РАСПРЕДЕЛЕННЫЕ ИНФОРМ.-ВЫЧИСЛ. РЕСУРСЫ И МАТ. МОДЕЛИРОВАНИЕ
МКВМ-2004 РАБОЧИЕ СОВЕЩАНИЯ С. 73–85

Проверка модели для вполне структурированных систем переходов автоматного типа*

Е.В. Кузьмин, В.А. Соколов†

1. Введение

Одной из важных задач при проектировании программного обеспечения является проверка корректности параллельных систем. Под корректностью понимается полное соответствие системы задачам, для которых она создается. Корректность определяется в соответствии с формальной спецификацией, описывающей желаемое поведение системы. Процесс проверки соответствия системы требованиям, заданным в спецификации, называется верификацией.

Проверка модели (model checking) – один из подходов к решению проблемы верификации. В качестве языков спецификации для выражения свойств систем при этом подходе используются темпоральные логики. Задача проверки модели состоит в определении выполнимости для системы, заданной формальным образом, свойства, записанного в виде формулы темпоральной логики.

Проверка модели широко используется при верификации систем с конечным числом состояний, но эта автоматическая техника может быть применена для некоторых классов систем с бесконечным числом состояний и подмножеств темпоральных логик.

Существует большое количество формальных моделей, которые используются для описания параллельных систем. Многие из этих моделей могут быть рассмотрены как вполне структурированные системы переходов [3].

Вполне структурированные системы переходов – это весьма широкий класс систем переходов с бесконечным числом состояний, для которых разрешимость многих свойств следует из существования совместимого с отношением переходов вполне упорядочиваемого квазипорядка на множестве состояний.

В данной работе рассматривается класс вполне структурированных систем переходов автоматного типа [10], примером которых может служить формализм взаимодействующих раскрашивающих автоматов [4]. Исследуется разрешимость задачи проверки модели для данного класса систем переходов и различных подмножеств автоматной логики, в рамках которой элементарные высказывания интерпретируются верхними и нижними конусами.

Автоматная логика AL является одной из самых выразительных программных логик. Например, логика AL более мощная по выразительной силе, чем такие широко используемые логики как CTL и LTL, поскольку формулы этих логики могут быть адекватно интерпретированы в терминах автоматов.

Логика AL определяется в духе расширенной темпоральной логики ETL [8, 9]. Ранее автоматная логика была представлена в [1], где исследовался во-

*Работа поддержана грантом РФФИ 03-01-00804.
†Ярославский государственный университет. E-mail: sokolov@uniyar.ac.ru

прос разрешимости свойств, заданных автоматной логикой, для сетей Петри с потерями.

В рамках AL возмжно построение формул, выражающих темпоральные свойства систем с учетом действий при переходе из одного состояния в другое и элементарных высказываний над множеством состояний системы. Логика позволяет работать с конечными и бесконечными исполнениями системы переходов, а также учитывать временное ветвление, т. е. позволяет строить формулы снабженные кванторами всеобщности и существования.

Автоматная логика позволяет рассматривать разрешимость некоторых не охваченных ранее классов темпоральных свойств [10]. Более того, поскольку вполне структурированная система переходов автоматного типа по своей управляющей структуре является конечным автоматом, применение автоматной логики для задания свойств, а также исследование вопросов разрешимости этих свойств представляется весьма интересным и логичным.

В работе доказываются утверждения, следствием которых является, например, разрешимость некоторых подмножеств темпоральных логик CTL и LTL. В частности, имеем разрешимость задачи проверки модели для вполне структурированных систем переходов автоматного типа и формул логики CTL, построенных над множеством элементарных высказываний, которые интерпретируются *нижними конусами*, с помощью операторов конъюнкции, дизъюнкции и оператора U (“until”, что позволяет выражать свойства системы переходов, связанные только с конечной частью исполнения системы). Также разрешима данная задача и для формул логик CTL и LTL, которые строятся над множеством элементарных высказываний, интерпретирующихся *верхними конусами*, с помощью операторов конъюнкции, дизъюнкции и оператора $\tilde{U}$ (“releases”, что уже позволяет выражать свойства, связанные с *бесконечными* исполнениями системы). И, наконец, разрешимы свойства, выражаемые формулами логики $\neg LTL$, построенные с помощью конъюнкций, дизъюнкций и оператора U при интерпретации элементарных высказываний как *верхних конусов*.

Кроме того, с использованием теории счетчиковых машин показана и общая неразрешимость задачи проверки модели для ряда подмножеств автоматной логики и вполне структурированных систем переходов автоматного типа.

Работа организована следующим образом. В п. 2 даются основные понятия и определения. Затем вводится автоматная логика, а в пп. 4 и 5 приводятся основные результаты.

2. Основные понятия и определения

2.1. Системы переходов. Бинарное отношение $\leq$ называется отношением *частичного порядка*, если оно рефлексивно, транзитивно и антисимметрично. Только рефлексивное и транзитивное отношение называется отношением *квазипорядка*. Квазипорядок $\leq$ на множестве X называется *вполне упорядочиваемым* (well-quasi-ordering), если для любой бесконечной последовательности $x_0, x_1, x_2, \ldots$ элементов из X существуют индексы $i < j$ такие, что $x_i \leq x_j$.

Пусть $\leq$ – вполне упорядочиваемый квазипорядок на множестве X. *Идеалом или верхним конусом* (соответственно, *нижним конусом*) в X называется подмножество $I \subseteq X$, такое что для $x \in I$, $y \in X$ и $x \leq y$ (соответ-

ственно, $x \geq y$), следует $y \in I$. Идеал может быть получен замыканием кверху некоторого множества. Каждый элемент $x \in X$ порождает верхний конус $\uparrow x \stackrel{\text{def}}{=} \{y |\ y \geq x\}$. *Базисом* верхнего конуса I называется множество $\min(I)$ такое, что $I = \cup_{x \in \min(I)} \uparrow x$.

