Модел. и анализ информ. систем. Т. **17**, № **1** (2010) 52–64

УДК 519.7

О множестве достижимости автоматных счетчиковых машин

Кузьмин Е. В., Чалый Д. Ю. [1]

Ярославский государственный университет им. П.Г. Демидова

e-mail: {kuzmin,chaly}@uniyar.ac.ru

получена 15 декабря 2009

Ключевые слова: абстрактные счетчиковые машины, автоматные счетчиковые машины, взаимодействующие раскрашивающие автоматы, множества достижимости, полулинейные множества

Исследуются свойства автоматных счетчиковых машин. Доказывается, что множество достижимых состояний любой автоматной односчетчиковой машины является полулинейным множеством. Приводится алгоритм построения этого множества. Кроме того, показывается, что множество достижимости любой автоматной счетчиковой машины с ограничением на количество перемен направлений роста/убывания значений счетчиков и множество достижимости любой плоской автоматной счетчиковой машины также полулинейны.

1. Введение

В последние годы для моделирования и анализа вычислительных систем (программных и аппаратных) все чаще применяются подходы, связанные с верификацией систем с бесконечным числом состояний. Для реализации процедуры автоматической верификации предлагаются и изучаются модели (систем с бесконечным числом состояний) различных классов, среди которых можно выделить счетчиковые машины, системы с ненадежными каналами передачи данных, магазинные автоматы, временные автоматы и т. п. Существует ряд программных средств верификации, таких как FAST [3], LASH [12] и TREX [1], базирующихся на формализме счетчиковых систем, представляющих собой конечный автомат, расширенный с помощью операций над целочисленными переменными (счетчиками).

Одной из главных проблем верификации (в частности, счетчиковых систем) является задача достижимости заданного состояния системы. Известно, что для счетчиковых машин Минского [22], представляющих собой наиболее мощный класс счетчиковых систем, в котором допускается безусловное увеличение значения счетчика

[1]Работа проводилась при финансовой поддержке Федерального агентства по науке и инновациям, ФЦП «Научные и научно-педагогические кадры инновационной России на 2009-2013 годы».

на единицу и условное вычитание единицы до нуля (если значение счетчика равно нулю, происходит переход в альтернативное состояние), проблема достижимости не является разрешимой даже в случае всего лишь двух счетчиков. В связи с этим с целью получить разрешимость проблемы достижимости были предложены различные ослабления для счетчиковых систем, такие как недетерминизм переходов, невозможность совершать проверку на нуль, ограничения на число счетчиков и т. д. Например, системы векторного сложения с состояниями, или сети Петри, можно рассматривать как специальный «слабый» класс счетчиковых машин, машины в рамках которого, имея недетерминизм переходов, не могут осуществлять проверку на нуль, для которого проблема достижимости разрешима [15].

Другой важной (но более общей) проблемой является задача построения множества достижимых состояний изучаемой системы. Ранее было найдено несколько классов счетчиковых систем с полулинейным множеством достижимости. Например, для сетей Петри выделяют следующие подклассы с полулинейным множеством достижимости: BPP-сети (BPP — Basic Parallel Processes) [4], циклические сети Петри [2], устойчивые сети Петри [14], регулярные сети Петри [17], 2-мерные системы векторного сложения с состояниями [7]. Полулинейность множества достижимости позволяет говорить о разрешимости проблемы достижимости заданного состояния, а также о разрешимости классических проблем включения и эквивалентности множеств достижимости, поскольку, например, каждая из указанных проблем может быть легко задана формулой арифметики Пресбургера на основе полученного конечного представления множества достижимости (известно [16], что арифметика Пресбургера разрешима, т. е. существует алгоритм, определяющий истинность для любой формулы арифметики Пресбургера).

В данной работе исследуются свойства класса автоматных счетчиковых машин [11, 20], которые «лежат в основе» (моделируются с помощью) взаимодействующих раскрашивающих автоматов, введенных в [10, 19] как средство моделирования перемещения данных различного типа между компонентами распределенной системы. Основное отличие этого формализма от других заключается в переносе управления над данными на дуги переходов. В автоматных счетчиковых машинах каждый переход определяется недетерминированно в соответствии с локальными управляющими состояниями и независимо от манипулируемых данных.

Известно [20, 11], что для автоматных счетчиковых машин проблемы включения и эквивалентности неразрешимы, а также неразрешимы проблема ограниченности множества достижимости для автоматных машин с тремя счетчиками и проблема достижимости ненулевого вектора значений счетчиков для автоматных четырехсчетчиковых машин (однако разрешима проблема достижимости нулевого вектора для произвольной автоматной счетчиковой машины). Проблемы ограниченности и достижимости для автоматных двухсчетчиковых машин остаются открытыми. В работе [21] было показано, что существуют автоматные трехсчетчиковые машины, множество достижимости которых не является полулинейными. Является ли множество достижимости любой автоматной двухсчетчиковой машины полулинейным — открытая проблема.

