

УДК 510.522

Александр Евгеньевич Будько

канд. физ.-мат. наук, доц.,

доц. каф. алгебры, геометрии и математического моделирования
Брестского государственного университета имени А. С. Пушкина

Aleksandr Budko

PhD (Physics and Mathematics), Associate Professor,
Associate Professor of Department of Algebra, Geometry, and Mathematical Modelling
at the Brest State A. S. Pushkin University

e-mail: budzko@brsu.brest.by

О ВРЕМЕНИ ОТЫСКАНИЯ КОМАНД В ПРОГРАММАХ ЭЛЕМЕНТАРНЫХ МАШИН ТЬЮРИНГА

Рассматривается одноленточная машина Тьюринга с внешним алфавитом, состоящим из двух символов. Программа машины задается как списком команд, так и ориентированным графом. Определяется, каким образом машина отыскивает очередную команду в своей программе, заданной списком, которую она должна выполнять. Исходя из структуры программы машины, заданной графом, определяются элементарные машины Тьюринга. Для элементарных машин указан порядок следования команд, при котором время их отыскания будет наибольшим, а также определено наибольшее время их отыскания.

Ключевые слова: элементарная машина Тьюринга, команда, программа, порядок следования команд, время отыскания команд.

On the Time of Finding Commands in the Programs of Elementary Turing Machines

A single-tape Turing machine with an external alphabet consisting of two characters is considered. The program of the machine is set both by a list of commands and by an oriented graph. It is determined how the machine finds the next command in its program, specified by the list, which it must execute. Based on the program structure of the machine set by the graph, elementary Turing machines are determined. For elementary machines, the order of following commands is indicated, at which the retrieval time will be the greatest. Also, the longest retrieval time is estimated.

Key words: elementary Turing machine, command, program, order of following commands, command retrieval time.

Введение

Машина Тьюринга является одной из классических моделей алгоритма, позволяющей исследовать свойства самого алгоритма. Самая значительная часть исследований по машинам Тьюринга посвящена рассмотрению сложности тьюринговых вычислений. Обзоры исследований по этому направлению даны в [1–3]. Исследуется также сложность самого алгоритма и, в частности, сложность универсальных машин Тьюринга [4; 5]. Разрабатывается программное обеспечение анимации работы машины [6; 7]. Данная работа посвящена исследованию времени отыскания команд в программах машин Тьюринга и продолжает ранее начатые исследования [8–13].

В настоящей работе рассматриваются машины Тьюринга с одной лентой, одной головкой и внешним алфавитом из двух символов. Программу машины Тьюринга можно задавать как списком команд, так и ориентированным графом. При задании списком программа определяется перечнем команд вида $q_i a_i \rightarrow a_k Dq_j$, где q_i, q_j – внутренние состояния машины, $a_i, a_k \in \{0,1\}$, $D \in \{L, P, C\}$. Состояние q_0 называется конечным и

команда, в правой части которой записано это состояние, – тоже конечной. В [8] определено, каким образом машина отыскивает в списке команду, которую она должна выполнять в данный момент:

1. Пусть в начальный момент головка обозревает символ a_r . Тогда машина, начиная с первой команды списка, сравнивает $q_1 a_r$ с левыми частями команд до тех пор, пока не найдет команду с левой частью $q_1 a_r$, которую она и будет выполнять.

2. Пусть в момент после выполнения команды $q_i a_i \rightarrow a_k Dq_j$ ($q_j \neq q_0$) головка обозревает символ a_c . Тогда, начиная с команды, следующей за выполненной, машина сравнивает $q_j a_c$ с левыми частями команд списка до тех пор, пока не найдет команду с левой частью $q_j a_c$, которую она и будет выполнять.

3. Сравнение производится в том порядке, в котором команды следуют в списке. При этом, если после сравнения с левой частью последней команды списка нужная команда не найдена, сравнение продолжается, начиная с первой команды списка.

4. Сравнение любого $q_j a_k$ с левой частью любой команды списка проводится за одинаковое время: за одну единицу времени.

В [8] определяется оптимальный порядок следования команд в программе как порядок, при котором время их отыскания будет наименьшим. Под наименьшим понимается наименьшее среднее время как отношение суммы времени отыскания по всем начальным конфигурациям к числу этих конфигураций.

В [9] введена классификация $L_0, L_1, L_2, \dots$ машин Тьюринга в зависимости от структуры ориентированного графа, задающего программу машины. Классу L_0 принадлежат машины, программы которых удовлетворяют требованиям:

- 1) граф является связным и не содержит циклов;
- 2) в каждую неконечную вершину, отличную от начальной, входит ровно одна дуга. В начальную вершину не входит ни одна дуга.