Утверждение 1. *Если $\leq$ – вполне упорядочиваемый квазипорядок на множестве X, то всякий верхний конус I имеет конечный базис.*

Утверждение 2. *Если $\leq$ – вполне упорядочиваемый квазипорядок на множестве X, то всякая бесконечно возрастающая (по отношению вложения множеств) последовательность $I_0 \subseteq I_1 \subseteq I_2 \subseteq \ldots$ верхних конусов стабилизируется, т. е. найдется такое $k \in \mathbb{N}$, что $I_k = I_{k+1} = I_{k+2} = \ldots$.*

Доказательства этих утверждений см., например, в [3].

Система помеченных переходов есть четверка $LTS = (S, T, \rightarrow, s_0)$, где S – множество состояний с элементами $s_0, s_1, s_2, \ldots$; T – некоторый алфавит пометок (множество имен действий); $\rightarrow \subseteq (S \times T \times S)$ – отношение переходов между состояниями; $s_0 \in S$ – начальное состояние системы.

Переход (s, t, s') обычно обозначается как $s \stackrel{t}{\rightarrow} s'$, что означает, что действие с именем t переводит состояние s в состояние s'. Через $\mathrm{Succ}(s)$ обозначается множество последующих состояний для s, через $\mathrm{Pred}(s)$ – множество его предыдущих состояний. Система LTS будет конечно ветвящейся, если для любого s множество $\mathrm{Succ}(s)$ конечно. В данной работе рассматриваются системы с конечным ветвлением и бесконечным числом состояний.

Последовательное исполнение для LTS есть конечная или бесконечная цепочка переходов $s_0 \stackrel{t_1}{\rightarrow} s_1 \stackrel{t_2}{\rightarrow} s_2 \rightarrow \ldots$, где s_0 – начальное состояние системы.

Система помеченных переходов – одна из наиболее распространенных моделей для описания поведения систем. Для решения задач анализа семантических свойств систем переходов полезной оказывается теория вполне структурированных систем переходов.

Определение 1. Вполне структурированной системой переходов с совместимостью по возрастанию (соответственно, по убыванию) называется система переходов $LTS = (S, T, \rightarrow, s_0)$, дополненная отношением квазипорядка $\leq \subseteq S \times S$, удовлетворяющая следующим условиям:

1) отношение $\leq$ является вполне упорядочиваемым квазипорядком;
2) квазипорядок $\leq$ совместим по возрастанию с отношением переходов $\rightarrow$, а именно, для любых состояний $s_1 \leq s_2$ и перехода $s_1 \stackrel{t}{\rightarrow} s_1'$ существует переход $s_2 \stackrel{t}{\rightarrow} s_2'$, такой что $s_1' \leq s_2'$ (соответственно, квазипорядок $\leq$ совместим по убыванию с отношением переходов $\rightarrow$, а именно, для любых состояний $s_1 \leq s_2$ и перехода $s_2 \stackrel{t}{\rightarrow} s_2'$ существует переход $s_1 \stackrel{t}{\rightarrow} s_1'$, такой что $s_1' \leq s_2'$);
3) для каждого состояния $s \in S$ и действия $t \in T$ можно вычислить предбазис, т. е. вычислимо множество $\min(\mathrm{Pred}_t(\uparrow s))$.

Определение 2. Вполне структурированная система переходов автоматного типа WSA – вполне структурированная система переходов $LTS{=}(S, T, \rightarrow, s_0)$, дополненная отношением вполне упорядочиваемого квазипорядка $\leq \subseteq S \times S$, где $S = Q \times D$, Q – конечное множество управляющих состояний, D – конечное или бесконечное множество значений некоторых характеристик, $s_0 \in S$ – начальное состояние, $\rightarrow \subseteq S \times T \times S$ такое, что $\exists \rightarrow_\bullet \subseteq Q \times T \times Q$: для любых состояний (q, d), (q', d') и некоторой метки перехода t переход $(q, d) \xrightarrow{t} (q', d')$ существует тогда и только тогда, когда существует переход $q \xrightarrow{t}_\bullet q'$; отношение $\leq$ определяется так, что $(q, d) \leq (q', d') \iff q{=}q'$ и $d \leq d'$.

Из определения следует, что WSA обладает свойствами нижней и верхней совместимости отношения $\leq$ с отношением переходов $\rightarrow$.

Система переходов является эффективной по пересечению, если для любых состояний s, s' вычислимо множество $\min(\uparrow s \cap \uparrow s')$.

Для анализа некоторых свойств вполне структурированных систем переходов используется конструкция *покрывающего дерева* [2].

Определение 3. *Покрывающим деревом* вполне структурированной системы переходов $(LTS, \leq)$ для состояния $s_0 \in S$ называется конечный ориентированный граф (дерево), такой что

1) вершины дерева помечены состояниями системы LTS;
2) каждая вершина объявлена либо *живой*, либо *мертвой*;
3) корню дерева приписана пометка s_0, и эта вершина живая;
4) мертвые вершины не имеют потомков;
5) живая вершина с пометкой s имеет по одному потомку, помеченному s', для каждого состояния $s' \in \mathrm{Succ}(s)$;
6) если на пути от корня до некоторой вершины s' встречается вершина с пометкой s такой, что $s \leq s'$, то говорят, что s' *покрывает* s, и вершина s' объявляется мертвой; в противном случае s' – живая вершина.