Поскольку для автоматных трехсчетчиковых машин множество достижимости не является полулинейным, а исследование машин с двумя счетчиками пока не привело ни к положительному, ни к отрицательному результату, остается рассмотреть случай автоматных односчетчиковых машин. Кроме того, помимо ослаблений для классов систем, связанных с уменьшением количества счетчиков, часто исследуют классы счетчиковых систем с ограничениями на число перемен направлений роста/убывания значений счетчиков (см., например, [8, 6]).

В этой работе мы покажем, что множество достижимости автоматных односчетчиковых машин является полулинейным. Поскольку автоматные счетчиковые машины за один шаг во время исполнения могут увеличивать или уменьшать значения счетчиков только лишь на единицу, то для построения множества достижимости автоматной односчетчиковой машины достаточно найти минимальное, максимальное значение счетчика или показать его неограниченность. Мы несколько усложним задачу. Следуя [7], предлагается алгоритм построения множества достижимости значений счетчика в каждом локальном состоянии машины, т. е. для каждого локального состояния машины мы построим (полулинейное) множество значений счетчиков, которое может быть достигнуто в этом локальном состоянии.

В статье [6] изучался специальный класс расширенных n-счетчиковых машин с ограничением на количество перемен направлений роста/убывания значений счетчиков. В ней было показано, что множество достижимости машин этого класса представляет собой полулинейное множество. Поскольку, очевидно, этот класс машин включает в себя класс автоматных счетчиковых машин с ограничением на количество перемен направлений роста/убывания значений счетчиков, то все положительные результаты (в том числе про полулинейность множества достижимости) могут быть распространены и на последний подкласс счетчиковых машин. Однако в работах [8, 6] было доказано, что проблема определения, является ли произвольная n-счетчиковая машина Минского ограниченной по числу перемен направлений роста/убывания значений счетчиков, неразрешима. В [6] утверждается, что, например, для систем векторного сложения с состояниями указанная проблема разрешима. Является ли разрешимой эта задача для произвольной автоматной счетчиковой машины — открытая проблема. Однако из работ [5, 13] фактически следует, что если граф управления некоторой автоматной n-счетчиковой машины не содержит вложенных циклов, т. е. автоматная счетчиковая машина является «плоской», то множество достижимости такой машины полулинейно.

2. Основные понятия и определения

Абстрактная счётчиковая машина M — это набор элементов $(v_0, q_0, Q, X, \to, T)$, где $X = \{x_1, \ldots, x_m\}$ — конечное множество счетчиков, $Q = \{q_0, \ldots, q_n\}$ — конечное множество состояний, q_0 — начальное состояние, q_n — финальное (заключительное) состояние, v_0 — вектор начальных значений счетчиков, T — конечное множество меток действий, соответствующих выражению (преобразованию) над счетчиками, $\to \subseteq Q \setminus \{q_n\} \times T \times Q$ — отношение переходов, запись $q \xrightarrow{t} q'$ используется для обозначения (правила) перехода $(q, t, q') \in \to$.

Автоматная счётчиковая машина AM — это абстрактная счетчиковая машина, у которой переход из одной конфигурации в другую может быть одного из следующих видов (при $q \xrightarrow{t} q'$):

если t соответствует выражению $x_i := x_i + 1$, тогда

$$(q, c_1, \dots, c_i, \dots, c_m) \xrightarrow{t} (q', c_1, \dots, c_i + 1, \dots, c_m);$$

если метка t соответствует выражению $x_i := x_i \ominus 1$, тогда

$$(q, c_1, \dots, c_i, \dots, c_m) \xrightarrow{t} (q', c_1, \dots, c_i', \dots, c_m),$$

где $c_i' = 0$ при $c_i = 0$ и $c_i' = c_i - 1$ при $c_i > 0$;

если метка t соответствует выражению $x_i := x_i + \min(x_j, 1)$, тогда

$$(q, c_1, \dots, c_i, \dots, c_m) \xrightarrow{t} (q', c_1, \dots, c_i', \dots, c_m),$$

где $c_i' = c_i + 1$ при $c_j \geq 1$ и $c_i' = c_i$ при $c_j = 0$;

если t соответствует выражению $x_i := x_i$, тогда

$$(q, c_1, \dots, c_i, \dots, c_m) \xrightarrow{t} (q', c_1, \dots, c_i, \dots, c_m).$$

Для удобства правила переходов будем представлять в виде $q \to (q', w)$, означающем, что существует переход из состояния q в q', при котором к вектору старых значений счетчиков прибавляется вектор w. Например, правила переходов, соответствующие выражениям $x_i := x_i + \min(x_j, 1)$ и $x_i := x_i \ominus 1$ можно переписать соответственно как

$$q \to (q', (0, \dots, 0, \min(x_j, 1), 0, \dots, 0))$$

(выражения $\min(x_j, 1)$ и $\ominus 1$ стоят в i-х позициях) и

$$q \to (q', (0, \dots, 0, \ominus 1, 0, \dots, 0)),$$

где $\ominus 1$ — вычитание единицы до нуля, а $\min(x_j, 1)$ — функция, возвращающая минимум из единицы и значения счетчика x_j.

Множеством достижимости, или *сферой достижимости*, $R(\mathcal{M})$ автоматной счетчиковой машины АМ $\mathcal{M}$ назовем множество всех векторов значений счетчиков, достижимых машиной из начальной конфигурации (q_0, v_0), где q_0 — начальное состояние, v_0 — вектор начальных значений счетчиков.

