Math-Net.Ru

Общероссийский математический портал

Л. М. Павлоцкая, О машинах Тьюринга, связанных относительно свойства неразрешимости проблемы остановки, *Матем. заметки*, 2002, том 71, выпуск 5, 732–741

DOI: <https://doi.org/10.4213/mzm381>

Использование Общероссийского математического портала Math-Net.Ru подразумевает, что вы прочитали и согласны с пользовательским соглашением
<http://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 54.198.64.247

26 апреля 2023 г., 15:23:21

УДК 510.53+512.54.05

О МАШИНАХ ТЬЮРИНГА, СВЯЗНЫХ ОТНОСИТЕЛЬНО СВОЙСТВА НЕРАЗРЕШИМОСТИ ПРОБЛЕМЫ ОСТАНОВКИ

Л. М. Павлоцкая

Хорошо известна задача поиска машины Тьюринга с неразрешимой проблемой остановки, программа которой содержит минимальное число инструкций. Очевидно, для такой машины выполняется следующее условие: если из ее программы удалить хотя бы только одну инструкцию, получим машину с разрешимой проблемой остановки. В настоящей работе машины Тьюринга с неразрешимой проблемой остановки, обладающие вышеформулированным свойством, названы связными. В работе получены некоторые общие свойства таких машин и простейшие следствия из них, относящиеся к минимальной машине с неразрешимой проблемой остановки.

Библиография: 2 названия.

1. Введение. Мы будем рассматривать машины Тьюринга с лентой, бесконечной в обоих направлениях. Пусть $Q = \{q_1, \dots, q_p\}$ – алфавит состояний головки, а $\mathcal{A} = \{0, a_1, \dots, a_m\}$ – алфавит входных символов. Символ “0” называется *пустым символом*. Программы рассматриваемых машин содержат инструкции вида $q_i a_k T a_l q_m$, где T есть либо символ правого движения (R), либо символ левого движения (L).

Таким образом, мы рассматриваем машины Тьюринга, головка которых за один такт работы выполняет одну из инструкций указанного вида, и если это инструкция $q_i a_k T a_l q_m$, то она выполняет следующие действия: заменяет в обозреваемой ею ячейке символ a_k символом a_l , переходит из состояния q_i в состояние q_m и перемещается вдоль ленты на одну ячейку вправо, если $T = R$, или влево, если $T = L$.

Под *программой* $\mathcal{P}$ будем понимать произвольную совокупность инструкций указанного вида, удовлетворяющую условию: никакие две инструкции в ней не начинаются одной и той же парой $q_i a_k$. Будем считать, что инструкции программы занумерованы числами $1, 2, \dots$, обозначать инструкцию с номером j символом p_j и записывать ее в виде: $p_j = \alpha_j x_j T_j y_j \beta_j$, где $\alpha_j = q_i$, $x_j = a_k$, $T_j = T$, $y_j = a_l$, $\beta_j = q_m$, если $p_j = q_i a_k T a_l q_m$. Число инструкций в программе $\mathcal{P}$ будем называть ее *длиной* и обозначать через $l(\mathcal{P})$.

Пусть ячейки ленты занумерованы следующим образом. Некоторой выделенной ячейке присвоен номер 0, и эта ячейка обозначена J_0 . Ячейки справа от J_0 в порядке их следования слева направо занумерованы числами $1, \dots, m, \dots$ и обозначены $J_1, \dots, J_m, \dots$. Ячейки слева от J_0 в порядке их следования справа налево занумерованы числами

$-1, \dots, -m, \dots$ и обозначены $J_{-1}, \dots, J_{-m}, \dots$. Обозначим $[a, b]$ участок ленты, образованный ячейками $J_a, J_{a+1}, \dots, J_b$.

Машина Z задана, если заданы совокупность алфавитов $\mathcal{A}$ и Q и программа $\mathcal{P}$ или, короче, $Z = \{\mathcal{A}, Q, \mathcal{P}\}$.

Множество всех слов в алфавите $\mathcal{A}$, включая пустое, обозначим $\mathcal{A}^*$. Длину слова u обозначим $|u|$. Любое слово вида uq_ia_kv , где $u \in \mathcal{A}^*$, $q_i \in Q$, $a_k \in \mathcal{A}$, $v \in \mathcal{A}^*$, назовем *конфигурацией машины* $Z = \{\mathcal{A}, Q, \mathcal{P}\}$.

Множество всех конфигураций машины Z обозначим $\mathcal{K}(Z)$ или просто $\mathcal{K}$, если из контекста ясно, о какой машине идет речь.

Для машины Z определим вычисление на ее конфигурации uq_ia_kv как процесс применения программы $\mathcal{P}$, начавшийся при следующем состоянии машины Z :

- 1) на участке ленты $[1, |ua_kv|]$ написано слово ua_kv ;
- 2) головка в состоянии q_i находится напротив ячейки $J_{|u|+1}$;
- 3) лента вне участка $[1, |ua_kv|]$ пустая, т.е. каждая ячейка J_m при $m > |ua_kv|$ или $m < 1$ содержит символ 0.

Определим конфигурацию машины Z в момент t вычисления на конфигурации uq_ia_kv , обозначив такую конфигурацию через $(uq_ia_kv)^t$.

Пусть $(uq_ia_kv)^0 = uq_ia_kv$.