Листья в покрывающем дереве помечены финальными или покрывающими состояниями. Для вполне структурированных систем переходов частичный порядок $\leq$ является вполне упорядочиваемым. Благодаря этому все пути в покрывающем дереве конечны, так как всякий бесконечный путь должен был бы содержать покрывающую вершину. В силу леммы Кенига о конечных деревьях, если покрывающее дерево не имеет бесконечных ветвлений, то оно конечно.

Более того, очевидно, имеет место следующее

Утверждение 3. *Если отношение порядка $\leq$ разрешимо и отображение* Succ *вычислимо, то покрывающее дерево для вполне структурированной системы переходов может быть эффективно построено.*

Для построения покрывающего дерева не требуется совместимость между отношениями упорядоченности $\leq$ и перехода $\rightarrow$. Однако, именно при выполнении условия совместимости покрывающее дерево содержит полезную информацию о свойствах поведения системы.

2.2. Счетчиковые машины. *Машина Минского* или *счетчиковая машина* M – это набор $(\{q_0, \ldots, q_n\}, \{x_1, \ldots, x_m\}, \{\delta_0, \ldots, \delta_{n-1}\})$, где x_i – счетчик, q_i – состояние, q_0 – начальное состояние, q_n – финальное (заключительное) состояние, δ_i – правило переходов для q_i $(0 \leq i \leq n-1)$.

Состояния q_i, $0 \leq i \leq n-1$, подразделяются на два типа. Состояния первого типа имеют правила переходов вида $(1 \leq j \leq m,\ 0 \leq k \leq n)$:

$$\delta_i : x_j := x_j + 1;\ \texttt{goto}\ q_k.$$

Для состояний второго типа имеем $(0 \leq k' \leq n)$:

$$\delta_i :\ \texttt{if}\ x_j > 0\ \texttt{then}\ (x_j := x_j - 1;\ \texttt{goto}\ q_k)\ \texttt{else goto}\ q_{k'}.$$

Конфигурация машины Минского – это набор $(q_i, c_1, \ldots, c_m)$, где q_i – состояние машины, $c_1, \ldots, c_m$ – целые неотрицательные числа, являющиеся значениями соответствующих счетчиков.

Исполнение машины – это последовательность конфигураций $s_0 \to s_1 \to s_2 \to \ldots$, начинающаяся в начальной конфигурации s_0 и индуктивно определяемая в соответствии с правилами переходов. Необходимо отметить, что исполнение детерминировано, так как каждое состояние имеет не более одного правила переходов. Машина Минского останавливается, если исполнение содержит конфигурацию с состоянием q_n, т. е. достигает финального состояния. Так как машина Минского уже всего с двумя счетчиками может моделировать машину Тьюринга, проблема (останова) достижения финального состояния из некоторой начальной конфигурации (q_0, c_1, c_2) для двухсчетчиковой машины является неразрешимой [11].

3. Автоматная логика

Автоматная логика AL определяется над системами с тотальным отношением переходов, т. е. над системами, где из каждого состояния имеется хотя бы один переход. Рассмотрим произвольную вполне структурированную систему переходов автоматного типа $\mathcal{S}$. Выделим управляющие состояния системы $\mathcal{S}$, из которых не существует переходов. В отношение переходов $\to_\bullet$ добавим для каждого такого локального состояния q пустой переход τ, т.е. переход, переводящий некоторое локальное состояние q само в себя $q \xrightarrow{\tau}_\bullet q$. Также добавим в отношение $\to$ соответствующий τ переход для каждого состояния вида $s = (q, d)$, т. е. $s = (q, d) \xrightarrow{\tau} s = (q, d)$. Таким образом мы получили вполне структурированную систему переходов автоматного типа $\mathcal{S}'$, у которой любое исполнение может продолжаться бесконечно долго. Далее будет предполагаться, что все рассматриваемые системы имеют тотальное отношение переходов.

Определение 4. Конечный автомат над алфавитом Σ, принимающий конечные последовательности (исполнения), – набор $\mathcal{A}_f = (\Sigma, Q, q_0, \delta, F)$, где Q – это конечное множество состояний, q_0 – начальное состояние, $\delta \subseteq Q \times \Sigma \times Q$ – функция переходов, $F \subseteq Q$ – множество финальных (принимающих) состояний. Конечная последовательность $a_0 a_1 \ldots a_n \in \Sigma^*$ принимается конечным автоматом $\mathcal{A}_f$, если существует последовательность переходов $q_0 a_0 q_1 a_1 \ldots a_n q_n$ автомата

$\mathcal{A}_f$ из состояния q_0 такая, что $\forall\, i$ $(0 \le i \le n)$: $q_i \in Q$, $a_i \in \Sigma$, и $q_n \in F$. Обозначим $L(\mathcal{A}_f)$ (язык) множество последовательностей, принимаемых конечным автоматом $\mathcal{A}_f$.

Определение 5. Автомат Бюхи над алфавитом Σ, принимающий бесконечные последовательности, – это набор $\mathcal{A}_\omega = (\Sigma, Q, q_0, \delta, F)$. Бесконечная последовательность $a_0 a_1 \cdots \in \Sigma^\omega$ принимается автоматом Бюхи $\mathcal{A}_\omega$, если существует бесконечная последовательность переходов $q_0 a_0 q_1 a_1 \ldots$ автомата $\mathcal{A}_\omega$ из состояния q_0, бесконечно часто проходящая через некоторое финальное состояние $q \in F$. Обозначим $L(\mathcal{A}_\omega)$ множество последовательностей, принимаемых автоматом Бюхи $\mathcal{A}_\omega$.