Машина Тьюринга класса L_0 с n внутренними состояниями называется элементарной [10], если в ее программе, заданной графом, из каждой вершины i ($1 \leq i \leq n-1$) одна дуга уходит в вершину $i+1$, а другая – в конечную вершину. Из вершины n обе дуги уходят в конечную вершину.

Будем рассматривать только те начальные конфигурации, у которых головка машины за время выполнения программы просматривает все символы начальной конфигурации.

Теорема 1 [10]. Для элементарной машины Тьюринга, имеющей n внутренних состояний, выполняются следующие утверждения:

1) минимальная сумма времени отыскания команд по всем начальным конфигурациям равна $n^2 + 2n$;

2) оптимальным является тот порядок, при котором вначале следуют команды дуг, исходящих из вершины 1, затем – команды дуг, исходящих из вершины 2, и т. д. При этом в каждой паре команд дуг, исходящих из вершины i ($1 \leq i \leq n-1$), первой следует команда дуги, уходящей в конечную вершину.

Следствие 1 [10]. Для элементарной машины Тьюринга минимальное среднее время отыскания команд по всем начальным конфигурациям равно

$$\frac{n^2 + 2n}{n+1} = n + \frac{n}{n+1}.$$

В [10–13] рассматривалась задача нахождения такого порядка следования команд в программах машин Тьюринга, при котором время их отыскания было бы наименьшим. Здесь рассматривается задача отыскания такого порядка следования команд, при котором время их отыскания является наибольшим.

Основная часть

Лемма 1 [12]. При наибольшем времени отыскания команд команды любого пути из начальной вершины в конечную следуют в следующем порядке: первой следует последняя команда пути, второй – предпоследняя команда и т. д. Последней следует первая команда пути.

Лемма 2. Пусть дана элементарная машина M с n внутренними состояниями и пусть команды машины в списке команд следуют в следующем порядке:

– вначале следует пара команд дуг, исходящих из вершины n , затем – пара команд дуг, исходящих из вершины $n - 1$, и т. д.;

– в каждой паре команд дуг, исходящих из вершины i ($1 \leq i \leq n - 1$), первой следует команда дуги, уходящей в вершину $i + 1$.

Тогда время отыскания команд при работе над:

– конфигурацией P_i , которой соответствует путь H_i длины i ($1 \leq i \leq n - 1$) из начальной вершины в конечную, равно $2(ni - i + 1)$;

– конфигурациями P_n и P_{n+1} , которым соответствуют пути H_n и H_{n+1} длины n из начальной вершины в конечную, равно соответственно $2n(n - 1) + 1$ и $2n(n - 1) + 2$.

Доказательство. Пусть условия леммы выполняются. Рассмотрим произвольный путь H_i длины i ($1 \leq i \leq n - 1$) из начальной вершины в конечную. Дуги этого пути в порядке их следования в пути обозначим через $d_1, d_2, d_3, \dots, d_i$. Команды этих дуг при указанном выше порядке их следования находятся на местах $2n - 1, 2n - 3, 2n - 5, \dots, 2n - 2i + 1$ соответственно. Тогда из правила отыскания команд следует, что при работе над конфигурацией P_i ($1 \leq i \leq n - 1$) время отыскания команды дуги:

– d_1 будет равно $2n - 1$ и затем одно сравнение после выполнения команды дуги d_1 , т. е. в сумме $2n$;

– d_2 будет равно $2n - 3$ и затем три сравнения после выполнения команды дуги d_2 , т. е. в сумме $2n$;

– d_3 будет равно $2n - 5$ и затем пять сравнений после выполнения команды дуги d_3 , т. е. в сумме $2n$;

....

– d_{i-1} будет равно $2n - (2i - 1)$ и затем $(2i - 1)$ сравнений после выполнения команды дуги d_{i-1} , т. е. в сумме $2n$;

– d_i будет равно $2n - (2i - 1) + 1 = 2(n - i + 1)$.

Следовательно, время отыскания команд при работе над конфигурацией P_i будет равно $2n(i - 1) + 2(n - i + 1) = 2(ni - i + 1)$.

Аналогично определяется время отыскания команд при работе над конфигурациями P_n и P_{n+1} . Лемма 2 доказана.

Следствие 2. При отыскании команд во время работы над конфигурацией P_i , которой соответствует путь H_i длины i ($i \geq 1$) из начальной вершины в конечную, при порядке их следования, указанном в лемме 2, весь список команд пути полностью просматривается $(i - 1)$ раз и затем просматривается только первая команда списка.

