Модел. и анализ информ. систем. Т.**15**, №**1** (2008) 16–26

УДК 519.68/.69

О разрешимости проблем ограниченности для счетчиковых машин Минского [1]

Кузьмин Е. В., Чалый Д. Ю.
Ярославский государственный университет
150 000, Ярославль, Советская, 14

получена 12 февраля 2008

Аннотация

Исследуется разрешимость проблем ограниченности для счетчиковых машин Минского. Доказывается, что для машин Минского с двумя счетчиками проблема ограниченности лишь частично разрешима, а проблема тотальной ограниченности не является даже частично разрешимой. Для односчетчиковых машин Минского указанные проблемы разрешимы за время, полиномиально зависящее от общего количества локальных состояний счетчиковой машины.

1. Введение

В статье исследуется разрешимость проблем ограниченности и тотальной ограниченности для счетчиковых машин Минского [5]. Рассмотрение счетчиковых машин здесь проводится с точки зрения моделирования и анализа программных систем. Давно уже стал стандартным прием использования абстрактных счетчиковых машин в качестве общего средства для демонстрации алгоритмической неразрешимости ряда проблем для формальных моделей программных и аппаратных систем, способных моделировать поведение этих машин. В частности все неразрешимые проблемы для машин Минского автоматически переходят и на сети Петри с ингибиторными дугами и сети Петри с приоритетами [2], неразрешимость проблем счетчиковых машин с потерями [11] ведет к неразрешимости тех же проблем для сетей Петри с обнуляющими дугами, FIFO-канальных систем с потерями и т. д. Исследуемая в статье проблема ограниченности является одной из самых важных для теории формальных моделей, поскольку в случае разрешимости позволяет определять, является ли множество состояний модели программной системы конечным, что критично, например, для автоматического метода верификации model checking [1]. К сожалению, как будет показано далее, для двухсчетчиковых машин Минского проблема ограниченности лишь частично разрешима, а проблема тотальной ограниченности не является даже частично разрешимой. Интересно, что в литературе проблеме ограниченности для каждого отдельного формализма уделяется довольно много внимания. Однако ограниченность непосредственно для счетчиковых машин Минского практически не исследовалась. Отметим лишь, что в работе [9] факт ограниченности двухсчетчиковых машин как очевидное следствие того, что машина Минского с двумя счетчиками может моделировать машину Тьюринга [5], был бездоказательно использован для анализа сетей Петри с обнуляющими дугами.

Особого внимания заслуживают односчетчиковые машины Минского. Несмотря на наличие всего лишь одного счетчика, абстрактные машины этого класса имеют широкий спектр применения, например, для верификации криптографических протоколов [10], валидации XML потоков [7] и решения задачи идентификации [12]. Кроме того, односчетчиковые машины могут быть промоделированы специальным подклассом магазинных автоматов, алфавит которых состоит только из одного символа [6]. В данной статье показывается разрешимость за полиномиальное время проблем ограниченности и тотальной ограниченности. Более того, из леммы о конечном пути, задающем полное исполнение (возможно бесконечное) односчетчиковой машины, из работы [8], можно также говорить о полиномиальной разрешимости и задач достижимости, останова, тотальности, пустоты и т. п.

[1]Работа поддержана РФФИ, грант №07-01-00702-а.

2. Счетчиковые машины Минского

Счетчиковая машина Минского 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 перехода для q_i $(0 \le i \le n-1)$.

Состояния q_i, $0 \le i \le n-1$, подразделяются на два типа. Состояния первого типа имеют правила переходов вида:

$$\delta_i\colon\ x_j := x_j + 1;\ \mathsf{goto}\ q_k,$$

где $1 \le j \le m$, $0 \le k, l \le n$. Для состояний второго типа имеем:

$$\delta_i\colon\ \mathsf{if}\ x_j > 0\ \mathsf{then}\ (x_j := x_j - 1;\ \mathsf{goto}\ q_k)\ \mathsf{else}\ \mathsf{goto}\ q_l.$$

Конфигурация машины Минского представляет собой набор $(q_i, c_1, \ldots, c_m)$, где q_i — состояние машины, $c_1, \ldots, c_m$ — натуральные числа (включая ноль), являющиеся значениями соответствующих счетчиков. Размер конфигурации определяется как $\mathrm{size}(q, c_1, \ldots, c_m) = \sum_{i=1}^{m} c_i$.

Исполнение счетчиковой машины — это последовательность конфигураций $s_0 \to s_1 \to s_2 \to \ldots$ с начальной конфигурацией s_0, индуктивно определяемая в соответствии с правилами переходов. Необходимо отметить, что счетчиковая машина имеет всего лишь одно (детерминированное) исполнение из начальной конфигурации s_0, 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. Поскольку 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 Минского является неразрешимой [5]. Однако существует простой частичный алгоритм решения этой проблемы. Достаточно всего лишь запустить счетчиковую машину из начальной конфигурации и, если финальное состояние достижимо, просто дождаться, когда она сгенерирует конфигурацию с финальным состоянием. Двойственная же проблема, проблема зацикливания, не является даже частично разрешимой, т.е. не существует даже частичного алгоритма, который устанавливает для начальной конфигурации (q_0, c_1, c_2), будет ли машина Минского с двумя счетчиками работать бесконечно долго [6]. Более того, все указанное выше остается справедливым и при фиксированной нулевой начальной конфигурации $(q_0, 0, 0)$.

Говорят, что машина Минского *ограничена* при некоторой начальной конфигурации тогда и только тогда, когда существует натуральное число c' такое, что на протяжении всего исполнения машины из начальной конфигурации выполняется условие $x_1 + \ldots + x_m \le c'$, т.е. в любой момент времени сумма значений счетчиков не превосходит c'.

Машина Минского M *тотально ограничена*, если и только если она ограничена для всех возможных начальных конфигураций.

Теорема 1. *Для трехсчетчиковых машин Минского проблема ограниченности неразрешима.*

Доказательство осуществляется сведением проблемы останова для машины Минского с двумя счетчиками к проблеме ограниченности машины Минского с тремя счетчиками. Рассмотрим некоторую машину Минского $2cM = (\{q_0, \ldots, q_n\}, \{x_1, x_2\}, \{\delta_0, \ldots, \delta_{n-1}\})$. Преобразуем $2cM$ в машину Минского $3cM$ с тремя счетчиками. Добавим новый счетчик x_3. И для каждого состояния q_i, $0 \le i \le n-1$, добавим состояния q'_i, q''_i, правила переходов δ'_i, δ''_i и изменим δ_i следующим образом.