Определение 6. Замкнутый ω-автомат – это автомат Бюхи $\mathcal{A}_{\omega c} = (Q, q_0, \Pi, \delta, F)$ такой, что $F = Q$.

Определение 7 (автоматная логика). Пусть P есть множество элементарных высказываний. Тогда множество формул логики $AL(P)$ определяется по следующей грамматике:

$$\varphi, \varphi_i ::= p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \exists \mathcal{A}_f(\varphi_1, \ldots, \varphi_m) \mid \forall \mathcal{A}_f(\varphi_1, \ldots, \varphi_m) \\ \mid \exists \mathcal{A}_\omega(\varphi_1, \ldots, \varphi_m) \mid \forall \mathcal{A}_\omega(\varphi_1, \ldots, \varphi_m),$$

где $\mathcal{A}_f$ (соответственно, $\mathcal{A}_\omega$) – конечный автомат (соответственно, автомат Бюхи) над алфавитом $\Lambda \times \Sigma$, $\Lambda = \{\varphi_1, \ldots, \varphi_m\}$.

Определение 8. Пусть $\mathcal{S} = (S, T, \to, s_0)$ – вполне структурированная система переходов автоматного типа. Определим отношение выполнимости между состоянием s системы $\mathcal{S}$ и формулой логики $AL(P)$ следующим образом ($\star$ используется вместо f и ω):

$$\begin{aligned}
s \models p &\iff s \in [\![p]\!] \\
s \models \neg\varphi &\iff s \not\models \varphi \\
s \models \varphi_1 \vee \varphi_2 &\iff s \models \varphi_1 \text{ или } s \models \varphi_2 \\
s \models \varphi_1 \wedge \varphi_2 &\iff s \models \varphi_1 \text{ и } s \models \varphi_2 \\
s \models \exists \mathcal{A}_\star(\varphi_1, \ldots, \varphi_m) &\iff \exists\, \pi = s_0 a_0 s_1 a_1 \ldots \in \mathrm{Run}(s, \mathcal{S}): \\
&\qquad \exists\, \sigma = q_0(\varphi_{i_0}, a_0) q_1(\varphi_{i_1}, a_1) \ldots \in L(\mathcal{A}_\star): \\
&\qquad \forall j : 0 \le j < |\sigma| : s_j \models \varphi_{i_j} \\
s \models \forall \mathcal{A}_\star(\varphi_1, \ldots, \varphi_m) &\iff \forall\, \pi = s_0 a_0 s_1 a_1 \ldots \in \mathrm{Run}(s, \mathcal{S}): \\
&\qquad \exists\, \sigma = q_0(\varphi_{i_0}, a_0) q_1(\varphi_{i_1}, a_1) \ldots \in L(\mathcal{A}_\star): \\
&\qquad \forall j : 0 \le j < |\sigma| : s_j \models \varphi_{i_j}
\end{aligned}$$

Для каждой формулы φ полагаем, что $[\![\varphi]\!]_{\mathcal{S}} = \{s \in S \mid s \models \varphi\}$. $\mathrm{Run}(s, \mathcal{S})$ обозначает множество исполнений системы $\mathcal{S}$ из состояния s.

Определение 9 (подмножества AL). Логики $\exists$AL(P) и $\forall$AL(P) – это подмножества логики AL, содержащие формулы, построенные из элементарных высказываний $p \in P$, операторов конъюнкции, дизъюнкции и соответственно кванторов существования и всеобщности. Если $\mathcal{L}$ – подмножество логики AL, тогда $\mathcal{L}_f$ (соответственно, $\mathcal{L}_\omega$, $\mathcal{L}_{\omega c}$) обозначается подмножество логики $\mathcal{L}$, где используются только конечные автоматы (соответственно, автоматы Бюхи, замкнутые ω-автоматы).

Будем обозначать P_{UC} множество элементарных высказываний, интерпретирующихся верхними конусами состояний системы переходов $\mathcal{S}$=$(S, T, \to, s_0)$, т. е. для любого $p \in P_{UC}$ выполняется условие $s \models p \Longrightarrow \forall s' \geq s : s' \models p$, где $s, s' \in S$, а P_{DC} – множество элементарных высказываний, интерпретирующихся нижними конусами, т. е. для любого $p \in P_{DC}$ выполняется условие, что $s \models p \Longrightarrow \forall s' \leq s : s' \models p$.

Определение 10.

Локальная проверка модели: для заданной модели (системы переходов) $\mathcal{S}$ определить, истинна ли формула φ в состоянии s, т. е. проверить для $\mathcal{S}$, выполняется ли $s \models \varphi$.

Глобальная проверка модели: построить множество $[\![\varphi]\!]_{\mathcal{S}}$ всех состояний модели $\mathcal{S}$, в которых истинна формула φ.

4. Разрешимые свойства

Утверждение 4. *Задача локальной проверки модели разрешима для вполне структурированных систем переходов автоматного типа и логики* $AL_f(P_{DC})$.

Доказательство. Проводится структурной индукцией по вложению подформул в формуле φ логики $AL_f(P_{DC})$.

Базис индукции. Рассмотрим разрешимость данной задачи для произвольного состояния s_0 системы $\mathcal{S}$ и формул вида $\exists\mathcal{A}_f(\varphi_1, \ldots, \varphi_n)$ и $\forall\mathcal{A}_f(\varphi_1, \ldots, \varphi_n)$, где φ_i – формула логики $AL_f(P_{DC})$, построенная только из элементарных высказываний, принадлежащих множеству P_{DC}, и допустимых булевских операторов. Поскольку пересечение и объединение нижних конусов являются также нижними конусами, будем считать формулы φ_i элементарными высказываниями, т. е. $\varphi_i = p_i \in P_{DC}$.