Теорема 2. Для элементарной машины Тьюринга, имеющей n внутренних конечных состояний, выполняются следующие утверждения:

1) наибольшая сумма времени отыскания команд по всем начальным конфигурациям равна $n^3 + 2n^2 - n + 1$;

2) наибольшее время отыскания команд достигается при следующем порядке: вначале следует пара команд дуг, исходящих из вершины n , затем – пара команд дуг, исходящих из вершины $n - 1$, и т. д. При этом в каждой паре команд дуг, исходящих из вершины i ($1 \leq i \leq n - 1$), первой следует команда дуги, уходящей в вершину $i + 1$;

3) других порядков, при которых достигается наибольшее время отыскания команд, кроме указанных в п. 2), нет.

Доказательство. Доказательство теоремы проведем по принципу математической индукции относительно n .

1. $n = 1$. В этом случае программа машины состоит из двух команд и различных порядков следования этих команд будет два. Различных начальных конфигураций будет тоже две: $q_1 0$ и $q_1 1$. Непосредственной проверкой можно убедиться, что для каждого порядка следования команд сумма времени отыскания команд будет равна 3.

2. Пусть для $n = k$ доказываемое утверждение выполняется. Рассмотрим машину M с $n = k + 1$ внутренними состояниями, отличными от состояния q_0 . Этим состояниям соответствуют вершины $1, 2, 3, \dots, k, k + 1$ графа, задающего программу машины M . Машину M можно рассматривать как последовательное соединение двух машин M_1 и M_2 . Машина M_2 состоит из двух команд, которые соответствуют дугам, исходящим из вершины $k + 1$, машина M_1 состоит из оставшихся команд. Другими словами, машина M_1 определяется подграфом с начальной вершиной 1 и имеет k внутренних состояний.

Для машины M различных начальных конфигураций имеется, очевидно, $k + 2$. Эти конфигурации определяются $k + 2$ путями $H_1, H_2, H_3, \dots, H_k, H_{k+1}, H_{k+2}$ из начальной вершины в конечную, длины которых соответственно равны $1, 2, 3, \dots, k, k + 1, k + 1$. Обозначим эти конфигурации соответственно через $P_1, P_2, P_3, \dots, P_k, P_{k+1}, P_{k+2}$. При этом, конфигурации P_{k+1}, P_{k+2} соответствуют машине M_2 , конфигурации $P_1, P_2, P_3, \dots, P_k$ – машине M_1 .

По индуктивному предположению наибольшее время отыскания команд при работе машины M_1 (без учета команд машины M_2) равно $k^3 + 2k^2 - k + 1$. Это значение справедливо для того случая, когда для машины M_1 имеется $k + 1$ начальных конфигураций. Вместе с тем у машины M_1 одна из дуг, выходящих из вершины k , уходит не в конечную вершину, а в вершину $k + 1$. Поэтому для машины M_1 будет не $k + 1$ начальных конфигураций, а k . Соответственно в силу леммы 2 время отыскания команд при работе машины M_1 уменьшится на $(2k(k - 1) + 1)$:

$$k^3 + 2k^2 - k + 1 - (2k(k - 1) + 1) = k^3 + 2k^2 - k + 1 - (2k^2 - 2k + 1) = k^3 + k.$$

При этом по индуктивному предположению наибольшее указанное время отыскания команд $k^3 + k$ будет достигаться только в том случае, если команды машины M_1 (без учета команд машины M_2) будут следовать в порядке, указанном в п. 2) доказываемой теоремы. Поэтому при любом порядке следования команд машины M для достижения наибольшего времени отыскания команд команды машины M_1 должны следовать в порядке, определяемом п. 2) доказываемой теоремы.

Возможны следующие случаи следования команд в программе машины M .

Случай 1. Одна из двух команд машины M_2 в списке команд машины M следует первой. Пусть эта команда соответствует, например, конфигурации P_{k+1} .

Случай 1.1. Вторая команда машины M_2 в списке команд машины M следует второй. Эта команда будет соответствовать конфигурации P_{k+2} .

Определим время отыскания команд машины M_1 с учетом просмотра команд машины M_2 . Машине M_1 соответствует k начальных конфигураций $P_1, P_2, P_3, \dots, P_k$. Эти конфигурации определяются соответственно k путям $H_1, H_2, H_3, \dots, H_k$ из началь-

ной вершины в конечную, длины которых соответственно равны $1, 2, 3, \dots, k$. При работе над каждой из конфигураций $P_1, P_2, P_3, \dots, P_k$ машина M_1 просматривает две команды машины M_2 , находящиеся на первом и втором месте в списке команд машины M . Из следствия 2 следует, что каждая из двух команд машины M_2 при работе над конфигурацией:

- P_1 просматривается по одному разу;
- P_2 просматривается по два раза;
- P_3 просматривается по три раза;
-
- P_k просматривается по k раз.