Для состояния первого типа имеем:

$(\delta_i)\ q_i\colon\ x_j := x_j + 1;\ \mathsf{goto}\ q'_i;\quad (\delta'_i)\ q'_i\colon\ x_3 := x_3 + 1;\ \mathsf{goto}\ q_k.$

Для состояния второго типа:

$(\delta_i)\ q_i\colon\ \mathsf{if}\ x_j > 0\ \mathsf{then}\ (x_j := x_j - 1;\ \mathsf{goto}\ q'_i)\ \mathsf{else}\ \mathsf{goto}\ q''_i;$

$(\delta'_i)\ q'_i\colon\ x_3 := x_3 + 1;\ \mathsf{goto}\ q_k;\quad (\delta''_i)\ q''_i\colon\ x_3 := x_3 + 1;\ \mathsf{goto}\ q_l.$

Машина Минского $2cM$ имеет конечное исполнение тогда и только тогда, когда машина Минского $3cM$ ограничена, так как счетчик x_3, использующийся для подсчета количества шагов исполнения машины $2cM$, ограничен только в случае конечного исполнения. □

Теорема 2. *Для любой трехсчетчиковой машины Минского $3cM$ может быть построена эквивалентная (имитирующая ее работу) машина Минского с двумя счетчиками $2cM$ такая, что конфигурация со значениями счетчиков $y_1 = 2^a \cdot 3^b \cdot 5^c$ и $y_2 = 0$ машины $2cM$ будет существовать тогда и только тогда, когда существует конфигурация с $x_1 = a$, $x_2 = b$ и $x_3 = c$ машины $3cM$.*

Доказательство. Чтобы свести машину Минского $3cM$ со счетчиками x_1, x_2 и x_3 к машине Минского $2cM$ с двумя счетчиками y_1 и y_2, используют элементарный факт из арифметики, что каждое целое число может быть разложено на простые множители единственным способом. Итак, установим начальными значениями счетчиков y_1 и y_2 числа $2^{x_1} \cdot 3^{x_2} \cdot 5^{x_3}$ и 0 соответственно. Важно то, что из одного числа $2^{x_1} \cdot 3^{x_2} \cdot 5^{x_3}$ можно найти числа x_1, x_2 и x_3 просто путем определения, сколько раз исходное число может быть разделено на 2, на 3 и на 5 соответственно. Для достижения поставленной задачи нужно лишь показать, как можно получить результат операций увеличения на единицу и условного вычитания единицы до нуля для счетчиков x_1, x_2 и x_3 без нарушения требований в формулировке теоремы.

Начнем с приращения. Пусть необходимо увеличить значение счетчика x_1 на единицу, т. е. реализовать операцию $x_1 := x_1 + 1$. Это означает, что мы хотим заменить старое значение $2^{x_1} \cdot 3^{x_2} \cdot 5^{x_3}$ счетчика y_1 на новое $2^{x_1+1} \cdot 3^{x_2} \cdot 5^{x_3} = 2 \cdot 2^{x_1} \cdot 3^{x_2} \cdot 5^{x_3}$. Но это то же самое, что удвоение значения счетчика y_1. Подобно этому увеличение на 1 значения счетчика x_2 или x_3 есть соответственно утроение или упятерение содержимого счетчика y_1. Ниже представлена программа, с помощью которой реализуется умножение на 2. Программа умножения на 3 и на 5 строится аналогичным образом. Первый цикл в каждой из программ осуществляет подсчет нового содержимого счетчика y_1 путем прибавления двух (или трех, или пяти) единиц к значению счетчика y_2 столько раз, сколько единиц в счетчике y_1, второй цикл осуществляет передачу содержимого счетчика y_2 в y_1. Важным моментом в программе является то, что она начинается с увеличения нулевого значения счетчика y_2 на 1 и заканчивается уменьшением единичного значения y_2 до 0. Таким образом, все вспомогательные переходы (кроме последнего), переводящие конфигурацию $(q_i,\ 2^{x_1} \cdot 3^{x_2} \cdot 5^{x_3},\ 0)$ в конфигурацию $(q_k,\ 2^{x_1+1} \cdot 3^{x_2} \cdot 5^{x_3},\ 0)$, проходят по промежуточным конфигурациям со значением счетчика $y_2 > 0$.

Итак, для перехода

$(\delta_i)\ q_i\colon\ x_1 := x_1 + 1;\ \textsf{goto}\ q_k;$

имеем следующий набор правил переходов:

$(\delta_i)\ q_i\colon\ y_2 := y_2 + 1;\ \textsf{goto}\ q_i^1;$

$(\delta_i^1)\ q_i^1\colon\ y_2 := y_2 + 1;\ \textsf{goto}\ q_i^2;$

$(\delta_i^2)\ q_i^2\colon\ \textsf{if}\ y_1 > 0\ \textsf{then}\ (y_1 := y_1 - 1;\ \textsf{goto}\ q_i^3);$

$(\delta_i^3)\ q_i^3\colon\ \textsf{if}\ y_1 > 0\ \textsf{then}\ (y_1 := y_1 - 1;\ \textsf{goto}\ q_i^4)\ \textsf{else goto}\ q_i^5;$

$(\delta_i^4)\ q_i^4\colon\ y_1 := y_1 + 1;\ \textsf{goto}\ q_i;$

$(\delta_i^5)\ q_i^5\colon\ y_1 := y_1 + 1;\ \textsf{goto}\ q_i^6;$

$(\delta_i^6)\ q_i^6\colon\ y_1 := y_1 + 1;\ \textsf{goto}\ q_i^7;$

$(\delta_i^7)\ q_i^7\colon\ \textsf{if}\ y_2 > 0\ \textsf{then}\ (y_2 := y_2 - 1;\ \textsf{goto}\ q_i^8);$

$(\delta_i^8)\ q_i^8\colon\ \textsf{if}\ y_2 > 0\ \textsf{then}\ (y_2 := y_2 - 1;\ \textsf{goto}\ q_i^9);$

$(\delta_i^9)\ q_i^9\colon\ \textsf{if}\ y_2 > 0\ \textsf{then}\ (y_2 := y_2 - 1;\ \textsf{goto}\ q_i^{10})\ \textsf{else goto}\ q_k;$

$(\delta_i^{10})\ q_i^{10}\colon\ y_2 := y_2 + 1;\ \textsf{goto}\ q_i^5.$

Рассмотрим теперь условное вычитание единицы до нуля. Уменьшение значения счетчика x_1 на единицу оказывается несколько более сложной задачей, так как необходимо сначала определить, не является ли x_1 нулем; если значение x_1 равно нулю, то нам нужно оставить его неизменным и осуществить соответствующий условный переход. Если x_1 не равен нулю, то нужно заменить старое значение $2^{x_1} \cdot 3^{x_2} \cdot 5^{x_3}$ счетчика y_1 на новое $2^{x_1-1} \cdot 3^{x_2} \cdot 5^{x_3}$, т. е. разделить содержимое y_1 на 2. Аналогично уменьшение x_2 и x_3 (если они не равны нулю) эквивалентно делению на 3 и на 5. Программа для уменьшения значения счетчика x_3 на 1 представлена ниже. Деление осуществляется путем последовательного вычитания. Первый цикл переносит исходное значение счетчика y_1 в счетчик y_2, одновременно определяя, делится ли значение y_1 нацело. Если деление происходит нацело, т. е. без остатка, то с помощью другого цикла производится занесение частного от деления y_2 в счетчик y_1. Если деление происходит не нацело (т. е. x_3 равен нулю), третий цикл восстанавливает остаток (который в этот момент запоминается в виде состояния машины, т. е. в виде места в программе), направляя его в y_1, а затем прибавляет к нему частное (находящееся в y_2), умноженное на делитель. Как и в случае приращения, все промежуточные переходы (кроме последнего) из $(q_i,\ 2^{x_1} \cdot 3^{x_2} \cdot 5^{x_3},\ 0)$ в конфигурацию $(q_k,\ 2^{x_1} \cdot 3^{x_2} \cdot 5^{x_3-1},\ 0)$ или из $(q_i,\ 2^{x_1} \cdot 3^{x_2} \cdot 5^{0},\ 0)$ в конфигурацию $(q_l,\ 2^{x_1} \cdot 3^{x_2} \cdot 5^{0},\ 0)$ осуществляются при значении счетчика $y_2 > 0$.