Итак, рассмотрим произвольную вполне структурированную систему переходов автоматного типа $\mathcal{S} = (S, T, \to, s_0)$, а также конечный автомат $\mathcal{A}_f = (\Sigma, Q, q_0, \delta, F)$, где $\Sigma = P_{DC} \times T$. Построим вполне структурированну систему переходов $\mathcal{S}' = (S', T', \to', s'_0)$, представляющую собой произведение системы $\mathcal{S}$ и конечного автомата $\mathcal{A}_f$. Состояния системы $\mathcal{S}'$ имеют вид $s' = (s, q)$, где s – состояние системы $\mathcal{S}$, q – состояние автомата $\mathcal{A}_f$. Зададим на множестве $\mathcal{S}'$ вполне упорядочиваемый квазипорядок $\leq$ таким образом, что $s'_1 = (s_1, q_1) \leq s'_2 = (s_2, q_2) \iff (s_1 \leq s_2) \wedge (q_1 = q_2)$. Начальное состояние $s'_0 = (s_0, q_0)$. Состояние $s' = (s, q)$ системы $\mathcal{S}'$ является финальным, если состояние q автомата $\mathcal{A}_f$ финальное, т. е. $q \in F$. Переход $t \in T'$ из состояния $s'_1 = (s_1, q_1)$ в некоторое

состояние $s'_2 = (s_2, q_2)$ существует тогда и только тогда, когда одновременно существуют переходы $s_1 \xrightarrow{t} s_2$, $q_1 \overset{(\varphi,t)}{\rightarrow} q_2$ и $s_1 \models \varphi$.

Для системы $\mathcal{S}'$ построим дерево покрытия с корнем в начальном состоянии s'_0. Состояние s_0 системы $\mathcal{S}$ будет удовлетворять формуле $\varphi = \exists\mathcal{A}_f(p_1, \ldots, p_n)$ тогда и только тогда, когда в дереве покрытия будет хотя бы одно финальное состояние. Действительно, поскольку элементарные высказывания интерпретируются нижними конусами, если из дерева покрытия достраивать дерево достижимости, то новые состояния будут только покрывающими (новых состояний может и не быть в случае тупикового листа) и, поэтому, будут принадлежать к тому типу, к какому принадлежат покрываемые состояния. Следовательно, если в дереве покрытия не будет ни одного финального состояния, то не будет финальных состояний и во всем дереве достижимости.

Состояние s_0 системы $\mathcal{S}$ будет удовлетворять формуле $\varphi = \forall\mathcal{A}_f(p_1, \ldots, p_n)$ тогда и только тогда, когда в дереве покрытия с корнем в начальном состоянии s'_0 системы $\mathcal{S}'$ будет существовать поддерево с тем же корнем, удовлетворяющее следующим свойствам. Во-первых, каждое состояние поддерева $s' = (s_1, q_1)$ для всех состояний s_2 и переходов t системы $\mathcal{S}$ таких, что $s_1 \xrightarrow{t} s_2$, должно иметь переходы $(s_1, q_1) \xrightarrow{t} (s_2, q_2)$. Во-вторых, каждая ветка поддерева должна содержать финальное состояние. Действительно, первое требование необходимо для выполнения условия всеобщности. Нарушение второго требования хотя бы для одной ветки, означает наличие конечного или бесконечного исполнения системы $\mathcal{S}'$, не содержащего в себе финальных состояний, т. е. исполнение системы $\mathcal{S}$ не принимается.

Необходимо отметить, что, так как все элементарные высказывания интерпретируются нижними конусами, если для состояния s_0 формула φ вида $\exists\mathcal{A}_f(p_1, \ldots, p_n)$ или $\forall\mathcal{A}_f(p_1, \ldots, p_n)$ не выполнена, т. е. $s_0 \not\models \varphi$, то эта формула будет также не верна и для любого состояния $s \geq s_0$.

Предположение индукции. Рассмотрим автомат $\mathcal{A}_f(\varphi_1, \ldots, \varphi_n)$ с алфавитом $\Sigma = \{\varphi_1, \ldots, \varphi_n\} \times T$, где φ_i – произвольные формулы логики $AL_f(P_{DC})$. Предполагая, что для произвольного состояния s системы $\mathcal{S}$ и формулы φ_i существует алгоритм проверки $s \models \varphi_i$, и проводя построения и рассуждения аналогичные указанным выше получим алгоритм проверки формул в общем случае. □

Утверждение 5. *Задача локальной проверки модели разрешима для вполне структурированных систем переходов автоматного типа и логики* $AL_{\omega c}(P_{UC})$.

Доказательство. Проводится аналогично доказательству утверждения 4 структурной индукцией по вложению подформул в формуле φ логики $AL_{\omega c}(P_{UC})$.

Базис индукции. Рассмотрим разрешимость данной задачи для произвольного состояния s_0 системы $\mathcal{S}$ и формул вида $\exists\mathcal{A}_{\omega c}(p_1, \ldots, p_n)$ и $\forall\mathcal{A}_{\omega c}(p_1, \ldots, p_n)$, где $p_i \in P_{UC}$.

Как и раньше (см. доказательство утверждения 4) построим вполне структурированную систему переходов $\mathcal{S}' = (S', T', \rightarrow', S'_0)$, представляющую

собой произведение системы $\mathcal{S} = (S, T, \to, s_0)$ и замкнутого ω-автомата $\mathcal{A}_{\omega c} = (\Sigma, Q, q_0, \delta, F{=}Q)$, где $\Sigma = P_{UC} \times T$. Затем, для системы $\mathcal{S}'$ построим дерево покрытия с корнем в начальном состоянии $s'_0 = (s_0, q_0)$.

