заключение. Подводя итоги сказанному, отметим две основные проблемы, с которыми сталкиваются разработчики систем анализа программ.

Основная проблема, препятствующая широкому внедрению описанных методов, – *проблема комбинаторного взрыва*, которая состоит в том, что при моделировании реальной системы относительно небольших размеров, ее математическая модель может иметь

астрономическое число состояний, и такой объект (СП или ТС) не может поместиться в память компьютера, что приводит к невозможности его дальнейшего анализа и обработки. На поиск решения этой проблемы направлены главные усилия специалистов в области разработки формальных методов анализа программного обеспечения.

Вторая проблема состоит в том, что имеющиеся алгоритмы анализа свойств обладают *высокой временной и емкостной сложностью,* что затрудняет их широкое использование в коммерческих системах анализа и верификации свойств формальных моделей реальных систем.

В общем случае ситуация выглядит так, что с каждым годом сложность программного обеспечения возрастает и необходимы автоматизированные средства их анализа. С другой стороны, имеющиеся средства анализа свойств таких программных систем не могут обеспечить надежную и качественную их верификацию. В этом и состоит, по мнению авторов, главное противоречие и трудности, связанные с разработкой надежного и высококачественного программного обеспечения.

1. *Карпов Ю.Г.* MODEL CHECKING: Верификация параллельных и распределенных программных систем. – С-Пб.: БХВ, 2010. – 551 с.
2. *Котов В.Е.* Сети Петри. – М.: Наука, 1984. – 157 с.
3. *Крывый С.Л.* О некоторых методах решения и критериях совместности систем линейных диофантовых уравнений в области натуральных чисел // Кибернетика и системный анализ. – 1999. – № 4. – С. 12–36.
4. *Схрейвер А.* Теория линейного и целочисленного программирования, В 2-х т. – М.: Мир, 1991. – 742 с.
5. *Трахтенброт Б.А.*, *Барздинь Я.М.* Конечные автоматы (Поведение и синтез). – М.: Наука, 1970. – 400 с.
6. *Contejan E.*, *Ajili F.* Avoiding slack variables in the solving of linear diophantine equations and inequations // Theoretical Comp. Science. – 1997. – **173**. – P. 183–208.
7. *Clausen M.*, *Fortenbacher A.* Efficient solution of linear diophantine equations // Symbolic Computation. – 1989. – **8**, N 1, 2. – P. 201–216.
8. *Esparza J.*, *Heljanko K.* Unfoldings. A Partial-Order Approach to Model Checking. EATCS Mongraphs in Theoretical Computer Science. – Springer-Verlag, 2008. – 172 p.
9. *Hack M.H.T.* Decidability Questions for Petri Nets // Ph. D. Thesis, M.I.T. – 1976. – 32 p.
10. *Jonathan L. Gross*, *Jay Yellen* Handbook of Graph Theory. – CRC Press, 2004. – P. 57– 68.
11. *Johnson D.S.* A Catalogue of Complexity Classes // J. van Leeuwen / Ed. «Handbook of Theoretical Computer Science». – V.A., 1990: Elsevier. – P. 67–161.
12. *Kryvyi S.* A Criteria of compatibility systems of linear diophantine constraints. – Lect. Notes in Comp. Sci. – Springer-Verlag, 2002. – **2328**. – P. 264–271.
13. *Murata T.* Petri Nets: Properties, Analysis and Applications // Proc. of the IEEE. – 1989. – **77, N** 4. – P. 541–580.
14. *Papadimitriou C.H.* Computational complexity. – Addison-Wesley. – 1994. – 532 p.
15. *Pettier L.* Minimal solution of linear diophantine systems: bounds and algorithms // Proc. of the Fourth Int. Conf. on Rewriting Techniq. and Appl., Como, Italy, 1991. – P. 162–173.
16. *Thomas W.* Automata on infinite objects. – Handbook on theoretical computer science. – 1990. – P. 135–191.

Поступила 03.12.2014

Тел. для справок: +38 044 259-0511 (Киев)
E-mail: *sl.krivoi@gmail.com*; *sdp@univ.net.ua*