Итак, для перехода

$(\delta_i)\ q_i\colon\ \textsf{if}\ x_3 > 0\ \textsf{then}\ (x_3 := x_3 - 1;\ \textsf{goto}\ q_k)\ \textsf{else goto}\ q_l;$

имеем следующий набор правил переходов:

$(\delta_i)\ q_i\colon\ y_2 := y_2 + 1;\ \textsf{goto}\ q_i^1;$

(δ_i^1) $q_i^1:$ if $y_1 > 0$ then ($y_1 := y_1 - 1;$ goto q_i^2);
(δ_i^2) $q_i^2:$ if $y_1 > 0$ then ($y_1 := y_1 - 1;$ goto q_i^3) else goto q_i^{20};
(δ_i^3) $q_i^3:$ $y_2 := y_2 + 1;$ goto q_i^4;
(δ_i^4) $q_i^4:$ if $y_1 > 0$ then ($y_1 := y_1 - 1;$ goto q_i^5) else goto q_i^{18};
(δ_i^5) $q_i^5:$ $y_2 := y_2 + 1;$ goto q_i^6;
(δ_i^6) $q_i^6:$ if $y_1 > 0$ then ($y_1 := y_1 - 1;$ goto q_i^7) else goto q_i^{16};
(δ_i^7) $q_i^7:$ $y_2 := y_2 + 1;$ goto q_i^8;
(δ_i^8) $q_i^8:$ if $y_1 > 0$ then ($y_1 := y_1 - 1;$ goto q_i^9) else goto q_i^{14};
(δ_i^9) $q_i^9:$ $y_2 := y_2 + 1;$ goto q_i^{10};
(δ_i^{10}) $q_i^{10}:$ if $y_1 > 0$ then ($y_1 := y_1 - 1;$ goto q_i^{11}) else goto q_i^{24};
(δ_i^{11}) $q_i^{11}:$ $y_1 := y_1 + 1;$ goto q_i;
(δ_i^{12}) $q_i^{12}:$ $y_1 := y_1 + 1;$ goto q_i^{13};
(δ_i^{13}) $q_i^{13}:$ if $y_2 > 0$ then ($y_2 := y_2 - 1;$ goto q_i^{14});
(δ_i^{14}) $q_i^{14}:$ $y_1 := y_1 + 1;$ goto q_i^{15};
(δ_i^{15}) $q_i^{15}:$ if $y_2 > 0$ then ($y_2 := y_2 - 1;$ goto q_i^{16});
(δ_i^{16}) $q_i^{16}:$ $y_1 := y_1 + 1;$ goto q_i^{17};
(δ_i^{17}) $q_i^{17}:$ if $y_2 > 0$ then ($y_2 := y_2 - 1;$ goto q_i^{18});
(δ_i^{18}) $q_i^{18}:$ $y_1 := y_1 + 1;$ goto q_i^{19};
(δ_i^{19}) $q_i^{19}:$ if $y_2 > 0$ then ($y_2 := y_2 - 1;$ goto q_i^{20});
(δ_i^{20}) $q_i^{20}:$ $y_1 := y_1 + 1;$ goto q_i^{21};
(δ_i^{21}) $q_i^{21}:$ if $y_2 > 0$ then ($y_2 := y_2 - 1;$ goto q_i^{22});
(δ_i^{22}) $q_i^{22}:$ if $y_2 > 0$ then ($y_2 := y_2 - 1;$ goto q_i^{23}) else (goto q_l /* если $x_3 = 0$ */);
(δ_i^{23}) $q_i^{23}:$ $y_2 := y_2 + 1;$ goto q_i^{12};
(δ_i^{24}) $q_i^{24}:$ $y_1 := y_1 + 1;$ goto q_i^{25};
(δ_i^{25}) $q_i^{25}:$ if $y_2 > 0$ then ($y_2 := y_2 - 1;$ goto q_i^{26});
(δ_i^{26}) $q_i^{26}:$ if $y_2 > 0$ then ($y_2 := y_2 - 1;$ goto q_i^{27});
(δ_i^{27}) $q_i^{27}:$ if $y_2 > 0$ then ($y_2 := y_2 - 1;$ goto q_i^{28});
(δ_i^{28}) $q_i^{28}:$ if $y_2 > 0$ then ($y_2 := y_2 - 1;$ goto q_i^{29});
(δ_i^{29}) $q_i^{29}:$ if $y_2 > 0$ then ($y_2 := y_2 - 1;$ goto q_i^{30});
(δ_i^{30}) $q_i^{30}:$ if $y_2 > 0$ then ($y_2 := y_2 - 1;$ goto q_i^{31}) else (goto q_k /* если $x_3 > 0$ */);
(δ_i^{31}) $q_i^{31}:$ $y_2 := y_2 + 1;$ goto q_i^{24}.

Чтобы построить счетчиковую машину $2cM$, эквивалентную машине $3cM$, такую что конфигурация со значениями счетчиков $y_1 = 2^a \cdot 3^b \cdot 5^c$ и $y_2 = 0$ машины $2cM$ будет существовать тогда и только тогда, когда существует конфигурация с $x_1 = a$, $x_2 = b$ и $x_3 = c$ машины $3cM$, надо заменить все правила переходов машины $3cM$ на соответствующие наборы правил переходов так, как это показано выше. □

Замечания по теореме 2. Основная идея доказательства теоремы была взята из доказательства оригинальной теоремы М. Минского о моделировании трехсчетчиковой машины машиной с двумя счетчиками [5]. Однако в доказательстве оригинальной теоремы для двух машин $2cM$ и $3cM$ не обеспечивалась взаимная однозначность конфигураций со значениями счётчиков $y_1 = 2^a \cdot 3^b \cdot 5^c$, $y_2 = 0$ и $x_1 = a$, $x_2 = b$, $x_3 = c$ соответственно. Потребовалось изменить конструкции моделирования таким образом, чтобы все вспомогательные переходы порождали конфигурации со значением $y_2 > 0$. Как только значение счетчика y_2 становится нулевым, это означает, что y_1 содержит новое значение $2^{a'} \cdot 3^{b'} \cdot 5^{c'}$, соответствующее $x_1 = a'$, $x_2 = b'$, $x_3 = c'$. Вычисление же этого нового значения $2^{a'} \cdot 3^{b'} \cdot 5^{c'}$ начинается с увеличения нулевого значения y_2 на единицу. Таким образом, конструкции, приведенные в доказательстве теоремы 2, обеспечивают для машин $2cM$ и $3cM$ взаимную однозначность конфигураций указанного вида, что позволяет использовать это утверждение для доказательства неразрешимости проблем, связанных также и с достижимостью конфигураций. □

Теорема 3. *Для двухсчетчиковых машин Минского проблема ограниченности неразрешима.*

Доказательство проводится методом сведения проблемы ограниченности трехсчетчиковых машин Минского к данной проблеме с помощью преобразований из доказательства теоремы 2.