Состояние s_0 системы переходов $\mathcal{S}$ будет удовлетворять формуле

$$\varphi = \exists \mathcal{A}_{\omega c}(p_1, \ldots, p_n)$$

тогда и только тогда, когда в дереве покрытия будет существовать лист, представляющий собой покрывающее состояние для некоторого предыдущего состояния на этой же ветке. Действительно, поскольку элементарные высказывания интерпретируются верхними конусами, наличие такого покрывающего состояния (листа) обеспечивает существование бесконечного пути, состоящего из финальных состояний. Все состояния этого пути будут покрывающими и, следовательно, будут удовлетворять тому же элементарному высказыванию, что и покрываемое состояние. Если же не все листья дерева покрытия являются покрывающими, т. е. существуют тупиковые состояния, которые не имеют потомков, это означает, что не существует бесконечных исполнений системы $\mathcal{S}$, удовлетворяющих формуле φ.

Состояние s_0 системы переходов $\mathcal{S}$ будет удовлетворять формуле

$$\varphi = \forall \mathcal{A}_{\omega c}(p_1, \ldots, p_n)$$

тогда и только тогда, когда в дереве покрытия с корнем в начальном состоянии системы $\mathcal{S}'$ будет существовать поддерево с тем же корнем, удовлетворяющие следующим свойствам. Во-первых, каждое состояние поддерева $s'_1 = (s_1, q_1)$ для всех состояний s_2 и переходов t системы $\mathcal{S}$ таких, что $s_1 \xrightarrow{t} s_2$, должно иметь переходы $(s_1, q_1) \xrightarrow{t} (s_2, q_2)$. Во-вторых, каждая ветка поддерева должна заканчиваться покрывающим состоянием, т. е. не должно быть ни одного тупикового листа. Действительно, первое требование необходимо для выполнения условия всеобщности. Нарушение второго требования хотя бы для одной ветки, означает наличие конечного исполнения системы $\mathcal{S}'$, т. е. некоторое бесконечное исполнение системы $\mathcal{S}$ в этом направлении (этой веткой) не принимается.

Необходимо отметить, что, так как все элементарные высказывания интерпретируются верхними конусами, если для состояния s_0 формула φ вида $\exists \mathcal{A}_{\omega c}(p_1, \ldots, p_n)$ или $\forall \mathcal{A}_{\omega c}(p_1, \ldots, p_n)$ выполнена, т. е. $s_0 \models \varphi$, то эта формула будет также верна и для любого состояния $s \geq s_0$.

Предположение индукции. Рассмотрим автомат $\mathcal{A}_{\omega c}(\varphi_1, \ldots, \varphi_n)$ с алфавитом $\Sigma{=}\{\varphi_1, \ldots, \varphi_n\} \times T$, где φ_i – произвольные формулы логики $AL_{\omega c}(P_{UC})$. Предполагая, что для произвольного состояния s системы $\mathcal{S}$ и формулы φ_i существует алгоритм проверки $s \models \varphi_i$, и проводя построения и рассуждения аналогичные указанным выше получим алгоритм проверки формул в общем случае. □

Утверждение 6. *Задача глобальной проверки модели является разрешимой для вполне структурированных систем переходов автоматного типа и формул логики $A_f(P_{UC})$ вида $\exists \mathcal{A}_f(p_1, \ldots, p_n)$.*

Доказательство. Рассмотрим произвольную вполне структурированную систему переходов автоматного типа $\mathcal{S} = (S, T, \to, s_0)$, а также конечный автомат $\mathcal{A}_f = (\Sigma, Q, q_0, \delta, F)$ с алфавитом $\Sigma = \{p_1, \ldots, p_n\} \times T$, где $p_i \in P_{UC}$. Как и раньше построим вполне структурированную систему переходов $\mathcal{S}' = (S', T', \to', s'_0)$, представляющую собой произведение системы $\mathcal{S}$ и конечного автомата $\mathcal{A}_f$.

Рассмотрим множество S'_F финальных состояний системы $\mathcal{S}'$, где $S'_F = \{(s, q) \mid s \in S,\ q \in F\}$. Множество S'_F представляет собой верхний конус. Для S'_F построим множество состояний–предшественников $\mathrm{Pred}(S'_F)$. Множество $\mathrm{Pred}(S'_F)$ также является верхним конусом. Построим последовательность S'_0, S'_1, …, S'_k, где $S'_0 = S'_F$, $S'_{i+1} = S'_i \cup S''_i$, а S''_i представляет собой множество $\mathrm{Pred}(S'_i)$ состояний. Из утверждения 2 следует, что эта последовательность верхних конусов стабилизируется при некотором k. Выберем из множества S'_k только те состояния, в которых второй компонент является начальным состоянием конечного автомата $\mathcal{A}_f$, и спроецируем полученное множество на множество состояний системы $\mathcal{S}$. Получим верхний конус $[\![\varphi]\!]_{\mathcal{S}}$ состояний, которые удовлетворяют формуле $\varphi = \exists \mathcal{A}_f(p_1, \ldots, p_n)$.