Если $(uq_ia_kv)^t = u_1q_ja_mv_1$, то получим $(uq_ia_kv)^{t+1}$ следующим образом:

- 1) если программа машины Z содержит инструкцию $q_iamRa_pq_l$, то $(uq_ia_kv)^{t+1} = u_1a_pq_lv_1$, если v_1 непустое слово, и $(uq_ia_kv)^{t+1} = u_1a_pq_l0$, если v_1 пустое слово;
- 2) если $\mathcal{P}$ содержит инструкцию $q_ja_mLa_pq_l$, то $(uq_ia_kv)^{t+1} = u_2q_ia_r a_p v_1$, если $u_1 = u_2a_r$ при некотором u_2 , и $(uq_ia_kv)^{t+1} = q_l0a_p v_1$, если u_1 пустое слово;
- 3) если $\mathcal{P}$ не содержит инструкцию, начинающуюся парой q_ja_m , то процесс останавливается и конфигурация $(uq_ia_kv)^t$ называется *заключительной*.

Множество всех начальных конфигураций конечных вычислений обозначим $\mathcal{K}_{\text{fin}}(Z)$, или иногда просто $\mathcal{K}_{\text{fin}}$, а множество всех начальных конфигураций бесконечных вычислений обозначим $\mathcal{K}_{\text{inf}}(Z)$, или иногда просто $\mathcal{K}_{\text{inf}}$.

Далее будем записывать конфигурацию машины в виде слова uq_iv , считая при этом, что v непустое слово. Будем считать, что на множестве $(\mathcal{A} \cup Q)^*$ задана некоторая нумерация, например, алфавитная [1], и будем использовать обычные понятия рекурсивного и рекурсивно перечислимого словарных множеств. Очевидно, что $\mathcal{K}(Z)$ есть множество, рекурсивное для любой машины Z .

Проблема остановки машины Z – это проблема поиска алгоритма, с помощью которого для любой конфигурации машины Z можно определить, принадлежит она множеству $\mathcal{K}_{\text{fin}}(Z)$ или нет.

Если такой алгоритм существует, то проблема остановки машины Z разрешима, в противном случае такая проблема неразрешима.

При доказательстве различных утверждений будем использовать тезис Чёрча для частично вычислимых функций в следующих случаях. Указав алгоритм, с помощью которого можно определить для любой конфигурации $uq_iv \in \mathcal{K}$, принадлежит она множеству $\mathcal{K}_1 \subset \mathcal{K}$ или нет, будем считать множество $\mathcal{K}_1$ рекурсивным, не приводя формального доказательства. Указав алгоритм, применимый к любой конфигурации uq_iv , и такой, что его применение заканчивается получением результата только, если $uq_iv \in \mathcal{K}_1$, где $\mathcal{K}_1 \subset \mathcal{K}$, будем считать, что множество $\mathcal{K}_1$ рекурсивно перечислимо,

не проводя формального доказательства и в этом случае.

Среди всех машин Тьюринга с неразрешимой проблемой остановки существуют машины, в программе которых содержится минимальное число инструкций. Отсюда следует, что, удалив из программы такой машины хотя бы одну инструкцию, мы получим машину с разрешимой проблемой остановки. Таким свойством могут обладать и другие машины Тьюринга с неразрешимой проблемой остановки, не только простейшие из них. Мы установим некоторые общие свойства таких машин, назвав их связными относительно свойства неразрешимости проблемы остановки или, далее, просто связными.

Заметим, что если $\mathcal{P}_{\min}$ есть программа одной из простейших машин с неразрешимой проблемой остановки, то, добавив к ней одну подходящую инструкцию, получим программу простейшей машины с неразрешимой проблемой остановки в одном, специально выделенном и называемом заключительным, состоянии.

2. Свойства связных машин Тьюринга. Пусть $Z = \{\mathcal{A}, Q, \mathcal{P}\}$, где $\mathcal{P} = \{p_1, \dots, p_m\}$. Множество всех начальных конфигураций вычислений, в которых по меньшей мере один раз выполняется инструкция $p_i \in \mathcal{P}$, обозначим $\mathcal{K}_i$, а программу, полученную из программы $\mathcal{P}$ удалением i -й инструкции, обозначим $\mathcal{P}_i$.

ОПРЕДЕЛЕНИЕ 1. Назовем машину $Z = \{\mathcal{A}, Q, \mathcal{P}\}$ *связной*, если для нее проблема остановки неразрешима, а для любой из машин $Z_i = \{\mathcal{A}', Q', \mathcal{P}_i\}$ ($1 \leq i \leq m$) проблема остановки разрешима.

Здесь $\mathcal{A}'$ и Q' произвольные алфавиты. Легко доказать, что справедлива

ЛЕММА 1. Для связной машины Z каждое из множеств $\mathcal{K}_i(Z)$ рекурсивно.

ДОКАЗАТЕЛЬСТВО. Соответственно определению связной машины проблема остановки для машины $Z_i = \{\mathcal{A}, Q, \mathcal{P}_i\}$ разрешима и, следовательно, множества $\mathcal{K}_{\text{fin}}(Z_i)$ и $\mathcal{K}_{\text{inf}}(Z_i)$ рекурсивны. Пусть $uq_iv \in \mathcal{K}(Z)$. Если $uq_iv \in \mathcal{K}_{\text{inf}}(Z_i)$, то $uq_iv \notin \mathcal{K}_i(Z)$. Если $uq_iv \in \mathcal{K}_{\text{fin}}(Z_i)$, то найдем заключительную конфигурацию в вычислении машины Z_i на конфигурации uq_iv и определим состояние головки машины Z_i и символ в обозреваемой ею ячейке в момент остановки вычисления. Если они совпадают с парой $\alpha_i x_i$, то $uq_iv \in \mathcal{K}_i(Z)$, в противном случае $uq_iv \notin \mathcal{K}_i(Z)$. Используя эти соображения и алгоритмы решения проблем вхождения для множества $\mathcal{K}_{\text{inf}}(Z_i)$ и для множества $\mathcal{K}_{\text{fin}}(Z_i)$, легко составить алгоритм решения проблемы вхождения для множества $\mathcal{K}_i(Z)$, т.е. алгоритм, с помощью которого можно определить, принадлежит конфигурация uq_iv множеству $\mathcal{K}_i(Z)$ или нет, что и завершит доказательство леммы 1.