Итак, возьмем произвольную трехсчетчиковую машину Минского $3cM$ и преобразуем ее в машину $2cM$ так, как это показано в доказательстве теоремы 2. Исходя из этих преобразований получаем, что счетчиковая машина $2cM$ будет ограничена тогда и только тогда, когда ограничена машина $3cM$. Если предположить, что существует алгоритм решения проблемы ограниченности для двухсчетчиковых машин Минского, тогда мы будем иметь алгоритм решения аналогичной проблемы для произвольной машины Минского с тремя счетчиками. А эта проблема неразрешима по теореме 1. Пришли к противоречию. □

Заметим, что из доказательства теоремы 2 также следует неразрешимость проблемы ограниченности одного счетчика двухсчетчиковой машины Минского, так как счетчик y_1 машины $2cM$, содержащий значение $2^{x_1} \cdot 3^{x_2} \cdot 5^{x_3}$, где x_1, x_2 и x_3 — счетчики машины $3cM$, будет ограничен тогда и только тогда, когда ограничена машина $3cM$. Таким образом, справедлива следующая теорема.

Теорема 4. *Проблема ограниченности одного счетчика для двухсчетчиковых машин Минского не является разрешимой.*

Замечания по теореме 3. Очевидно, что проблема ограниченности двухсчетчиковых машин Минского является частично разрешимой. Если машина $2cM$ ограничена, то она будет иметь конечное множество всех возможных конфигураций. Таким образом, бесконечное исполнение машины (если оно существует) обязательно будет проходить одни и те же конфигурации. Следовательно, если после запуска машины $2cM$ она попадет в конфигурацию, в которой уже находилась ранее (на исполнении), это и будет означать ограниченность машины, так как, начиная с данной конфигурации, будет порождаться бесконечный цикл, целиком состоящий из уже пройденных конфигураций, т. е. далее в исполнении новые конфигурации порождаться не будут. □

Замечания по теореме 4. Проблема ограниченности одного счетчика машины Минского $2cM$ также является частично разрешимой. Если один из счетчиков ограничен, то, очевидно, имеем конечно число всех возможных значений этого счетчика. Поскольку множество состояний машины также конечно, можно рассмотреть такую машину (с ограниченным счетчиком) $2cM$ как односчетчиковую машину Минского $1cM$, состояния которой представляют собой пары (q, c), где q — состояние исходной машины $2cM$, а c — значение ограниченного счетчика. В отличие от обычной машины Минского машина $1cM$ будет иметь и пустые переходы по состояниям нового вида (ранее соответствующие переходам для ограниченного счетчика). Таким образом, для частичного решения проблемы ограниченности одного счетчика машины $2cM$ можно воспользоваться теоремой 11 об ограниченности односчетчиковой машины Минского. □

Поскольку преобразования, приведенные в доказательстве теоремы 2, исключают запуск машины $2cM$ из нулевой начальной конфигурации, докажем неразрешимость и проблемы ограниченности двухсчетчиковых машин Минского для нулевой начальной конфигурации.

Теорема 5. *Проблема ограниченности двухсчетчиковых машин Минского с нулевой начальной конфигурацией неразрешима.*

Доказательство проводится сведением проблемы ограниченности двухсчетчиковой машины Минского для произвольной начальной конфигурации к данной проблеме.

Пусть $2cM = (\{q_0, \dots, q_n\}, \{x_1, x_2\}, \{\delta_0, \dots, \delta_{n-1}\})$ — двухсчетчиковая машина Минского с начальной конфигурацией (q_0, c_1, c_2). Построим из машины $2cM$ новую счетчиковую машину $2cM'$. Добавим новые состояния, включая новое начальное q'_0, и новые правила переходов следующего вида:

(δ'_0) q'_0: $\mathsf{if}\ x_1 > 0\ \mathsf{then}\ (x_1 := x_1 - 1;\ \mathsf{goto}\ q'_0)\ \mathsf{else\ goto}\ q'_1;$

(δ'_1) q'_1: $\mathsf{if}\ x_2 > 0\ \mathsf{then}\ (x_2 := x_2 - 1;\ \mathsf{goto}\ q'_1)\ \mathsf{else\ goto}\ q'_2.$

q'_2: $x_1 := x_1 + c_1;\ \mathsf{goto}\ q'_3;$

q'_3: $x_2 := x_2 + c_2;\ \mathsf{goto}\ q_0;$

где выражение вида $x_j := x_j + c_j$ $(1 \le j \le 2)$ интерпретируется как последовательность из c_j команд $x_j := x_j + 1$. Построенная таким образом машина $2cM'$, начав работу из $(q'_0, 0, 0)$, через некоторое время будет вести себя как машина $2cM$, запущенная из конфигурации (q_0, c_1, c_2).

Получили, что машина $2cM'$ будет ограниченной при нулевой начальной конфигурации $(q'_0, 0, 0)$ тогда и только тогда, когда ограничена машина $2cM$, запущенная из (q_0, c_1, c_2). □

Рассмотрим теперь двойственную к проблеме ограниченности (соот. при нулевой конфигурации) задачу существования неограниченного исполнения (соот. из нулевой конфигурации) двухсчетчиковой машины Минского. Перефразировав известную теорему Поста (см., например, в [3]), можно утверждать, что, если проблема и её дополнение (двойственная проблема) являются частично разрешимыми, следовательно, они обе разрешимы. А поскольку проблема ограниченности (как и проблема ограниченности для нулевой начальной конфигурации) является частично разрешимой, то получаем, что проблема неограниченного пути не может быть частично разрешимой. Таким образом, справедлива следующая теорема.

Теорема 6. *Для двухсчетчиковых машин Минского (с нулевой начальной конфигурацией) проблема неограниченного пути не является частично разрешимой.*

3. Тотальная ограниченность

Теорема 7. *Для трехсчетчиковых машин Минского проблема тотальной ограниченности не является частично разрешимой.*

Доказательство проводится методом сведения проблемы зацикливания из нулевой начальной конфигурации для двухсчетчиковой машины Минского к данной проблеме.

Итак, пусть $2cM = (\{q_0, \ldots, q_n\}, \{x_1, x_2\}, \{\delta_0, \ldots, \delta_{n-1}\})$ — двухсчетчиковая машина Минского. Преобразуем $2cM$ в машину Минского $3cM$ с тремя счетчиками следующим образом. Добавим новый счетчик x_3. И для каждого состояния q_i, $0 \le i \le n-1$, добавим состояния q_i^1, q_i^2, правила переходов δ_i^1, δ_i^2 и изменим δ_i следующим образом.

Для состояния первого типа имеем:

(δ_i) $q_i\colon$ $x_j := x_j + 1;$ goto $q_i^1;$ $\quad(\delta_i^1)$ $q_i^1\colon$ if $x_3 > 0$ then $(x_3 := x_3 - 1;$ goto $q_k)$ $\mathsf{else\ goto}$ q_f.

Для состояния второго типа:

(δ_i) $q_i\colon$ if $x_j > 0$ then $(x_j := x_j - 1;$ goto $q_i^1)$ $\mathsf{else\ goto}$ q_i^2;

(δ_i^1) $q_i^1\colon$ if $x_3 > 0$ then $(x_3 := x_3 - 1;$ goto $q_k)$ $\mathsf{else\ goto}$ q_f;

(δ_i^2) $q_i^2\colon$ if $x_3 > 0$ then $(x_3 := x_3 - 1;$ goto $q_l)$ $\mathsf{else\ goto}$ q_f.