Таким образом, вектор принадлежит множеству $R(\mathcal{M})$, если он достижим из начального вектора v_0 (и начального состояния q_0) последовательностью переходов.

Конфигурация счетчиковой машины представляет собой набор $(q_i, c_1, \dots, c_m)$, где q_i — состояние машины, $c_i \in \mathbb{N}$ — значение соответствующего счетчика ($\mathbb{N}$ — множество натуральных чисел, включая ноль).

Исполнение счетчиковой машины — это последовательность конфигураций $s_0 \to s_1 \to s_2 \to \dots$ с начальной конфигурацией $s_0 = (q_0, v_0)$, индуктивно определяемая в соответствии с правилами переходов.

Пусть $L \subseteq \mathbb{N}^n$ — произвольное множество векторов. Вектор $p \in \mathbb{N}^n$ называется *периодом* множества L, если для любого $r \in L$ вектор $r + p$ также принадлежит множеству L.

Пусть C и P — подмножества множества $\mathbb{N}^n$ и $\mathcal{L}(C;P)$ — множество всех векторов из $\mathbb{N}^n$, представимых в виде $c+p_1+\dots+p_m$, где $c \in C$ и $p_1,\dots,p_m$ — некоторая (возможно, пустая) последовательность векторов из P, т. е.

$$\mathcal{L}(C;P)=\{x \mid \exists\, c\in C, k_1,\dots,k_m\in\mathbb{N},\, p_1,\dots,p_m\in P\colon x=c+\sum_{i=1}^{m}k_ip_i\}.$$

Тогда множества C и P называются соответственно *системой предпериодов* и *системой периодов* множества $\mathcal{L}(C;P)$.

Итак, $\mathcal{L}(C;P)$ — множество всех x из множества $\mathbb{N}^n$, таких что $x=c+\sum_{i=1}^{m}k_ip_i$, где $c\in C$, $p_1,\dots,p_m\in P$ и $k_1,\dots,k_m\in\mathbb{N}$.

Если множество C состоит в точности из одного элемента, т. е. $C=\{c\}$, то для удобства вместо $\mathcal{L}(C;P)$ будем писать $\mathcal{L}(c;P)$.

Множество $\mathcal{L}\subseteq\mathbb{N}^n$ называется *линейным*, если $\mathcal{L}=\mathcal{L}(c;P)$, где система периодов P конечна, $P=\{p_1,\dots,p_r\}$, а система предпериодов состоит в точности из одного элемента c, т. е. $C=\{c\}$. В этом случае вместо $\mathcal{L}(C;P)$ будем писать $\mathcal{L}(c;p_1,\dots,p_r)$.

Всякий раз, когда говорится, что множество $\mathcal{L}(C;P)$ линейно, имеется в виду, что P конечно.

Подмножество множества $\mathbb{N}^n$ называется *полулинейным*, если оно является объединением конечного числа линейных множеств.

Например, пусть c_1, c_2, p_1, $\dots$, p_m, q_1, $\dots$, $q_r\in\mathbb{N}^n$, тогда множество

$$\mathcal{L}(c_1;p_1,\dots,p_m)\cup\mathcal{L}(c_2;q_1,\dots,q_r)$$

является полулинейным подмножеством множества $\mathbb{N}^n$.

Объединение конечного числа полулинейных подмножеств множества $\mathbb{N}^n$ также является полулинейным множеством. Более того, класс полулинейных множеств замкнут относительно операций пересечения и дополнения [18, 7].

3. Автоматные односчетчиковые машины

Коротким путем назовем последовательность переходов автоматной односчетчиковой машины $1cAM$ (из некоторой конфигурации) без повторяющихся состояний за исключением того, что состояния первой и последней конфигураций этого исполнения совпадают.

Отметим, что существует лишь конечное число коротких путей.

Короткий положительный путь — это короткий путь с позитивной разностью между значениями счетчика первой и последней конфигурации пути.

Плохой конфигурацией на пути назовем конфигурацию с нулевым значением счетчика, полученную из другой нулевой конфигурации посредством перехода с выражением $x:=x+\min(x,1)$ или $x:=x\ominus 1$.

Хороший путь — путь, не содержащий плохих конфигураций.

Приведенный ниже алгоритм осуществляет построение дерева, вершины которого помечаются тройками вида $[c, q, A_c]$, где $c \in \mathbb{N}$, $A_c \subseteq \mathbb{N}$, а q — состояние. Метка $[c, q, A_c]$ обозначает тот факт, что каждая точка линейного множества $\mathcal{L}(c; A_c)$ может быть достигнута автоматной односчетчиковой машиной $1cAM$ в локальном состоянии q.

Алгоритм.