ОПРЕДЕЛЕНИЕ 2. Вычисление машины Z на конфигурации uq_iv назовем *правосторонним* (*левосторонним*), если в процессе такого вычисления головка ни разу не посещает ячейку $J_0(J_{|uv|+1})$.

Множество всех начальных конфигураций правосторонних (левосторонних) вычислений обозначим $\mathcal{K}^{(r)}(Z)$ или иногда просто $\mathcal{K}^{(r)}$ ($\mathcal{K}^{(l)}(Z)$ или иногда просто $\mathcal{K}^{(l)}$).

Как правостороннее, так и левостороннее вычисление будем называть также *односторонним*.

Вычисление на некоторой конфигурации назовем *двусторонним*, если в процессе такого вычисления головка посещает каждую ячейку ленты. Множество всех начальных конфигураций таких вычислений обозначим $\mathcal{K}^{(2)}$. Очевидно, $\mathcal{K}^{(2)}(Z) \subseteq \mathcal{K}_{\text{inf}}(Z)$.

ОПРЕДЕЛЕНИЕ 3. Бесконечное вычисление назовем *равномерно бесконечным*, если в нем каждая инструкция программы выполняется бесконечное число раз, и множество всех начальных конфигураций таких вычислений обозначим $\mathcal{K}_{(u)}$, а $\mathcal{K}_{\text{inf}} \setminus \mathcal{K}_{(u)}$ обозначим через $\mathcal{K}_{(c)}$.

Введем также обозначения для следующих множеств: $\mathcal{K}_{\text{inf}}^{(r)} = \mathcal{K}^{(r)} \cap \mathcal{K}_{\text{inf}}$, $\mathcal{K}_{\text{inf}}^{(l)} = \mathcal{K}^{(l)} \cap \mathcal{K}_{\text{inf}}$, $\mathcal{K}_{(u)}^{(r)} = \mathcal{K}^{(r)} \cap \mathcal{K}_{(u)}$, $\mathcal{K}_{(u)}^{(l)} = \mathcal{K}^{(l)} \cap \mathcal{K}_{(u)}$, $\mathcal{K}_{(u)}^{(2)} = \mathcal{K}^{(2)} \cap \mathcal{K}_{(u)}$.

ТЕОРЕМА 1. Для любой связной машины Z множество $\mathcal{K}_{(u)}$ отлично от рекурсивно перечислимого.

ДОКАЗАТЕЛЬСТВО. Так как $\mathcal{K}_{\text{inf}} = \mathcal{K}_{(u)} \cup \mathcal{K}_{(c)}$, а множество $\mathcal{K}_{\text{inf}}$ для связной машины не может быть рекурсивно перечислимым, то для доказательства теоремы достаточно показать, что для любой связной машины множество $\mathcal{K}_{(c)}$ рекурсивно перечислимо. Так как Z связная машина, то существует алгоритм B , с помощью которого для любой конфигурации $uq_iv \in \mathcal{K}$ можно определить, существует ли j такое, что $uq_iv \notin \mathcal{K}_j$. Проводя вычисление машины Z на конфигурации uq_iv , будем применять алгоритм B к каждой конфигурации $(uq_iv)^t$ до тех пор, пока не наступит момент t_j , в который с помощью алгоритма B будет определено, что $(uq_iv)^{t_j} \notin \mathcal{K}_j$. Последнее означает, что инструкция p_j не выполняется в вычислении ни в какой из моментов вычисления, начиная с момента t_j . Так как для машины $Z_j = \{\mathcal{A}, Q, \mathcal{P}_j\}$, где $\mathcal{A}$ и Q – алфавиты машины Z , проблема остановки разрешима, то с помощью соответствующего алгоритма можно определить, остановится процесс вычисления машины Z_j на конфигурации $(uq_iv)^{t_j}$ или нет. Если получен ответ нет, то $uq_iv \in \mathcal{K}_{(c)}$, если получен положительный ответ, то $uq_iv \in \mathcal{K}_{\text{fin}}$. В вычислении на любой конфигурации $uq_iv \in \mathcal{K}_{\text{fin}} \cup \mathcal{K}_{(c)}$ наступит момент, начиная с которого одна из инструкций программы ни разу не выполняется, поэтому с помощью описанного выше алгоритма для любой из таких конфигураций можно определить, будет ли она принадлежать одному из множеств $\mathcal{K}_{\text{fin}}$ или $\mathcal{K}_{(c)}$ и какому именно. Применение же описанного выше алгоритма к любой конфигурации $uq_iv \in \mathcal{K}_{(u)}$ никогда не останавливается и, следовательно, ни к какому результату не приводит. Из вышесказанного следует, что множество $\mathcal{K}_{(c)}$ рекурсивно перечислимо. Следовательно, множество $\mathcal{K}_{(u)}$ не может быть рекурсивно перечислимым, так как в противном случае рекурсивно перечислимым было бы множество $\mathcal{K}_{\text{inf}}$. Теорема 1 доказана.

ТЕОРЕМА 2. Для любой машины Z из того, что множества $\mathcal{K}_{(u)}^{(r)}$, $\mathcal{K}_{(u)}^{(l)}$ и $\mathcal{K}_u^{(2)}$ рекурсивно перечислимы, следует, что перечислимо множество $\mathcal{K}_{(u)}$.