Новое состояние q_f сделаем финальным состоянием машины $3cM$. А для старого финального состояния (машины $2cM$) построим петлю следующего вида:

(δ_n) $q_n\colon$ $x_3 := x_3 + 1;$ goto q_n.

И наконец, добавим еще пять состояний q_0', q_1', q_2', q_3', q_4' и пять переходов δ_0', δ_1', δ_2', δ_3', δ_4', ведущие в конечном итоге в старое начальное (для машины $2cM$) состояние q_0:

(δ_0') $q_0'\colon$ if $x_3 > 0$ then $(x_3 := x_3 - 1;$ goto $q_0')$ $\mathsf{else\ goto}$ q_1';

(δ_1') $q_1'\colon$ if $x_1 > 0$ then $(x_1 := x_1 - 1;$ goto $q_2')$ $\mathsf{else\ goto}$ q_3';

(δ_2') $q_2'\colon$ $x_3 := x_3 + 1;$ goto q_1';

(δ_3') $q_3'\colon$ if $x_2 > 0$ then $(x_2 := x_2 - 1;$ goto $q_4')$ $\mathsf{else\ goto}$ q_0;

(δ_4') $q_4'\colon$ $x_3 := x_3 + 1;$ goto q_3'.

Состояние q_0' установим новым начальным состоянием машины $3cM$.

Машина $3cM$ будет тотально ограничена тогда и только тогда, когда машина $2cM$ имеет бесконечное исполнение из начальной конфигурации $(q_0, 0, 0)$. Действительно, если запустить машину $3cM$ из конфигурации (q_0', c_1, c_2, c_3), то она через $c_3 + 2c_1 + 2c_2$ переходов будет ровно $c_1 + c_2$ шагов симулировать работу машины $2cM$, запущенную из $(q_0, 0, 0)$. Если за это время машина $2cM$ остановится (перейдет в состояние q_n), то машина $3cM$ зациклится с помощью специально введенной петли для состояния q_n, при этом далее по исполнению значение счетчика x_3 будет бесконечно возрастать, т. е. в таком случае машина $3cM$ не будет ограниченной. Если же $2cM$ не останавливается после $c_1 + c_2$ шагов, то машина $3cM$ завершает работу и переходит в свое заключительное состояние q_f. В этом случае для входа (q_0', c_1, c_2, c_3) машина $3cM$ является ограниченной. Таким образом, счетчиковая машина $3cM$ будет всегда (при любом входе) ограничена тогда и только тогда, когда машина $2cM$ имеет бесконечное исполнение из $(q_0, 0, 0)$. Получили, что проблема тотальной ограниченности для трехсчетчиковых машин Минского не является частично разрешимой, так как не является частично разрешимой проблема зацикливания из нулевой начальной конфигурации для двухсчетчиковых машин Минского. □

Теорема 8. *Для двухсчетчиковых машин Минского проблема тотальной ограниченности не является частично разрешимой.*

Доказательство. Применяя преобразования, приведенные в доказательстве теоремы 2, можно построить для машины $3cM$ двухсчетчиковую машину $2cM$, которая будет тотально ограничена тогда и только тогда, когда тотально ограничена машина $3cM$. □

Теорема 9. *Проблема ограниченности хотя бы при одном входе для двухсчетчиковых машин Минского частично разрешима.*

Доказательство. Для построения частичного алгоритма применяется известная техника «поиска в ширину». Упорядочим множество всех возможных входов для машины $2cM$ следующим образом: $(0,0)$, $(1,0)$, $(0,1)$, $(0,2)$, $(1,1)$, $(2,0)$, $(3,0)$, $(2,1)$, $(1,2)$, $(0,3)$ и т. д. На каждом шаге будем запускать новый экземпляр машины Минского $2cM$ с новым входом (взятым в соответствии с указанным порядком), одновременно совершая по одному переходу для каждого из уже запущенных экземпляров машины $2cM$. Таким образом, после n шагов алгоритма будет запущено n экземпляров машины $2cM$, причем к этому моменту первый запущенный экземпляр машины совершит n переходов, второй — $(n-1)$ и т. д. После каждого шага для всех запущенных экземпляров машины $2cM$ проверяются условия ограниченности (см. замечания к теореме 3. Если существует хотя бы один вход, на котором машина $2cM$ ограничена, то рано или поздно он будет найден. Однако, если такого входа не существует, алгоритм будет работать бесконечно долго. □

Теорема 10. *Проблемы тотальной неограниченности и неограниченности хотя бы при одном входе для двухсчетчиковых машин Минского не являются частично разрешимыми.*

Доказательство. Проблема тотальной неограниченности является двойственной к проблеме ограниченности хотя бы при одном входе и, следовательно, по теореме Поста не может быть частично разрешимой. Проблема же неограниченности хотя бы на одном входе двойственна к проблеме тотальной ограниченности. Но тотальная ограниченность не является частично разрешимой, поэтому воспользуемся методом сведения.

Итак, произведем сведение проблемы неограниченного пути из начальной конфигурации для двухсчетчиковых машин Минского к проблеме неограниченности хотя бы на одном входе.

Пусть $2cM = (\{q_0,\ldots,q_n\}, \{x_1,x_2\}, \{\delta_0,\ldots,\delta_{n-1}\})$ — двухсчетчиковая машина Минского с начальной конфигурацией $(q_0,0,0)$. Построим из машины $2cM$ новую счетчиковую машину $2cM'$. Добавим новые состояния, включая новое начальное q_0', и новые правила переходов следующего вида:

(δ_0') q_0': $\mathsf{if}\ x_1 > 0\ \mathsf{then}\ (x_1 := x_1 - 1;\ \mathsf{goto}\ q_0')\ \mathsf{else\ goto}\ q_1'$;

(δ_1') q_1': $\mathsf{if}\ x_2 > 0\ \mathsf{then}\ (x_2 := x_2 - 1;\ \mathsf{goto}\ q_1')\ \mathsf{else\ goto}\ q_0$.

Построенная таким образом машина $2cM'$, начав работу из произвольной начальной конфигурации (q_0', c_1, c_2), через некоторое время будет вести себя как машина $2cM$, запущенная из $(q_0,0,0)$.

Машина $2cM'$ будет неограниченной хотя бы на одном входе (а на самом деле тотально неограниченной) тогда и только тогда, когда машина $2cM$ имеет неограниченное исполнение из $(q_0,0,0)$. □

4. Ограниченность для односчетчиковой машины Минского

Счетчиковая машина Минского имеет всего лишь одно исполнение. Это исполнение может быть либо конечным, либо бесконечным.

Пусть $1cM = (Q, \{x\}, \Delta)$ — односчетчиковая машина Минского. Не ограничивая общности рассуждений, будем всегда полагать в качестве начальной конфигурацию $(q_0,0)$, поскольку для любой односчетчиковой машины Минского $1cM$ с начальной конфигурацией (q_0,n), где $n \in \mathbb{N}$, с помощью n команд первого типа $(x := x+1)$ легко построить машину $1cM'$ такую, что, начав работу из новой начальной конфигурации $(q_0',0)$, через n шагов она будет вести себя в точности так же, как машина $1cM$, запущенная из (q_0,n).