1. *Вход*: автоматная односчетчиковая машина $1cAM = (c_0, q_0, Q, \{x\} \to, T)$.
2. $A_{c_0} := \emptyset$.
3. Создать корневую вершину дерева, помеченную $[c_0, q_0, A_{c_0}]$.
4. Повторять до тех пор, пока существует хотя бы один необработанный, т. е. не имеющий маркировки, лист текущего дерева.
   a) Выбрать необработанный лист текущего дерева, имеющий некоторую метку $[c, q, A_c]$.
   b) Добавить к множеству A_c только те значения-разности хороших коротких положительных путей, выходящих из конфигурации (q, c), которые не являются линейной комбинацией одномерных векторов из A_c.
   c) Если существует предок $[d, q, A_d]$ вершины $[c, q, A_c]$, такой что между ними в дереве нет плохой вершины и $c - d > 0$, то добавить к A_c элемент $k = c - d$ при условии, что k не является линейной комбинацией элементов из A_c.
   d) Если существует хороший короткий *неположительный* путь из q в q с разностью v ($v < 0$), то добавить к множеству A_c все $k = (k_1 u_1 + \cdots + k_m u_m) + v$, соответствующие минимальному набору $(k_1, \ldots, k_m)$, такому что $k > 0$, где $k_i \in \mathbb{N}$, $A_c = \{u_1, \ldots, u_m\}$, при условии, что k не является линейной комбинацией элементов из A_c.
   e) Если существует такой предок $[d, q, A_d]$ вершины $[c, q, A_c]$, что $\mathcal{L}(d; A_d)$ содержит c и $A_d = A_c$, тогда пометить вершину $[c, q, A_c]$ как обработанный лист, т. е. поставить маркировку;
   f) в противном случае для каждого перехода $q \to (q', w)$ машины $1cAM$ выполнить следующее.

      Если $(q', c + w)$ — хорошая конфигурация, то построить дочернюю вершину с пометкой $[d, q', A_d]$, где $d = c + w$ и $A_d = A_c$. Если $(q', c + w)$ — плохая конфигурация, рассмотреть следующие варианты. Если $w = \ominus 1$, тогда построить дочернюю «плохую» вершину с пометкой $[0, q', \emptyset]$ и для каждого $u \in A_c$ построить дочернюю «хорошую» вершину с пометкой $[d, q', A_d]$, где $d = u - 1$ и $A_d = A_c$. Если $w = \min(x, 1)$, тогда построить дочернюю «плохую» вершину с пометкой $[0, q', \emptyset]$ и для каждого $u \in A_c$ построить дочернюю «хорошую» вершину с пометкой $[d, q', A_d]$, где $d = u + 1$ и $A_d = A_c$.
   g) Если вершина $[c, q, A_c]$ не имеет потомков, то пометить ее как обработанный лист (поставить маркировку).

Построенное с помощью этого алгоритма дерево $\mathcal{T}(1cAM)$ в каждой вершине содержит подробную информацию о том, какие конфигурации могут быть порождены автоматной односчетчиковой машиной $1cAM$ из начальной конфигурации.

Лемма 1 (Лемма Кёнига [9]). *Пусть T — корневое дерево, в котором каждая вершина имеет конечное число преемников и не существует бесконечного пути, исходящего из корня. Тогда дерево T конечно.*

Лемма 2. *Алгоритм построения дерева $\mathcal{T}(1cAM)$ всегда останавливается. Дерево $\mathcal{T}(1cAM)$ конечно и эффективно вычислимо.*

Доказательство. Предположим, что алгоритм никогда не останавливается. Все операции внутри цикла имеют конечное время исполнения, поэтому остается лишь возможность того, что сам цикл никогда не завершится. С каждым полным проходом цикла происходит обработка и создание новых вершин дерева. Следовательно, в таком случае происходит построение бесконечного дерева. Поскольку дерево для каждой вершины имеет конечную степень ветвления, так как каждое множество A_c конечно, то по лемме Кёнига в дереве должен существовать бесконечный путь. Покажем, что все пути в дереве конечны, что в свою очередь будет означать обязательное завершение алгоритма.

Предположим (противное), что существует бесконечный путь по вершинам дерева $\mathcal{T}(1cAM)$: $\{[c_i, q_i, A_{c_i}] \mid i = 0, 1, \dots\}$. Сразу отметим, что этот бесконечный путь не может содержать бесконечное число «плохих» вершин, соответствующих плохим нулевым конфигурациям, так как мы имеем лишь конечное множество меток вида $[0, q, A_0]$ (из-за «обнуления» множества A_0 при порождении «плохой» вершины дерева и рассмотрении только хороших коротких путей из конфигурации $(q, 0)$ при добавлении новых элементов в A_0), где $q \in Q$; иначе на бесконечном пути обязательно встретятся две вершины с одинаковой меткой, что будет удовлетворять условию конечности пути. Следовательно, на бесконечном пути через некоторое время не будут встречаться «плохие» вершины.