ДОКАЗАТЕЛЬСТВО. Пусть $g(n)$ есть общерекурсивная функция, множество значений которой совпадает с множеством всех номеров конфигураций из множества $\mathcal{K}_{(u)}^{(1)} \cup \mathcal{K}_{(u)}^{(2)}$, где $\mathcal{K}_{(u)}^{(1)} = \mathcal{K}_{(u)}^{(r)} \cup \mathcal{K}_{(u)}^{(l)}$. Если $uq_iv \in \mathcal{K}_{(u)}^{(1)} \cup \mathcal{K}_{(u)}^{(2)}$, то $(uq_iv)^t \in \mathcal{K}_{(u)}^{(1)} \cup \mathcal{K}_{(u)}^{(2)}$ при любом t . Если $uq_iv \notin \mathcal{K}_{(u)}^{(1)} \cup \mathcal{K}_{(u)}^{(2)}$, но $uq_iv \in \mathcal{K}_{(u)}$, то существует t_0 такое, что $(uq_iv)^t \in \mathcal{K}_{(u)}^{(1)} \cup \mathcal{K}_u^{(2)}$ для любого $t \geq t_0$. Поэтому, проводя вычисление машины Z на конфигурации uq_iv и проверяя для каждого t , содержит ли множество $\{(uq_iv)^0, \dots, (uq_iv)^t\}$ конфигурацию с номером, принадлежащим множеству $\{g(0), \dots, g(t)\}$, мы найдем такую конфигурацию тогда и только тогда, когда $uq_iv \in \mathcal{K}_{(u)}$. Следовательно, существует частично вычислимая, а следовательно, и частично рекурсивная функция,

принимаяющая значение, равное единице для тех и только тех значений n , которые являются номерами конфигураций из множества $\mathcal{K}_{(u)}$. Отсюда получаем, что $\mathcal{K}_{(u)}$ есть множество рекурсивно перечислимое. Теорема 2 доказана.

Из теорем 1 и 2 получаем

СЛЕДСТВИЕ. Для связной машины Z по меньшей мере одно из множеств $\mathcal{K}_u^{(r)}$, $\mathcal{K}_u^{(l)}$ и $\mathcal{K}_{(u)}^{(2)}$ отлично от рекурсивно перечислимого.

ТЕОРЕМА 3. Для любой машины $Z = \{\mathcal{A}, Q, \mathcal{P}\}$ с неразрешимой проблемой остановки по меньшей мере одно из множеств $\mathcal{K}_{\text{inf}}^{(r)}$, $\mathcal{K}_{\text{inf}}^{(l)}$, $\mathcal{K}^{(2)}$ отлично от рекурсивно перечислимого.

Доказательство аналогично доказательству теоремы 2.

ТЕОРЕМА 4. Если для машины $Z = \{\mathcal{A}, Q, \mathcal{P}\}$ одно из множеств $\mathcal{K}_{\text{inf}}^{(r)}$, $\mathcal{K}_{\text{inf}}^{(l)}$ есть множество нерекурсивное, то существует алфавит $\mathcal{A}'$ такой, что проблема остановки машины $Z = \{\mathcal{A}', Q, \mathcal{P}\}$ неразрешима.

ДОКАЗАТЕЛЬСТВО. Пусть для $Z = \{\mathcal{A}, Q, \mathcal{P}\}$ множество $\mathcal{K}_{\text{inf}}^{(r)}$ нерекурсивное. Очевидно, для доказательства теоремы надо рассмотреть случай, когда проблема остановки для Z разрешима. В этом случае рекурсивным будет множество $\mathcal{K}^{(r)} \cap \mathcal{K}_{\text{fin}}$ и, следовательно, множество всех конфигураций, в вычислении на которых головка посещает ячейку J_0 , есть множество нерекурсивное. Тогда проблема остановки машины $Z = \{\mathcal{A}', Q, \mathcal{P}\}$, где $\mathcal{A}' = a^* \cup \mathcal{A}$, $a^* \notin \mathcal{A}$, алгоритмически неразрешима, что и требовалось доказать. Таким образом, из того, что для некоторой машины $Z = \{\mathcal{A}, Q, \mathcal{P}\}$ хотя бы одно из множеств $\mathcal{K}_{\text{inf}}^{(r)}$ или $\mathcal{K}_{\text{inf}}^{(l)}$ отлично от рекурсивно перечислимого, следует, что существует машина с программой $\mathcal{P}$, для которой проблема остановки неразрешима.

ОПРЕДЕЛЕНИЕ 4. Пусть $Z = \{\mathcal{A}, Q, \mathcal{P}\}$, где $\mathcal{P} = \{p_1, \dots, p_m\}$. Назовем состояние $q_i \in Q$ начальным, если $\beta_j \neq q_i$ при любом j ($1 \leq j \leq m$). Назовем состояние $q_i \in Q$ заключительным, если $\alpha_i \neq q_i$ при любом j ($1 \leq j \leq m$).

Символ $a_i \in \mathcal{A}$ назовем символом остановки, если $x_j \neq a_i$ при любом j ($1 \leq j \leq m$).

Преобразуем любую машину $Z = \{\mathcal{A}, Q, \mathcal{P}\}$ в машину $Z' = \{\mathcal{A}', Q', \mathcal{P}'\}$ следующим образом. Алфавит Q' получим из алфавита Q , исключив из последнего те символы состояний, которые не встречаются в записи инструкций программы $\mathcal{P}$. Если таких символов нет, то $Q' = Q$.

Если алфавит $\mathcal{A}$ не содержит символов остановки или содержит ровно один такой символ, то $\mathcal{A}' = \mathcal{A}$ и $\mathcal{P}' = \mathcal{P}$. В противном случае, т.е. когда $\mathcal{A}$ содержит более одного символа остановки, получим $\mathcal{A}'$ из алфавита $\mathcal{A}$, исключив из последнего все символы остановки, кроме одного. Если в инструкциях программы $\mathcal{P}$ присутствуют только символы алфавита $\mathcal{A}'$, то $\mathcal{P}' = \mathcal{P}$. В противном случае получим программу $\mathcal{P}'$ из программы $\mathcal{P}$, заменив в каждой ее инструкции символ, не входящий в алфавит $\mathcal{A}'$, на тот единственный символ остановки, который в нем оставлен.