Конечное исполнение машины $1cM$ обозначим π, а бесконечное — $\rho = (\mathsf{q}_0,0)(\mathsf{q}_1,n_1)(\mathsf{q}_2,n_2)(\mathsf{q}_3,n_3)\ldots$, где (q_i,n_i) — это i-я конфигурация исполнения ρ (i — номер текущей конфигурации в последовательности), q_i — состояние из множества Q, а n_i — значение счетчика x. Если машина $1cM$ имеет бесконечное исполнение ρ, будем писать $\mathrm{ZERO}(1cM)$ для обозначения множества номеров конфигураций исполнения ρ, в которых проверка на ноль прошла успешно, т. е. множество $\mathrm{ZERO}(1cM)$ будет состоять из номеров i конфигураций вида $(\mathsf{q}_i,0)$, где q_i — состояние второго типа: $\mathrm{ZERO}(1cM) \overset{\text{def}}{=} \{0\} \cup \{i > 0 : n_i = n_{i+1} = 0\}$.

Лемма 1. *Пусть машина Минского $1cM$ имеет бесконечное исполнение ρ. Выберем элементы $i < j$ множества $\mathrm{ZERO}(1cM)$, для которых не существует такого $k \in \mathrm{ZERO}(1cM)$, что $i < k < j$. Тогда будет выполняться следующее неравенство: $(j-i) \leq |Q|^2/4$.*

Доказательство по существу устанавливает, что значение счетчика не может выйти за пределы $|Q|$ между двумя конфигурациями с успешной проверкой на ноль в исполнении ρ.

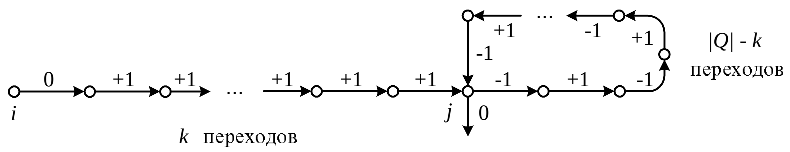

Рис. 1. Структура максимального по длине отрезка пути между конфигурациями $(\mathsf{q}_i, 0)$ и $(\mathsf{q}_j, 0)$

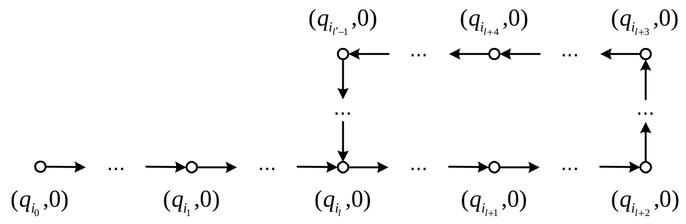

Рис. 2. Исполнение, соответствующее последовательности $i_0 < i_1 < i_2 < \ldots < i_l < \ldots < i_{l'} < \ldots$

Сначала заметим, что не существует $i < k < k' < j$ таких, что $\mathsf{q}_k = \mathsf{q}_{k'}$ и $n_k \leq n_{k'}$. Действительно, если бы это было не так, то, поскольку не существует успешной проверки на ноль в последовательности $(\mathsf{q}_{i+1}, n_{i+1}) \ldots (\mathsf{q}_k, n_k) \ldots (\mathsf{q}_{k'}, n_{k'})$, а исполнение машины $1cM$ детерминировано, мы бы получили бесконечный путь из конфигурации $(\mathsf{q}_{k'}, n_{k'})$ без успешной проверки на ноль на всем его протяжении, что противоречит существованию конфигурации (q_j, n_j). Отсюда, если найдутся такие $i < k < k' < j$, что $\mathsf{q}_k = \mathsf{q}_{k'}$, тогда $n_{k'} < n_k$.

Теперь предположим, что существует k, $i < k < j$, такое, что $n_k \geq |Q|$. Тогда мы можем извлечь подпоследовательность $(\mathsf{q}_{i_0}, n_{i_0}) \ldots (\mathsf{q}_{i_s}, n_{i_s})$ из последовательности конфигураций $(\mathsf{q}_i, n_i) \ldots (\mathsf{q}_{n_k}, n_k)$ такую, что $i_0 = i$, $i_s = k$ и для любого l, $0 \leq l < s$, имеем $n_{i_{l+1}} = n_{i_l} + 1$. Следовательно, существуют l и l' такие, что $\mathsf{q}_{i_l} = \mathsf{q}_{i_{l'}}$ и $n_{i_l} < n_{i_{l'}}$, а это противоречит предыдущему замечанию. Отсюда для любого $k \in \{i, \ldots, j\}$ имеем $n_k < |Q|$.

Построим самый длинный отрезок исполнения, который возможен между конфигурациями с номерами i и j. Этот отрезок будет состоять из двух частей: прямого участка и цикла (см. рис. 1). Прямой участок представляет собой последовательность из k переходов с помощью команды $x := x + 1$, при этом задействованы соответственно k состояний первого типа. Таким образом, следующий за этим участком цикл начинает свою работу в конфигурации (q_k, k). После нескольких полных проходов по циклу он завершается (или не завершается) в конфигурации $(\mathsf{q}_j, 0)$, при этом $\mathsf{q}_k = \mathsf{q}_j$. С использованием оставшихся $|Q| - k - 1$ состояний (финальное состояние в расчет не принимается), можно построить такой цикл, который, начав работу при значении счетчика равном k, будет при полном проходе в итоге уменьшать значение счетчика лишь на единицу. В этом цикле количество состояний первого и второго типов должно быть равным (без учёта состояния q_j). Таким образом, после k переходов цикл совершит k витков длиной в $|Q| - k - 1$ шагов каждый. Следовательно, количество шагов от i-й до j-й конфигурации в исполнении оценивается как $k + k \cdot (|Q| - k - 1) = k \cdot (|Q| - k)$, где $k < |Q|$. Верхняя оценка достигается при $k = |Q|/2$.

Таким образом, в итоге получаем, что $(j - i) \leq |Q|^2/4$. □

Лемма 2. *Пусть машина Минского* $1cM$ *имеет бесконечное исполнение* ρ. *Тогда существуют такие константы* K_1, K_2 *и* K_3, *что* $K_1 + K_2 \leq |Q|^2/4$, $K_3 \leq K_2 \leq |Q|$ *и для любого* $i \geq K_1$ *выполняется* $(\mathsf{q}_{i+K_2}, n_{i+K_2}) = (\mathsf{q}_i, n_i + K_3)$.