Таким образом, с учетом этого факта и по построению дерева на бесконечном пути начиная с некоторого индекса $j \geq 0$ имеем бесконечную последовательность меток вершин $\{[c_i, q_i, A_{c_i}] \mid i \in \mathbb{N} \wedge j \leq i\}$, для которой выполняется $A_{c_i} \subseteq A_{c_{i+1}}$ для всех $i \geq j$. Покажем, что начиная с некоторого индекса k, $k \geq j$, в этой последовательности будет выполняться условие $A_{c_i} = A_{c_{i+1}}$ для всех $i \geq k$, т. е. множество A_{c_i} будет оставаться неизменным. Для этого воспользуемся идеей доказательства леммы 1.4 из работы [7]. Без потери общности рассуждений будем предполагать, что множество A_{c_k} содержит некоторый элемент a. Тогда пространство $\mathbb{N}$ разбивается на конечное число классов эквивалентности по модулю a. Поскольку элемент a должен находиться в каждом множестве A_{c_i}, $i \geq k$, то если b принадлежит $\mathcal{L}(0; A_{c_i})$, тогда все $b' \geq b$ того же класса эквивалентности также должны принадлежать множеству $\mathcal{L}(0; A_{c_i})$. Наряду с тем, что мы имеем конечное число классов эквивалентности, для заданного элемента b существует лишь конечное число элементов $b' < b$, принадлежащих к одному классу эквивалентности. Таким образом, существует только лишь конечное число индексов i, таких что множество $\mathcal{L}(0; A_{c_i})$ содержит в себе некоторый элемент d при том, что $d \notin \mathcal{L}(0; A_{c_l})$ для всех $l < i$.

Пусть p — индекс в бесконечной последовательности меток $[c_i, q_i, A_{c_i}]$, начиная с которого множество $A_{c_i} = A_{c_p}$ для всех $i \geq p$. Рассмотрим эту бесконечную последовательность с индекса p. Пусть $q \in Q$ — состояние, которое встречается бесконечно часто в этой последовательности. Тогда построим бесконечную подпоследовательность вида $\{[c_i, q, A_{c_i}] \mid i \in \mathbb{N} \wedge p \leq i\}$ при $A_{c_i} = A_{c_p}$ для всех $i \geq p$. В этой подпоследовательности обязательно найдутся две метки с индексами n и m, $n < m$, такие, что $c_n \leq c_m$. Поскольку путь из $[c_n, q, A_{c_n}]$ в $[c_m, q, A_{c_m}]$, где $A_{c_n} = A_{c_m} = A_{c_p}$, не содержит «плохих» вершин, то по построению дерева $\mathcal{T}(1cAM)$ разность $d = c_m - c_n$ должна содержаться в A_{c_p} или представлять собой линейную комбинацию элементов из множества A_{c_p}. Отсюда следует, что элемент $c_m = c_n + d$ будет принадлежать множеству $\mathcal{L}(c_n; A_{c_n})$, так как $d \in \mathcal{L}(0; A_{c_n})$. Но это удовлетворяет условию конечности пути, т. е. вершина $[c_m, q, A_{c_m}]$ должна быть помечена как обработанный лист (иметь маркировку). Поскольку множество состояний Q машины $1cAM$ конечно, исходя из всего приведенного выше, приходим к противоречию. Бесконечного пути существовать не может, все ветки дерева $\mathcal{T}(1cAM)$ конечны, а это означает, что алгоритм всегда останавливается. □

Пусть $\mathcal{T}_q = \cup \mathcal{L}(c; A_c)$, где объединение проводится по всем вершинам $[c, q, A_c]$ дерева $\mathcal{T}(1cAM)$ для заданного состояния q.

Лемма 3. *Множество $\mathcal{T}_q$ является полулинейным и эффективно вычислимым.*

Доказательство. Поскольку дерево $\mathcal{T}(1cAM)$ конечно, справедливость данного утверждения следует из предыдущей леммы. □

Следующая лемма показывает, что алгоритм строит в точности множество достижимости автоматной односчетчиковой машины $1cAM$.

Лемма 4. *Пусть R_q — множество значений счетчика машины $1cAM$, достижимых в состоянии q. Тогда $R_q = \mathcal{T}_q$ для всех состояний $q \in Q$.*

Доказательство. Докажем сначала, что $\mathcal{T}_q \subseteq R_q$. Покажем по индукции в глубину для дерева $\mathcal{T}(1cAM)$, что для каждого узла $[c, q, A_c]$ выполняется $\mathcal{L}(c; A_c) \subseteq R_q$.

Базис индукции. Пусть $[c_0, q_0, A_{c_0}]$ — метка корня дерева, а u — произвольный элемент из множества A_{c_0}. Тогда элемент u представляет собой либо разность допустимого для c_0 короткого положительного пути из состояния q_0 в q_0, либо значение, полученное как $u = (k_1 u_1 + \cdots + k_m u_m) + v$, где $v < 0$ — разность хорошего короткого *неположительного* пути из q_0 в q_0, $u_i \in A_{c_0}$, а $(k_1, \ldots, k_m) \in \mathbb{N}^m$ такой минимальный набор, что $u > 0$. В последнем случае после прохождения k_1 раз пути, который соответствует элементу u_1, и т. д. вплоть до прохождения k_m раз пути, соответствующего элементу u_m, может быть пройден путь, соответствующий v. Отсюда следует, что путь для элемента u ($u > 0$) также допустим из конфигурации (q_0, c_0). В обоих случаях значение счетчика $c_0 + u$ достижимо в состоянии q_0 и справедливо $c_0 + u > c_0$. Следовательно, значение $c_0 + v'$, где $v' \in \mathcal{L}(0; A_{c_0})$, также достижимо в состоянии q_0 из c_0 и $\mathcal{L}(c_0; A_{c_0}) \subseteq R_{q_0}$.