Очевидно, если Z связная машина, то машина Z' также связная, причем программа $\mathcal{P}'$ имеет ту же длину, что и программа $\mathcal{P}$.

Пусть каждая машина Тьюринга преобразована описанным выше способом, и будем далее рассматривать только машины, полученные после преобразования.

Из теоремы 1 получаем два следствия.

СЛЕДСТВИЕ 1. Если $Z = \{\mathcal{A}, Q, \mathcal{P}\}$ – связная машина, то для любого $q_i \in Q$ существуют инструкция $p_j \in \mathcal{P}$, в которой $\alpha_j = q_i$, и инструкция $p_{\mathcal{K}} \in \mathcal{P}$, возможно, совпадающая с p_j , такая, в которой $\beta_{\mathcal{K}} = q_i$.

Другими словами, алфавит Q связной машины не содержит ни начальных, ни заключительных состояний. Действительно, если бы в Q содержалось одно из таких состояний, множество $\mathcal{K}_{(u)}$ было бы пустым.

СЛЕДСТВИЕ 2. Для алфавита $\mathcal{A}$ связной машины $Z = \{\mathcal{A}, Q, \mathcal{P}\}$ выполняется следующее свойство: для любого $a_i \in \mathcal{A}$, отличного от пустого символа и от символа остановки, существует инструкция $p_j \in \mathcal{P}$, в которой $y_j = a_i$.

Действительно, предположим, что для некоторого символа $a_i \in \mathcal{A}$ указанное свойство не выполняется. Тогда любая инструкция p_j , в которой $x_j = a_i$, может быть выполнена в вычислении на любой конфигурации uq_rv не более $|uv|$ раз, а это означало бы, что множество $\mathcal{K}_{(u)}$ пусто. Следовательно, наше предположение неверно.

ОПРЕДЕЛЕНИЕ 5. Назовем машину $Z = \{\mathcal{A}, Q, \mathcal{P}\}$ замкнутой, если выполнены следующие условия:

- 1) множество Q не содержит ни начальных, ни заключительных состояний;
- 2) алфавит $\mathcal{A}$ не содержит символа остановки;
- 3) для любого $a_i \neq 0$ и $a_i \in \mathcal{A}$ существует инструкция p_j , в которой $y_j = a_i$.

ТЕОРЕМА 5. Пусть $Z = \{\mathcal{A}, Q, \mathcal{P}\}$ есть связная машина. Тогда существует замкнутая машина $Z' = \{\mathcal{A}', Q, \mathcal{P}'\}$ такая, что $l(\mathcal{P}') = l(\mathcal{P})$, и для которой по меньшей мере одно из множеств $\mathcal{K}_{(u)}^{(r)}$, $\mathcal{K}_{(u)}^{(l)}$ и $\mathcal{K}_{(u)}^{(2)}$ отлично от рекурсивно перечислимого.

ДОКАЗАТЕЛЬСТВО. Если алфавит $\mathcal{A}$ связной машины $Z = \{\mathcal{A}, Q, \mathcal{P}\}$ не содержит символа остановки, то, применив к ней следствия 1, 2 и следствие из теоремы 2, получаем, что сама машина Z и будет той машиной, существование которой утверждается в теореме 5.

Рассмотрим случай, когда алфавит $\mathcal{A}$ содержит символ остановки, и пусть таким символом является символ a_1 . Заметим, что для связной машины символ a_1 отличен от пустого. Рассмотрим машину $Z' = \{\mathcal{A}', Q, \mathcal{P}'\}$, где $\mathcal{A}' = \mathcal{A} \setminus a_1$, а $\mathcal{P}'$ получена из программы $\mathcal{P}$ заменой в ней каждого символа y_j , равного a_1 , на любой другой символ алфавита $\mathcal{A}$. Если же в любой инструкции p_j символ y_j отличен от a_1 , то $\mathcal{P}' = \mathcal{P}$.

Предположим, что $\mathcal{K}_{(u)}^{(r)}(Z')$ рекурсивно перечислимо, и покажем, что в этом случае рекурсивно перечислимо ее множество $\mathcal{K}_{(u)}^{(r)}(Z)$. Действительно, пусть $g(n)$ общерекурсивная функция, множество значений которой совпадает с множеством всех номеров конфигураций, принадлежащих $\mathcal{K}_{(u)}^{(r)}(Z')$. Если $uv \in (\mathcal{A}')^*$, то $uq_iv \in \mathcal{K}_{(u)}^{(r)}(Z)$ тогда и только тогда, когда $uq_iv \in \mathcal{K}_{(u)}^{(r)}(Z')$. Если $uq_iv = u_1a_1u_2q_iv$, где $u_2v \in (\mathcal{A}')^*$, то $uq_iv \in \mathcal{K}_{(u)}^{(r)}(Z)$ тогда и только тогда, когда $u_2q_iv \in \mathcal{K}_{(u)}^{(r)}(Z')$. В первом случае номер конфигурации uq_iv принадлежит области значений функции $g(n)$, а во втором случае ей принадлежит номер конфигурации u_2q_iv . Тогда будем вычислять значения