Доказательство. Рассмотрим случай, когда множество $\mathrm{ZERO}(1cM)$ является бесконечным. Построим из элементов множества $\mathrm{ZERO}(1cM)$ бесконечную последовательность $i_0 < i_1 < i_2 < \ldots$ $(i_0 = 0)$. В этой последовательности обязательно найдутся такие $l, l' \leq |Q|$, что $(\mathsf{q}_{i_l}, 0) = (\mathsf{q}_{i_{l'}}, 0)$. Следовательно, исполнение машины $1cM$, выходящее из конфигурации $(\mathsf{q}_{i_0}, 0)$, обязательно зациклится (см. рис. 2). Причем в этом исполнении все состояния, входящие в конфигурации с номерами из последовательности $i_0 < i_1 < i_2 < \ldots < i_l < \ldots < i_{l'-1}$, попарно различны. Будем называть такие конфигурации и состояния выделенными. Оценим длину отрезка исполнения, находящегося между конфигурациями $(\mathsf{q}_{i_0}, 0)$ и $(\mathsf{q}_{i_{l'}}, 0)$. Во-первых, число выделенных состояний не превышает $|Q|$. Если предположить, что все отрезки между выделенными конфигурациями имеют такую же структуру, как показано на рис. 1, то искомую длину всего отрезка исполнения по лемме 1 можно оценить как $|Q|^3$. Однако это будет очень грубой оценкой, которую можно значительно уменьшить. Дело в том, что каждый отрезок между выделенными конфигурациями имеет свой уникальный прямой участок, состояния которого не встречаются ни в каком другом отрезке. В противном случае нарушалось бы условие уникальности выделенных состояний, так как происходил бы «прыжок» в другой отрезок без участия выделенного состояния, помечающего конечную для данного отрезка конфигурацию. Суммарная длина таких прямых участков составляет $k_1 + k_2 + \ldots + k_n$ и ограничена числом $|Q|$, где n — количество участков, а k_i — длина i-го участка. Таким образом, $k_1 + k_2 + \ldots + k_n = n \times k \leq |Q|$, где k — среднее арифметическое длин прямых участков. Учитывая возможную циклическую структуру каждого отрезка, получаем оценку $k_1 \cdot (|Q| - n \cdot k) + k_2 \cdot (|Q| - n \cdot k) + \ldots + k_n \cdot (|Q| - n \cdot k) = (|Q| - n \cdot k) \cdot (k_1 + k_2 + \ldots + k_n) = (|Q| - n \cdot k) \cdot n \cdot k \leq |Q|^2/4$. Итак, получили $i_{l'} \leq |Q|^2/4$. Осталось взять $K_1 = i_l$, а $K_2 = i_{l'} - i_l$.

Теперь рассмотрим случай, когда множество ZERO$(1cM)$ конечно. Пусть ZERO$(1cM) = \{0, i_1, \ldots, i_l\}$ для $l \leq |Q| - 1$ (если $l \geq |Q|$, имеем предыдущий случай). Далее, для всех $i_l \leq k < k'$, если $\mathsf{q}_k = \mathsf{q}_{k'}$, то $n_k \leq n_{k'}$, поскольку на пути из $(\mathsf{q}_{i_l}, n_{i_l})$ не существует конфигурации с успешной проверкой на ноль. Более того, всегда найдутся такие $i_l \leq k < k' \leq i_l + |Q|$, что $q_k = q_{k'}$ и, следовательно, $n_k \leq n_{k'}$. Возьмем $K_1 = k$, $K_2 = k' - k$ и $K_3 = n_{k'} - n_k$. Отметим, что все состояния, входящие в конфигурации из отрезка между $(\mathsf{q}_{i_l}, 0)$ и $(\mathsf{q}_{k'}, n_{k'})$, попарно различны и не встречаются в исполнении до конфигурации $(\mathsf{q}_{i_l}, 0)$. Обозначим K' количество таких состояний. Исходя из соображений, приведенных в предыдущем абзаце, получим следующую оценку: $K_1 + K_2 \leq (|Q| - l \cdot k_l - K') \cdot l \cdot k_l + K' \leq |Q|^2/4$; $K_3 \leq K_2 \leq |Q|$, поскольку $k' - k \leq |Q|$. □

Из этой леммы следует, что бесконечный путь ρ машины $1cM$ может быть представлен всего лишь первыми $K_1 + K_2$ конфигурациями.

Замечания по леммам 1 и 2. Доказательства лемм почти полностью повторяют доказательства этих утверждений, приведенных в статье [8]. Отличие состоит в оценке времени работы алгоритма. В оригинальной статье приведена оценка $(K_1 + K_2) \leq |Q|^3$. □

Теорема 11. *Проблема ограниченности односчетчиковой машины Минского разрешима за время* $O(|Q|^2)$.

Доказательство. Как следует из доказательств лемм 1 и 2, за время, ограниченное числом $|Q|^2/4$, может быть реализовано полное конечное исполнение π или же построен конечный отрезок ρ' бесконечного исполнения ρ, который будет иметь длину в $K_1 + K_2$ конфигураций. В случае наличия исполнения π ограниченность очевидна. Если же была построена последовательность ρ', ограниченность счетчика на всем исполнении зависит от значения K_3. При $K_3 = 0$ делаем вывод, что машина $1cM$ ограничена, иначе $(0 < K_3 \leq |Q|)$ значение счетчика после K_1 переходов от начала исполнения начинает бесконечно возрастать на K_3 единиц с периодичностью в K_2 конфигураций. □

Теорема 12. *Проблема тотальной ограниченности односчетчиковой машины Минского разрешима за время* $O(|Q|^3)$.

Доказательство. Как следует из доказательства леммы 2, односчетчиковая машина $1cM$ не будет ограниченной тогда и только тогда, когда существует увеличивающий значение счетчика цикл из некоторой достижимой конфигурации $(\mathsf{q}, 0)$, где q — состояние второго типа, или начальной конфигурации (q_0, n), который не содержит успешных проверок на ноль. Таким образом, условие тотальной ограниченности нарушается лишь в том случае, когда существует вход, ведущий в некоторую конфигурацию $(\mathsf{q}, 0)$ (второго типа), представляющую собой начало увеличивающего цикла, либо сам вход порождает увеличивающий цикл, минуя какие-либо нулевые конфигурации с состояниями второго типа. Чтобы подтвердить или опровергнуть наличие такой возможности, проделаем следующую последовательность шагов.