Предположение индукции. Предположим, что для каждой вершины $[c, q, A_c]$ глубины не более $n-1$ выполняется $\mathcal{L}(c; A_c) \subseteq R_q$.

Шаг индукции. Пусть $[d, q', A_d]$ — некоторая вершина глубины n, полученная из ее родительской вершины $[c, q, A_c]$ посредством перехода $q \to (q', w)$. По построению рассмотрим два случая.

Первый случай. Пусть $[d, q', A_d]$ — «хорошая» вершина. Возьмем произвольный элемент $z \in \mathcal{L}(d; A_d)$. Покажем, что вектор $z \in R_{q'}$, т. е. $z = d + v$, $v \in \mathcal{L}(0; A_d)$. Поскольку $A_c \subseteq A_d$, имеем $v = v_2 + v_3$, $v_2 \in \mathcal{L}(0; A_c)$, $v_3 \in \mathcal{L}(0; A_d - A_c)$. Значение $d = c + u + w$, где $u = 0$ при $c > 0$ и $u \in A_c$ при $c = 0$.

Рассмотрим первый вариант, при котором $c > 0$ и $u = 0$. Тогда $d = c + w$ и $z = c+w+v_2+v_3$, где $c+v_2 \in \mathcal{L}(c; A_c)$ — значение счетчика, достижимое в состоянии q по предположению индукции. Поскольку из (q, c) переход $q \to (q', w)$ порождает хорошую конфигурацию (q', d), то и из большей конфигурации $(q, c+v_2)$ этот переход порождает хорошую конфигурацию. Отсюда имеем $c + v_2 + w = d + v_2 \in R_{q'}$.

Рассмотрим второй вариант, при котором $c = 0$ и $u \in A_c$. Тогда $d = u + w$ и $z = u+w+v_2+v_3$, где $u+v_2 \in \mathcal{L}(c; A_c) = \mathcal{L}(0; A_c)$ — значение счетчика, достижимое в состоянии q по предположению индукции. Из конфигурации $(q, u+v_2)$ переход $q \to (q', w)$ порождает хорошую конфигурацию $(q', u+v_2+w)$. Отсюда имеем $u+v_2+w = d+v_2 \in R_{q'}$. Этот вариант соответствует тому, что прежде, чем произойдет переход $q \to (q', w)$, может быть пройден хороший положительный путь из конфигурации $(q, 0)$ в конфигурацию (q, u), соответствующий некоторому периоду $u \in A_c$.

Итак, $d + v_2 \in R_{q'}$ при любом из рассмотренных вариантов значения d, где $v_2 \in \mathcal{L}(0; A_c)$. Покажем, что $d + v_2 + v_3 = z \in R_{q'}$. По аргументам, аналогичным тем, которые были приведены в базисе индукции, если каждый элемент множества $A_d - A_c$ представляет собой для конфигурации (q', d) либо разность хорошего короткого положительного пути, либо разность пути, являющегося комбинацией хороших коротких положительных и хороших коротких неположительных путей, то соответствующие пути возможны и из большей конфигурации $(q', d + v_2)$. Более того, элемент множества $A_d - A_c$ может быть разностью $d - d' > 0$, где d' — значение счетчика в вершине $[d', q', A_{d'}]$, являющейся предком вершины $[d, q', A_d]$, между которыми не лежит «плохих» вершин. Это означает, что из конфигурации (q', d') есть хороший положительный путь, приводящий в конфигурацию (q', d), который возможен и из большей конфигурации $(q', d+v_2)$. Таким образом, $z = d+v_2+v_3 \in R_{q'}$.

Второй случай. Пусть $[d, q', A_d]$ — «плохая» вершина. Тогда $d = 0$ и $c = 0$, а $w \in \{\ominus 1, \min(x, 1)\}$. Переход $q \to (q', w)$ порождает из $(q, 0)$ конфигурацию $(q', 0)$. По построению «плохой» вершины все элементы множества A_d представляют собой либо разность хорошего короткого положительного пути, либо разность пути, являющегося комбинацией хороших коротких положительных и хороших коротких неположительных путей. По аргументам, аналогичным тем, которые были приведены в базисе индукции $\mathcal{L}(d; A_d) = \mathcal{L}(0; A_d) \subseteq R_{q'}$.

Теперь докажем, что $R_q \subseteq \mathcal{T}_q$ для всех $q \in Q$. Покажем методом индукции по длине пути машины $1cAM$ из начальной конфигурации (q_0, c_0) в конфигурацию (q, d), что $d \in \mathcal{T}_q$.

Базис индукции. Если $d = c_0$, тогда $d \in \mathcal{L}(c_0; A_{c_0}) \subseteq \mathcal{T}_{q_0}$.

Предположение индукции. Предположим, что для каждого состояния q и каждого значения счетчика d, достижимого в состоянии q посредством пути длины не более $n - 1$, выполняется $d \in \mathcal{L}(c; A_c)$ для некоторой вершины $[c, q, A_c]$.