5. Неразрешимые свойства

Рассмотрим машину Минского с двумя счетчиками $2cM$. Построим для машины $2cM$ ее “слабую” модель $2cMw$ следующим образом. Добавим оператор недетерминированного выбора [] и в $2cM$ заменим каждое правило перехода `if` $x_j > 0$ `then` $\mathtt{comm}_1$ `else` $\mathtt{comm}_2$ на правило $\mathtt{comm}_1$ [] $\mathtt{comm}_2$. В выражении $x_j := x_j - 1$ оператор ‘$-$’ заменим оператором вычитания до нуля ‘$\ominus$’. Таким образом, мы получили недетерминированную машину $2cMw$.

Рассмотрим $2cMw$ как помеченную систему переходов. Для каждого счетчика x_j все переходы (из состояний первого типа) с выражением $x_j := x_j + 1$ пометим как inc_j, переходы (из состояний второго типа) с выражением $x_j := x_j \ominus 1$ как dec_j, а переходы, не изменяющие значения счетчика x_j, обозначим nul_j ($j = 1, 2$). Если переходы inc_j, dec_j, nul_j переводят машину в финальное состояние, то обозначим (переименуем) их соответственно $halt_j^+$, $halt_j^-$, $halt_j^0$. В соответствии с новыми правилами переходов определим отношение переходов на множестве конфигураций $\to \in S \times T \times S$, где $T = \{inc_1, inc_2, dec_1, dec_2, nul_1, nul_2\} \cup \{halt_1^+, halt_2^+, halt_1^-, halt_2^-, halt_1^0, halt_2^0\}$.

Зададим естественным образом отношение $\leq$ частичного порядка на множестве конфигураций S машины $2cMw$. Для двух конфигураций системы $2cMw$ имеем: $(q, c_1, c_2) \leq (q', c'_1, c'_2)$, если $q = q'$ и $c_i \leq c'_i$, $i = 1, 2$. Это отношение является вполне упорядочиваемым квазипорядком по лемме Диксона.

Очевидно, что система переходов $(2cMw, \leq)$ является вполне структурированной системой переходов автоматного типа.

Утверждение 7. *Задача проверки модели является неразрешимой для вполне структурированной системы переходов автоматного типа и автоматной логики $\exists AL_f(P_{UC}, P_{DC})$.*

Доказательство. Предположим противное. Допустим, что существует алгоритм позволяющий ответить на вопрос о разрешимости произвольной формулы

φ автоматной логики $\exists AL_f(P_{UC}, P_{DC})$ для произвольной системы автоматного типа.

Рассмотрим двухсчетчиковую машину Минского $2cM$ с начальной конфигурацией $(q_0, 0, 0)$ и построенную на ее основе недетерминированную систему переходов $2cMw$.

Для доказательства утверждения 7 достаточно построить конечный автомат $\mathcal{A}_f$ такой, что выполнимость формулы автоматной логики $\exists A_f(P_{UC}, P_{DC})$ для состояния s_0, т. е. $s_0 \models \exists A_f(P_{UC}, P_{DC})$, означала бы существование у системы $2cMw$ исполнения, равного конечному исполнению машины Минского $2cM$.

Итак, построим такой конечный автомат $\mathcal{A}_f = (\Sigma, Q, q_0, \delta, F)$ над алфавитом Σ, где

$$\Sigma = \{(p_j, inc_j), (p_j^+, dec_j), (p_j^-, nul_j), (p_j, halt_j^+), (p_j^+, halt_j^-), (p_j^-, halt_j^0); j = 1, 2\},$$
$$P_{UC} = \{p_1^+, p_2^+\} \cup \{p_1, p_2\}, \qquad P_{DC} = \{p_1^-, p_2^-\}, \qquad p_j^+ = \text{“}x_j > 0\text{”},$$
$$p_j^- = \text{“}x_j \le 0\text{”}, \qquad p_j = p_j^+ \vee p_j^- = \text{“}x_j \ge 0\text{”} \qquad (j = 1, 2).$$

Множество состояний $Q = \{q_0, q_f\}$, где q_0 – начальное состояние, а $F = \{q_f\}$. Функцию переходов определим следующим образом:

$$\begin{aligned}
\delta(q_0, (p_1, inc_1)) &= \{q_0\}, \qquad \delta(q_0, (p_2, inc_2)) = \{q_0\},\\
\delta(q_0, (p_1^+, dec_1)) &= \{q_0\}, \qquad \delta(q_0, (p_2^+, dec_2)) = \{q_0\},\\
\delta(q_0, (p_1^-, nul_1)) &= \{q_0\}, \qquad \delta(q_0, (p_2^-, nul_2)) = \{q_0\},
\end{aligned}$$

а также

$$\begin{aligned}
\delta(q_0, (p_1, halt_1^+)) &= \{q_f\}, \qquad \delta(q_0, (p_2, halt_2^+)) = \{q_f\},\\
\delta(q_0, (p_1^+, halt_1^-)) &= \{q_f\}, \qquad \delta(q_0, (p_2^+, halt_2^-)) = \{q_f\},\\
\delta(q_0, (p_1^-, halt_1^0)) &= \{q_f\}, \qquad \delta(q_0, (p_2^-, halt_2^0)) = \{q_f\}.
\end{aligned}$$

По построению конечный автомат $\mathcal{A}_f$ допускает только те переходы системы $2cMw$, которые соответствуют переходам машины Минского $2cM$, и формула $\exists A_f(P_{UC}, P_{DC})$ будет истинной для $2cMw$ тогда и только тогда, когда машина Минского останавливается. Следовательно, если существует алгоритм проверки истинности данной формулы $\exists A_f(P_{UC}, P_{DC})$ для системы $2cMw$, то существует алгоритм решения задачи останова машины Минского. Пришли к противоречию.