функции $g(n)$ для $n = 0, 1, \dots$, пока не получим значение, равное номеру конфигурации uq_iv , если $uv \in (\mathcal{A}')^*$, или номеру конфигурации u_2q_iv , если $uq_iv = u_1a_1u_2q_iv$ и $u_2v \in (\mathcal{A}')^*$. Если слово v содержит символ a_1 , то $uq_iv \notin \mathcal{K}_{(u)}^{(r)}(Z)$. Из вышесказанного следует, что существует частично вычислимая функция $f(n)$, равная единице для всех тех и только тех n , которые являются номерами конфигураций из $\mathcal{K}_{(u)}^{(r)}(Z)$. Следовательно, $\mathcal{K}_{(u)}^{(r)}(Z)$ рекурсивно перечислимое множество. Аналогично получаем, что из того, что рекурсивно перечислимо $\mathcal{K}_{(u)}^{(l)}(Z')(\mathcal{K}_{(u)}^{(2)}(Z'))$, следует, что рекурсивно перечислимо $\mathcal{K}_{(u)}^{(l)}(Z)(\mathcal{K}_{(u)}^{(2)}(Z))$. Следовательно, по меньшей мере, одно из множеств $\mathcal{K}_{(u)}^{(r)}(Z')$, $\mathcal{K}_{(u)}^{(l)}(Z')$ и $\mathcal{K}_{(u)}^{(2)}(Z')$ должно быть отлично от рекурсивно перечислимого, в противном случае приходим к противоречию со следствием из теоремы 2. Теорема 5 доказана.

Мы рассматриваем машины Тьюринга, алфавит входных символов которых содержит не более одного символа остановки.

ОПРЕДЕЛЕНИЕ 6. Назовем машину $Z' = \{\mathcal{A}', Q, \mathcal{P}'\}$ замыканием машины $Z = \{\mathcal{A}, Q, \mathcal{P}\}$, если $Z' = Z$, когда $\mathcal{A}$ не содержит символа остановки, и $Z' = \{\mathcal{A}', Q, \mathcal{P}'\}$, если $\mathcal{A}$ содержит символ остановки a_1 , а для машины Z' алфавит $\mathcal{A}'$ есть $\mathcal{A} \setminus a_1$, а программа $\mathcal{P}'$ получена заменой в программе $\mathcal{P}$ каждого символа u_j , равного a_1 , на один из символов алфавита $\mathcal{A}'$.

Машина Z' , полученная из машины Z при доказательстве теоремы 5, есть замыкание машины Z , и, следовательно, теорему 5 можно сформулировать и следующим образом: для замыкания Z' связной машины Z по меньшей мере одно из множеств $\mathcal{K}_{(u)}^{(r)}(Z')$, $\mathcal{K}_{(u)}^{(l)}(Z')$ или $\mathcal{K}_{(u)}^{(2)}(Z')$ отлично от рекурсивно перечислимого.

Аналогично доказывается

ТЕОРЕМА 6. Для замыкания Z' связной машины Z множество $\mathcal{K}_{(u)}(Z')$ отлично от рекурсивно перечислимого.

Пусть $Z = \{\mathcal{A}, Q, \mathcal{P}\}$, где $Q = \{q_1, \dots, q_s\}$, а $\mathcal{P} = \{p_1, \dots, p_m\}$.

Мы докажем, что справедлива

ТЕОРЕМА 7. Для связной машины Z число s удовлетворяет неравенствам $1 < s \leq m - 2$.

Теорему 7 получим как следствие доказываемой ниже леммы 2, для формулировки которой нам понадобятся следующие определения.

ОПРЕДЕЛЕНИЕ 7. Совокупность из k инструкций программы $\mathcal{P}$ назовем *циклом*, если все инструкции из этой совокупности можно расположить в виде последовательности $p_{i1} \dots p_{ik}$, в которой $\beta_{ij} = \alpha_{i(j+1)}$ для любого j такого, что $1 \leq j \leq k-1$, а $\beta_{ik} = \alpha_{i1}$. Саму последовательность $p_{i1} \dots p_{ik}$ также назовем циклом.

ОПРЕДЕЛЕНИЕ 8. Цикл C программы $\mathcal{P}$ назовем *простым*, если никакое собственное подмножество множества C не есть цикл программы $\mathcal{P}$.

ЛЕММА 2. Если программа $\mathcal{P}$ есть объединение не более чем двух ее простых циклов, то для машины Z с такой программой каждое из множеств $\mathcal{K}_{(u)}^{(r)}(Z)$, $\mathcal{K}_{(u)}^{(l)}(Z)$ и $\mathcal{K}_{(u)}^{(2)}(Z)$ рекурсивно.

ДОКАЗАТЕЛЬСТВО. Вначале предположим, что программа $\mathcal{P}$ представляет собой один простой цикл, и пусть $p_{i1} \dots p_{im}$ – последовательность всех ее инструкций, в которой $\beta_{ij} = \alpha_{i(j+1)}$ для любого $1 \leq j \leq m-1$, а $\beta_{im} = \alpha_{i1}$. Если число левых инструкций в $\mathcal{P}$ равно числу правых инструкций в ней, то любое бесконечное вычисление будет закикливающимся на участке ленты ограниченной длины. Если же число правых (левых) инструкций в $\mathcal{P}$ превосходит число левых (правых) инструкций в ней, то конфигурация $uq_iv \in \mathcal{K}_{(u)}^{(r)}(\mathcal{K}_{(u)}^{(l)})$ тогда и только тогда, когда в вычислении на uq_iv головка выходит в ячейку $J_{|uv|+1}$ и после этого выполнит цикл хотя бы два раза. Множества $\mathcal{K}_{(u)}^{(l)}(\mathcal{K}_{(u)}^{(r)})$ и $\mathcal{K}_{(u)}^{(2)}$ рекурсивны в этом случае.