Шаг индукции. Пусть d — значение счетчика, достижимое в состоянии q через путь длины n, и пусть $q' \to (q, w)$ — последний переход этого пути, переводящий значение k в d, т. е. $d = k + w$. По предположению индукции $k \in \mathcal{L}(c'; A_{c'})$ для некоторой вершины $[c', q', A_{c'}]$. Тогда $k = c' + y$, где $y \in \mathcal{L}(0; A_{c'})$. Будем предполагать, что вершина $[c', q', A_{c'}]$ не является листом дерева, так как в противном случае имеем следующие варианты.

1. Нет ни одного перехода, применимого к конфигурации (q', c'). Противоречие.
2. $\mathcal{L}(c'; A_{c'}) \subseteq \mathcal{L}(c''; A_{c''})$, где $A_{c'} = A_{c''}$, для некоторой внутренней (предшествующей) вершины $[c'', q', A_{c''}]$. Тогда можно заменить c' значением c'' и вместо $[c', q', A_{c'}]$ рассматривать вершину $[c'', q', A_{c''}]$.

Итак, пусть $A_{c'} = \{u_1, \ldots, u_m\}$. Если (q, d) — плохая конфигурация, то $d = 0$, а $[c', q', A_{c'}] = [0, q', A_{c'}]$ имеет дочернюю вершину, помеченную $[d, q, A_d] = [0, q, A_d]$. Имеем $d = 0 \in \mathcal{L}(0; A_d) \subseteq \mathcal{T}_q$. Пусть (q, d) — хорошая конфигурация. Тогда $[c', q', A_{c'}]$ имеет дочернюю вершину, помеченную базисным вектором $c = c' + w$ при $c' > 0$ или $c = u + w$ при $c' = 0$, где $u \in A_{c'}$. Рассмотрим первый вариант, при котором базисный вектор имеет вид $c = c' + w$ для $c' > 0$. В этом случае $d = k + w = c' + y + w = c + y$, где $y \in \mathcal{L}(0; A_{c'})$. Поскольку по построению $\mathcal{L}(0; A_{c'}) \subseteq \mathcal{L}(0; A_c)$, то $y \in \mathcal{L}(0; A_c)$. Следовательно, $d \in \mathcal{L}(c; A_c) \subseteq \mathcal{T}_q$. Теперь рассмотрим второй вариант, при котором базисный вектор имеет вид $c = u + w$ для $c' = 0$, где $u \in A_{c'}$. В этом случае $d = k + w = c' + y + w = y + w$, где $y \in \mathcal{L}(0; A_{c'})$. Существует период $u \in A_{c'}$ такой, что $y = u + v$, где $v \in \mathcal{L}(0; A_{c'}$. Получаем, что $d = u + v + w = c + v$. Из $\mathcal{L}(0; A_{c'}) \subseteq \mathcal{L}(0; A_c)$ имеем $d \in \mathcal{L}(c; A_c) \subseteq \mathcal{T}_q$.

Получили, что одновременно выполняется $R_q \subseteq \mathcal{T}_q$ и $\mathcal{T}_q \subseteq R_q$ для всех $q \in Q$, из чего следует справедливость $R_q = \mathcal{T}_q$ для всех $q \in Q$. Что и требовалось доказать.

□

Теорема 1. *Для любой автоматной односчетчиковой машины множество достижимых значений счетчика является полулинейным и эффективно вычисляется.*

Справедливость этой основной теоремы непосредственно вытекает из предыдущих двух лемм.

Следствие 1. *Проблема достижимости, проблемы R-включения и R-эквивалентности являются разрешимыми для автоматных односчетчиковых машин.*

Доказательство. Поскольку для любой автоматной односчетчиковой машины множество достижимости является полулинейным, то все рассматриваемые проблемы могут быть представлены в виде формул арифметики Пресбургера. Тогда решение этих проблем связывается с разрешимостью соответствующих формул. Известно, что арифметика Пресбургера разрешима [16]. □

4. Автоматные машины с ограничениями

Как уже было отмечено во введении, для счетчиковых машин, кроме таких ослаблений, как недетерминизм переходов, невозможность совершать проверку на нуль, ограничения на число счетчиков и т. п., обычно вводятся еще ограничения на число перемен направлений роста/убывания значений счетчиков и структуру графа переходов. В этом разделе мы коротко обсудим классы автоматных счетчиковых машин, имеющих последние два из указанных ослаблений.

Итак, *автоматная n-счетчиковая машина с ограничением на количество перемен направлений* представляет собой автоматную машину с n счетчиками, любое исполнение которой допускает лишь конечное число перемен направлений роста/убывания значений каждого из счетчиков. Для автоматных счетчиковых машин этого класса справедлива следующая теорема (из которой вытекает разрешимость всех основных свойств, таких как ограниченность, достижимость, эквивалентность, включения и т. д.).

Теорема 2. *Множество достижимости любой автоматной счетчиковой машины с ограничением на количество перемен направлений является полулинейным.*

Истинность этой теоремы следует из аналогичной теоремы для *расширенных счетчиковых машин с ограничением на количество перемен направлений* из работы [6] и того очевидного факта, что класс расширенных счетчиковых машин включает в себя класс автоматных счетчиковых машин (см. [6]).