Во-первых, необходимо проверить для всех состояний второго типа q_i, $i \leq |Q|$, достижимость увеличивающего цикла из конфигурации $(q_i, 0)$. Во-вторых, потребуется построить множество достижимых конфигураций из верхнего конуса $\uparrow(q_0, 0) = \{(q_0, c) \mid c \geq 0\}$, представленного своим конечным базисом $(q_0, 0)$, причем построение достижимых из этого конуса конфигураций должно осуществляться без привлечения второй части правила переходов второго типа, т. е. без перехода по нулю, если уменьшение значения счетчика более не возможно, то новая конфигурация не будет порождаться, и считается недостижимой из текущей конфигурации. Фактически необходимо построить множество $Post^*(\uparrow(q_0, 0))$ всех возможных достижимых из $\uparrow(q_0, 0)$ конфигураций для системы переходов следующего вида, построенной из машины $1cM$: для состояний первого типа — $\delta_i:\ x_1 := x_1 + 1;$ $\mathsf{goto}\ q_k$ (правило остается без изменений), для состояний второго типа — $\delta_i:\ \mathsf{if}\ x_1 > 0\ \mathsf{then}\ (x_1 := x_1 - 1;\ \mathsf{goto}\ q_k)$ (удаляется else). Построенная таким образом система переходов будет обладать свойством, что для любых двух конфигураций одного состояния q_i $(q_i, c_1) < (q_i, c_2)$, где $c_1 < c_2$, и любого перехода δ_i, переводящего (q_i, c_1) в (q_j, c_1'), существует этот же переход δ_i из (q_i, c_2) в (q_j, c'_2) такой, что $(q_j, c'_1) < (q_j, c'_2)$. Более того, для каждого текущего состояния существует только один переход. Это свойство позволяет строить множество $Post^*(\uparrow(q_0, 0))$ следующим образом. Если из состояния q_0 существует переход первого типа $\delta_i:\ x_1 := x_1 + 1;\ \mathtt{goto}\ q_k$, то множество $Post^1(\uparrow(q_0, 0))$ непосредственных последователей конуса $\uparrow(q_0, 0)$ (включая сам конус) имеет вид $\uparrow(q_0, 0) \cup \uparrow(q_k, 1)$, если переход второго типа $\delta_i:\ \mathtt{if}\ x_1 > 0\ \mathtt{then}\ (x_1 := x_1 - 1;\ \mathtt{goto}\ q_k)$ — $Post^1(\uparrow(q_0, 0)) = \uparrow(q_0, 0) \cup \uparrow(q_k, 0)$. Текущим состоянием становится q_k. $Post^2(\uparrow(q_0, 0)) = Post^1(\uparrow(q_0, 0)) \cup Post^1(\uparrow(q_k, c))$, где c в зависимости от предыдущего перехода равно 0 или 1. Пусть $Post^r(\uparrow(q_0, 0))$ — множество достижимых за r шагов конфигураций из конуса $\uparrow(q_0, 0)$. Множество $Post^r(\uparrow(q_0, 0))$ задается своим конечным базисом, имеющим вид $\{(q_{i_1}, c_{i_1}), \ldots, (q_{i_l}, c_{i_l})\}$, где все состояния q_{i_j}, $i_j \le |Q|$, попарно различны. Пусть состояние q_{i_l} является текущим. Тогда $Post^{r+1}(\uparrow(q_0, 0)) = Post^r(\uparrow(q_0, 0)) \cup Post^1(\uparrow(q_{i_l}, c_{i_l}))$. Если q_{i_l} — состояние первого типа, то $Post^1(\uparrow(q_{i_l}, c_{i_l})) = \uparrow(q_{i_l}, c_{i_l}) \cup \uparrow(q_k, c_{i_l} + 1)$, если q_{i_l} является состоянием второго типа, то в случае $c_{i_l} > 0$ имеем $Post^1(\uparrow(q_{i_l}, c_{i_l})) = \uparrow(q_{i_l}, c_{i_l}) \cup \uparrow(q_k, c_{i_l} - 1)$, иначе $Post^1(\uparrow(q_{i_l}, 0)) = \uparrow(q_{i_l}, 0) \cup \uparrow(q_k, 0)$. Если состояние q_k уже задействовано в базисе конуса $Post^r(\uparrow(q_0, 0))$, т.е. $k = i_j$, $1 \le j \le l$, то при объединении конусов $Post^r(\uparrow(q_0, 0))$ и $Post^1(\uparrow(q_{i_l}, c_{i_l}))$ в базис конуса $Post^{r+1}(\uparrow(q_0, 0))$ поместим $(q_k, \min(c_{old}, c_{new}))$, где c_{old} и c_{new} — старое и новое значение счетчика в состоянии q_k соответственно. Заметим, что если новое значение больше старого, следовательно, существует вход машины $1cM$, который порождает увеличивающий цикл, нарушающий условие тотальной ограниченности. Алгоритм останавливается с отрицательным ответом. В противном случае получаем последовательность конусов $Post^1(\uparrow(q_0, 0)) \subseteq Post^2(\uparrow(q_0, 0)) \subseteq \ldots \subseteq Post^r(\uparrow(q_0, 0)) \subseteq Post^{r+1}(\uparrow(q_0, 0)) \subseteq \ldots$, которая обязательно стабилизируется. Более того, в худшем случае на это потребуется $|Q|^2/4$ шагов. Множество, к которому сходится эта последовательность, и будет искомым результатом $Post^*(\uparrow(q_0, 0))$.

После всех построений остается лишь проверить, входят ли те нулевые конфигурации, которые порождают увеличивающий цикл, в множество $Post^*(\uparrow(q_0, 0))$, что легко сделать, так как оно будет представлено конечным базисом из $|Q|$ элементов максимум. Поскольку количество различных нулевых конфигураций ограничено числом $|Q|$, а проверка на увеличивающий цикл для каждой конфигурации проводится за время $O(|Q|^2)$, общее время работы алгоритма можно оценить как $O(|Q|^3)$. Мы ограничимся этой оценкой. Однако заметим, что она является очень грубой и может, как нам кажется, быть уменьшена до оценки $O(|Q|^2)$. □

Список литературы

1. Кларк, Э. М. Верификация моделей программ: Model Checking / Э. М. Кларк, О. Грамберг, Д. Пелед. – МЦНМО, 2002. – 416 с.

2. Котов, В. Е. Сети Петри / В. Е. Котов — М.: Наука, 1984. — 160 с.

3. Котов, В. Е. Теория схем программ / В. Е. Котов, В. К. Сабельфельд. — М.: Наука, 1991. — 248 с.

4. Кузьмин, Е. В. Структурированные системы переходов / Е. В. Кузьмин, В. А. Соколов. – М.: ФИЗМАТЛИТ, 2006. – 178 с.

5. Минский, М. Вычисления и автоматы / М. Минский — М.: Мир, 1971. — 268 с.

6. Хопкрофт, Д. Введение в теорию автоматов, языков и вычислений / Д. Хопкрофт, Р. Мотвани, Д. Ульман. — 2-е изд.: Пер. с англ. — М.: Вильямс, 2002. — 528 с.

7. Chitic, C. On validation of XML streams using finite state machines / C. Chitic, D. Rosu. — WebDB, Paris, 2004. — P. 85-90.

8. Demri, S. Model checking freeze LTL over one-counter automata / S. Demri, R. Lazic, A. Sangnier // Proceedings of the 11th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'08), LNCS. — 2008. To appear. (http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/DLS-fossacs08.pdf).

9. Dufourd, C. Boundedness of Reset P/T nets / C. Dufourd, P. Jancar, Ph. Schnoebelen // Proc. ICALP'99, LNCS 1644. — 1999. — P. 301–310.

10. Lafourcade, P. Intruder deduction for AC-like equational theories with homomorphisms / P. Lafourcade, D. Lugiez, R. Treinen. — RTA'05, LNCS 3467, Springer, 2005. — P. 308–322.

11. Mayr R. Lossy counter machines / R. Mayr // Tech. Report TUM-I9827, Institut für Informatik, TUM, Germany, October 1998.

12. Wakatsuki, M. Polynomial time identification of strict deterministic restricted one-counter automata in some class from positive data / M. Wakatsuki, K. Teraguchi, E. Tomita. — ICGI 2004, Athens, LNAI 3264, Springer, 2004. — P. 260–272.

On the decidability of boundedness problems for counter Minsky machines

Kuzmin E. V., Chalyy D. Ju.

In the paper the decidability of boundedness problems for counter Minsky machines is investigated. It is proved, that for Minsky machines with two counters the boundedness is partial decidable, but for the total boundedness problem does not even exist a semidecision algorithm. On the other hand, for one-counter Minsky machines all these problems are polinomial (quantitatively of local machine states) decidable.