Пусть теперь $\mathcal{P} = C_1 \cup C_2$, C_1 и C_2 – простые циклы и не существует простого цикла, отличного от C_1 и C_2 . Если $\mathcal{K}_{(u)}$ не пусто, то существуют $p_i \in C_1$ и $p_j \in C_2$ такие, что $\alpha_i = \alpha_j$ и, следовательно, $x_i \neq x_j$. Если не существует простого цикла, отличного от C_1 и отличного от C_2 , то для любой пары инструкций p_k, p_m , отличной от пары p_i, p_j , должно быть выполнено условие $\alpha_k \neq \alpha_m$. Если $\mathcal{K}_{(u)}$ непусто, то в вычислении на любой конфигурации $uq_iv \in \mathcal{K}_{(u)}$ каждая из инструкций p_i и p_j выполнима бесконечное число раз, причем, начиная с инструкции p_i , всегда выполняется цикл C_1 , а начиная с инструкции p_j , всегда выполняется цикл C_2 . Покажем, что при выполнении всех вышеперечисленных условий такое вычисление может быть только периодическим, начиная с некоторого момента, из чего и будет следовать рекурсивность каждого из множеств $\mathcal{K}_{(u)}^{(r)}, \mathcal{K}_{(u)}^{(l)}, \mathcal{K}_{(u)}^{(2)}$.

Если в процессе некоторого вычисления головка выполнила последовательность инструкций $p_{i1} \dots p_{is}$, то после выполнения p_{is} она находится либо в ячейке, которую она посетила при выполнении какой-либо инструкции этой последовательности, либо она находится в ячейке, которую она ни разу не посетила при выполнении последовательности $p_{i1} \dots p_{is}$. Скажем, что в рассматриваемый момент головка находится в первом случае – во внутренней, а во втором – во внешней ячейке для последовательности $p_{i1} \dots p_{is}$.

Если после выполнения каждого из циклов C_1, C_2 головка находится во внутренней для соответствующего цикла ячейке, то, очевидно, любое из рассматриваемых бесконечных вычислений может быть только периодическим, начиная с некоторого момента.

Пусть теперь после выполнения C_1 головка находится во внутренней для C_1 ячейке, а после выполнения C_2 – во внешней для него ячейке. Тогда после выполнения цикла C_1 может быть выполнен только один из циклов C_1, C_2 , и если им является цикл C_1 , то периодичность вычисления очевидна. Пусть после цикла C_1 может быть выполнен только цикл C_2 . Предположим, что после $C_1 C_2$ головка находится во внутренней для $C_1 C_2$ ячейке. Тогда если вычисление не закикливается на участке ленты ограниченной длины, то либо наступит такой момент t , в который головка будет находиться в ячейке, внешней для $C_1 C_2 C$, где $C_1 C_2 C$ – последовательность инструкций, выполненных до момента t , либо такой момент никогда не наступает. Последнее возможно, только если при продолжении вычисления будет выполнен цикл C_1 , что будет означать периодичность вычисления. Если же в некоторый момент t головка находится во внешней для $C_1 C_2 C$

ячейке, то эта ячейка либо находится правее всех внутренних для $C_1 C_2 C$ ячеек, либо она левее этих ячеек. В первом случае в равномерно бесконечном вычислении головка выйдет на пустую ленту справа, а во втором – слева. И в том и в другом случае вычисление станет периодическим, так как из двух инструкций p_i, p_j только одна может быть применима на символе 0.

Останется рассмотреть случай, когда как после выполнения C_1 , так и после выполнения C_2 головка находится во внешней для цикла ячейке. Случай, когда оба цикла правые или оба левые, тривиален. Пусть C_1 – правый цикл, а C_2 – левый, а из инструкций p_i, p_j на символе 0 применима инструкция p_i . Тогда множество $\mathcal{H}_{(u)}^{(r)}$ пусто, так как после выхода на пустую ленту справа может выполняться только цикл C_1 . Отсюда следует, что и $\mathcal{H}_{(u)}^{(2)}$ пусто. Начиная с некоторого k , головка после выполнения последовательности $C_2^k C_1$ всегда находится во внутренней для $C_2^k C_1$ ячейке. Тогда если ни при каком k продолжение вычисления после $C_2^k C_1$ не закликивается на участке ленты, образованном внутренними для $C_2^k C_1$ ячейками, то при любом k в таком вычислении наступит момент, в который головка выйдет в ячейку, отличную от внутренних для $C_2^k C_1$ ячеек. Если при любом k такая ячейка находится справа от внутренних для $C_2^k C_1$ ячеек, то множество $\mathcal{H}_{(u)}^{(l)}$ пустое. Пусть существует минимальное k , равное k_0 , такое, что в продолжении вычисления после $C_2^{k_0} C_1$ первая ячейка, отличная от внутренних для $C_2^{k_0} C_1$, в которую выходит головка, находится слева от них. Тогда любое бесконечное вычисление, в котором перед выходом головки на пустую ленту слева цикл C_2 был выполнен не менее k_0 раз, будет периодическим. Если $uq_i v \in \mathcal{H}_{(u)}^{(l)}$, то в вычислении на $uq_i v$ должен существовать момент t , в который головка выходит на пустую ленту слева, выполнив перед этим цикл C_2 не менее k_0 раз. Следовательно, множество $K_{(u)}^{(l)}$ также рекурсивно.

Случай, когда на символе 0 применима инструкция p_j , аналогичен рассмотренному. Лемма 2 доказана.

Вернемся к теореме 7. Когда для связной машины выполнено равенство $s = t$, ее программа есть простой цикл, а когда $s = t - 1$, ее программа есть объединение не более чем двух простых циклов.