Автоматную счетчиковую машину назовем *плоской*, если ее (ориентированный) граф «управления», образованный состояниями и переходами машины, не содержит вложенных циклов.

Теорема 3. *Множество достижимости любой плоской автоматной счетчиковой машины является полулинейным.*

Справедливость теоремы фактически следует из работ [5, 13] о плоских счетчиковых системах.

Отметим, что существование алгоритма, определяющего, является ли произвольная автоматная счетчиковая машина ограниченной по числу перемен направлений или трансформируемой в эквивалентную по множеству достижимости плоскую автоматную счетчиковую машину, — открытая проблема. Другими словами, не решен вопрос о существовании алгоритма, решающего, имеет ли произвольная автоматная счетчиковая машина полулинейное множество достижимости.

Список литературы

1. *Annichini A.*, *Bouajjani A.*, *Sighireanu M.* TReX: A tool for reachability analysis of complex systems // LNCS 2102, Springer, 2001. P. 368–372.

2. *Araki T.*, *Kasami T.* Decidable problems on the strong connectivity of Petri net reachability sets // Theoretical Computer Science, 4(1), 1977. P. 99–119.

3. *Bardin S.*, *Leroux J.*, *Point G.* FAST extended release // LNCS 4144, Springer, 2006. P. 63–66.

4. *Esparza J.* Petri nets, commutative context-free grammars, and basic parallel processes // Fundamenta Informatica, 31(1), 1997. P. 13—25.

5. *Finkel A.*, *Leroux J.* How to compose Presburger-accelerations: Applications to broadcast protocols // LNCS 2556, Springer, 2002. P. 145–156.

6. *Finkel A.*, *Sangnier A.* Reversal-bounded Counter Machines Revisited // LNCS 5162, Springer, 2008. P. 323–334.

7. *Hopcroft J. E.*, *Pansiot J.* On the Reachability Problem for 5-Dimensional Vector Addition Systems. Computer science technical report. Cornell University, 1976. — http://hdl.handle.net/1813/6102

8. *Ibarra O. H.* Reversal-bounded multicounter machines and their decision problems // J. ACM, 25(1), 1978. P. 116—133.

9. *König D.* Theorie der Endlichen und Unendlichen Graphen // Akademische Verlagsgesellschaft, Leipzig, 1936.

10. *Kouzmin E. V.*, *Sokolov V. A.* Communicating Colouring Automata // Proc. Int. Workshop on Program Understanding (sat. of PSI'03), 2003. P. 40–46.

11. *Kuzmin E. V.*, *Sokolov V. A.*, *Chaly D. Ju.* Automaton Counter Machines // Proc. of Int. Workshop on Program Understanding (sat. of PSI'09), 2009. P. 1–4.

12. LASH. http://www.montefiore.ulg.ac.be/~boigelot/research/lash

13. *Leroux J.*, *Sutre G.* Flat counter almost everywhere! // LNCS 3707, Springer, 2005. P. 474–488.

14. *Mayr E. W.* Persistence of vector replacement systems is decidable // Acta Informatica, 15, 1981. P. 309–318.

15. *Mayr E. W.* An algorithm for the general Petri net reachability problem // SIAM J. Comput., 13(3), 1984. P. 441–460.

16. *Presburger M.* Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt / M. Presburger // Sprawozdanie z I Kongresu matematyków krajów slowiańskich, Warszawa 1929, Warsaw, 1930. P. 92–101, 395.

17. *Valk R.*, *Vidal-Naquet G.* Petri nets and regular languages // J. Comput. Syst. Sci., 23(3), 1981. P. 299–325.

18. *Гинзбург С.* Математическая теория контекстно-свободных языков. М.:Мир, 1970. 328 с.

19. *Кузьмин Е. В.*, *Соколов В. А.* Взаимодействующие раскрашивающие процессы // Моделирование и анализ информационных систем. 2004. Т. 11, №2. С. 8–17.

20. *Кузьмин Е. В.*, *Чалый Д. Ю.* Об одном классе счетчиковых машин // Моделирование и анализ информационных систем. 2009. Т. 16, №2. С. 75–82.

21. *Кузьмин Е. В., Чалый Д. Ю.* О множестве достижимости автоматных трехсчетчиковых машин // Моделирование и анализ информационных систем. 2009. Т. 16, №3. С. 77–84.

22. *Минский М.* Вычисления и автоматы. М.: Мир, 1971. 268 с.

On a reachability set of automaton counter machines

Kuzmin E. V., Chalyy D. J.

Keywords: abstract counter machines, automaton counter machine, Communicating Colouring Automata, reachability sets, semilinear sets

Properties of automaton counter machines are investigated. We prove that reachability sets of automaton one-counter machines are semilinear. An algorithm of construction of these semilinear reachability sets is resulted. Besides, it is shown that reachability sets of reversal-bounded automaton counter machines and reachability sets of flat automaton counter machines are also semilinear.

Сведения об авторах:
Кузьмин Егор Владимирович,
Ярославский государственный университет им. П.Г. Демидова,
канд. физ.-мат. наук, доцент;
Чалый Дмитрий Юрьевич,
Ярославский государственный университет им. П.Г. Демидова,
канд. физ.-мат. наук, доцент