Рассмотрим *счетчиковую машину с обнулениями* rcM [5], у которой правило переходов для состояний q_i второго типа имеет вид:

$$\delta_i : \texttt{if } x_j > 0 \texttt{ then } (x_j := x_j - 1; \texttt{ goto } q_k) \,[]\, x_j := 0; \texttt{ goto } q_{k'},$$

где [] – оператор недетерминированного выбора. Счетчиковая машина с обнулениями (RCM – reset counter machine) принадежит к классу *счетчиковых машин с потерями* [5].

Необходимо отметить, что *для счетчиковых машин с потерями задача существования исполнения в состояние, из которого существует бесконечное исполнение, является неразрешимой* [5].

Построим для счетчиковой машины с обнулениями rcM ее "слабую" модель $rcMw$ следующим образом. В машине rcM заменим в правилах переходов выражение $x_j := x_j - 1$ на $x_j := x_j \ominus 1$. Как и раньше рассмотрим недетерминированную счетчиковую машину $rcMw$ как вполне структурированную систему переходов автоматного типа. Для каждого счетчика x_j все переходы (из состояний первого типа) с выражением $x_j := x_j + 1$ пометим как inc_j, переходы (из состояний второго типа) с выражением $x_j := x_j \ominus 1$ как dec_j, а переходы с выражением $x_j := 0$ обозначим nul_j $(j = 1, \ldots, m)$. В соответствии с новыми правилами переходов определим отношение переходов на множестве конфигураций $\rightarrow \in S \times T \times S$, где T – множество, состоящее из меток переходов inc_j, dec_j, nul_j $(j = 1, \ldots, m)$.

Утверждение 8. *Задача проверки модели является неразрешимой для вполне структурированной системы переходов автоматного типа и автоматной логики* $\exists AL_\omega(P_{UC})$.

Доказательство. Рассмотрим m-счетчиковую машину с обнулениями rcM с начальной конфигурацией $(q_0, 0, \ldots, 0)$ и построенную на ее основе недетерминированную систему переходов $rcMw$.

Для доказательства утверждения 8 достаточно построить автомат Бюхи $\mathcal{A}_\omega$ такой, что выполнимость формулы автоматной логики $\exists A_\omega(P_{UC})$ для начального состояния s_0, т. е. $s_0 \models \exists A_\omega(P_{UC})$, означала бы существование у системы $rcMw$ исполнения, равного исполнению машины rcM, существование которого проверить алгоритмически невозможно.

Итак, построим автомат Бюхи $\mathcal{A}_\omega = (\Sigma, Q, q_0, \delta, F)$ над алфавитом $\Sigma = P_{UC} \times T$, где

$$P_{UC} = \{true, p_1^+, \ldots, p_m^+\}, \qquad p_j^+ = \text{“}x_j > 0\text{”} \quad (j = 1, \ldots, m),$$
$$T = \{inc_1, \ldots, inc_m, dec_1, \ldots, dec_m, nul_1, \ldots, nul_m\}.$$

Множество состояний $Q = \{q_0, q_f\}$, где q_0 – начальное состояние, а $F = \{q_f\}$.

Функция переходов определяется следующим образом $(j = 1, \ldots, m)$:

$$\delta(q_0, (true, inc_j)) = Q, \qquad \delta(q_0, (true, nul_j)) = Q, \qquad \delta(q_0, (p_j^+, dec_j)) = Q,$$
$$\delta(q_f, (true, inc_j)) = F, \qquad \delta(q_f, (true, nul_j)) = F, \qquad \delta(q_f, (p_j^+, dec_j) = F.$$

По построению автомат Бюхи $\mathcal{A}_\omega$ допускает только те переходы системы $rcMw$, которые соответствуют переходам машины rcM. И следовательно, автомат $\mathcal{A}_\omega$ допускает для машины rcM существование исполнения в конфигурацию, из которой существует бесконечное исполнение. Задача существования такого исполнения для счетчиковых машин с потерями является алгоритмически неразрешимой [5]. □

Список литературы

[1] Bouajjani A., Mayr R. Model checking lossy vector addition systems // Proc. STACS'99. LNCS 1563. – 1999.

[2] Finkel A. Reduction and covering of infinite reachability trees // Information and Computation. – 1990. – Vol. 89(2). – P. 144–179.

[3] Finkel A., Schnoebelen Ph. Well-structured transition systems everywhere! // Theoretical Computer Science. – 2001. – Vol. 256(1–2). – P. 63–92.

[4] Kouzmin E., Sokolov V. Communicating colouring automata // Proc. International Workshop on Program Understanding. – 2003. – P. 40–46.

[5] Mayr R. Lossy counter machines // Tech. Report TUM-I9830. – Munich, Germany: Institut für Informatik, October 1998.

[6] Stirling C. P. Modal and temporal logics // Handbook of Logic in Computer Science. – Oxford University Press, 1992. – Vol. 2. – P. 477–563.

[7] Stirling C. P. Modal and temporal logics for processes // LNCS 1043. – 1996. – P. 149–237.

[8] Vardi M.Y., Wolper P. Reasoning about infinite computations // Information and Computation. – 1994. – Vol. 115(1). – P. 1–37.

[9] Wolper P. Temporal logic can be more expressive // Information and Control. – 1983. № 56.

[10] Кузьмин Е. О разрешимости задачи проверки модели для модального μ-исчисления и некоторых классов вполне структурированных систем переходов // Моделирование и анализ информационных систем. – 2003. – Т. 10. № 1. – С. 50–55.

[11] Минский М. Вычисления и автоматы. – М.: Мир, 1971.

[12] Хопкрофт Д., Мотвани Р., Ульман Д. Введение в теорию автоматов, языков и вычислений. 2-е изд. – М.: Вильямс, 2002.