Из теоремы 7 получаем

СЛЕДСТВИЕ 3. *Для любой связной машины Z число инструкций в ее программе не меньше четырех.*

Докажем теперь, что справедлива

ТЕОРЕМА 8. *Любая машина с четырьмя инструкциями в программе отлична от связной.*

ДОКАЗАТЕЛЬСТВО. Вначале сформулируем некоторые утверждения, которые используются при доказательстве теоремы 8 и доказательства которых можно найти, например, в работе [2]. Проблема остановки машины $Z = \{\mathcal{A}, Q, \mathcal{P}\}$ разрешима, если выполнено хотя бы одно из следующих условий:

- 1) $T_i = R(L)$ при любом $1 \leq i \leq t$;
- 2) для любой инструкции $p_i \in \mathcal{P}$ из того, что $T_i = L$, следует, что $x_i \neq 0$, а $y_i = a_k$ (k фиксировано);
- 3) для любой инструкции $p_i \in \mathcal{P}$ из того, что $T_i = L$, следует, что $y_i = 0$;

4) $\mathcal{P}$ содержит единственную левую инструкцию, а $\mathcal{A} = \{0, 1\}$.

Пусть теперь $\mathcal{P} = \{p_1, p_2, p_3, p_4\}$, $s = 2$, причем $Q = \{q_1, q_2\}$. Если программа содержит единственную левую инструкцию и ею является p_1 , то в силу вышесказанного $p_1 = q_1 0 L 1 q_1$. Только в таком случае $\mathcal{P}$ могла бы быть программой связной машины. Для любой машины с единственной левой инструкцией в программе множества $\mathcal{K}_{\text{inf}}^{(l)}$ и $\mathcal{K}^{(2)}$ рекурсивны, а если для такой машины $\mathcal{A} = \{0, 1\}$, то, как показано в [2], и множество $\mathcal{K}_{\text{inf}}^{(r)}$ рекурсивно. Используя этот результат и теорему 6, легко доказать, что $\mathcal{K}_{\text{inf}}^{(r)}$ будет рекурсивным и в случае, когда $\mathcal{A}$ содержит большее число символов, но при этом в $\mathcal{P}$ существует единственная правая инструкция, выполняемая в состоянии q_1 . Остается убедиться в рекурсивности $\mathcal{K}_{\text{inf}}^{(r)}$ в случае, когда $p_2 = q_1 x_2 R y_2 \beta_2$, $p_3 = q_1 x_3 R y_3 \beta_3$, $p_4 = q_2 x_4 R y_4 q_1$. Один из символов x_2, x_3 должен быть равен 1, в противном случае машина не могла бы быть связной. Пусть $x_2 = 1$. Тогда если $\beta_2 = q_1$, а $x_4 \neq 0$, то $\mathcal{K}_{\text{inf}}^{(r)}$ пусто. Если же $x_4 = 0$, то $\mathcal{K}_{\text{inf}}^{(r)}$ должно содержать любую конфигурацию. Следовательно, и в этом случае машина с такой программой не может быть связной. Если $\beta_2 = q_2$, то программа имеет вид: $q_1 0 L 1 q_1, q_1 1 R y_2 q_1, q_1 x_3 R y_3 \beta_3, q_2 1 R y_4 q_1$. Для связной машины тогда должно выполняться: $y_2 \neq 0$ или $y_4 \neq 0$. Отсюда следует, что если перед выходом на пустую ленту справа головка выполнила $p_2 p_4$ или $p_4 p_2$, то все дальнейшее вычисление может происходить только справа от той ячейки, включая ее саму, где она начала выполнять эту последовательность инструкций, откуда следует рекурсивность множества $\mathcal{K}_{\text{inf}}^{(r)}$. Следовательно, и в этом случае программа $\mathcal{P}$ не может быть программой связной машины.

Пусть теперь $\mathcal{P} = \{p_1, p_2, p_3, p_4\}$, $T_1 = T_2 = L, T_3 = T_4 = R, Q = \{q_1, q_2\}$.

Заметим, что если $\{y_3, y_4\} \cap \{x_3, x_4\} = \emptyset$ и $\{y_1, y_2\} \cap \{x_1, x_2\} = \emptyset$, то нерекурсивным может быть только $\mathcal{K}^{(2)}$. Но тогда в программе должны существовать левая инструкция, применимая на символе 0, и правая инструкция, применимая на символе 0, а каждый из символов y_1, y_2, y_3, y_4 должен быть отличен от нуля. Отсюда следует, что связной могла бы быть только программа вида: $q_1 0 L y_1 q_2, q_2 y_4 L y_2 q_2, q_2 0 R y_3 q_1, q_1 y_2 R y_4 q_1$. Легко убедиться в том, что множество $\mathcal{K}^{(2)}$ рекурсивно.

Аналогично доказывается рекурсивность соответствующих множеств и в случае, когда $\{y_3, y_4\} \cap \{x_3, x_4\} \neq \emptyset$ или $\{y_1, y_2\} \cap \{x_1, x_2\} \neq \emptyset$.

Простейшая машина Тьюринга с неразрешимой проблемой остановки есть простейшая из связных машин. Множество связных машин бесконечно. Действительно, любую машину Тьюринга с неразрешимой проблемой остановки можно преобразовать в связную, программа которой есть подпрограмма программы исходной машины. Если бы множество связных машин было конечным, то программа любой машины с неразрешимой проблемой остановки должна была бы содержать подпрограмму, принадлежащую конечному набору программ, что невозможно.

СПИСОК ЦИТИРОВАННОЙ ЛИТЕРАТУРЫ

- [1] Мальцев А. И. Алгоритмы и рекурсивные функции. М.: Наука, 1965.
- [2] Павлоцкая Л. М. Разрешимость проблемы остановки для некоторых классов машин Тьюринга // Матем. заметки. 1973. Т. 13. № 6. С. 899–909.