Поэтому отыскание двух команд машины M_2 дает дополнительно следующее количество сравнений

$$2 \cdot 1 + 2 \cdot 2 + 2 \cdot 3 + \dots + 2 \cdot k = 2 \cdot (1 + 2 + 3 + \dots + k) = 2 \cdot \frac{1}{2} (k + 1) k = k^2 + k.$$

Тогда время отыскания команд машины M_1 с учетом просмотра команд машины M_2 будет равно $k^3 + k + k^2 + k = k^3 + k^2 + 2k$.

Определим теперь время отыскания команд машины M_2 . Машине M_2 соответствуют две начальные конфигурации P_{k+1}, P_{k+2} . Соответствующие им пути имеют длину $k + 1$. Поэтому в силу леммы 2 время отыскания команд при работе над конфигурациями P_{k+1} и P_{k+2} будет соответственно равно

$$2(k + 1)k + 1 \text{ и } 2(k + 1)k + 2$$

Тогда время отыскания команд машины M_2 будет равно

$$2(k + 1)k + 1 + 2(k + 1)k + 2 = 4k^2 + 4k + 3.$$

Таким образом, время отыскания команд машины M будет равно сумме времени отыскания машин M_1 и M_2 :

$$(k^3 + k^2 + 2k) + (4k^2 + 4k + 3) = k^3 + 5k^2 + 6k + 3.$$

Так как $n = k + 1$, то $k^3 + 5k^2 + 6k + 3 = n^3 + 2n^2 - n + 1$ и порядок следования команд в случае 1.1 соответствует п. 2) доказываемой теоремы.

Случай 1.2. Вторая команда машины M_2 в списке команд машины M находится на j -ом месте ($j \geq 3$).

Определим время отыскания команд машины M_1 с учетом просмотра двух команд машины M_2 . Машине M_1 соответствует k начальных конфигураций $P_1, P_2, P_3, \dots, P_k$. При работе над этими конфигурациями машина M_1 просматривает две команды машины M_2 , находящиеся на первом и j -ом местах ($j \geq 3$) в списке команд машины M . По аналогии со случаем 1.1 можно показать, что отыскание первой команды машины M_2 дает дополнительно следующее количество сравнений: $\frac{1}{2}(k^2 + k)$.

Вместе с тем, в отличие от первой команды машины M_2 , вторая ее команда следует уже не до всех команд машины M_1 . До второй команды машины M_2 следует команд $(j - 2)$ машины M_1 , после $-(2k + 2 - j)$ команд. Среди первых $(j - 2)$ команд машины M_1 количество конечных команд равно $\left\lfloor \frac{j-2}{2} \right\rfloor$, и им соответствуют начальные конфигурации $P_1, P_2, P_3, \dots, P_{\left\lfloor \frac{j-2}{2} \right\rfloor}$. Оставшимся $(2k + 2 - j)$ командам машины M_1 соответствуют начальные конфигурации $P_{\left\lfloor \frac{j-2}{2} \right\rfloor+1}, P_{\left\lfloor \frac{j-2}{2} \right\rfloor+2}, \dots, P_k$, при работе над которыми количество сравнений со второй командой будет таким же, как и количество сравнений с первой.

При работе над каждой из $\lfloor \frac{j-2}{2} \rfloor$ начальных конфигураций $P_1, P_2, P_3, \dots, P_{\lfloor \frac{j-2}{2} \rfloor}$ количество сравнений со второй командой машины M_2 будет на единицу меньше, чем количество сравнений с первой командой. Действительно, вторая команда следует после $\lfloor \frac{j-2}{2} \rfloor$ конечных команд, и поэтому в отличие от первой команды машины M_2 вторая перед выполнением конечной команды машины M_1 не просматривается. Поэтому по сравнению с первой командой машины M_2 у второй количество сравнений будет на $\lfloor \frac{j-2}{2} \rfloor$ меньше: $\frac{1}{2}(k^2 + k) - \lfloor \frac{j-2}{2} \rfloor$.

Таким образом, отыскание двух команд машины M_2 дает дополнительно следующее количество сравнений:

$$\frac{1}{2}(k^2 + k) + \frac{1}{2}(k^2 + k) - \lfloor \frac{j-2}{2} \rfloor = (k^2 + k) - \lfloor \frac{j-2}{2} \rfloor.$$

Тогда время отыскания команд машины M_1 с учетом просмотра команд машины M_2 будет равно

$$k^3 + k + k^2 + k - \lfloor \frac{j-2}{2} \rfloor = k^3 + k^2 + 2k - \lfloor \frac{j-2}{2} \rfloor,$$

т. е. по сравнению со случаем 1.1 на $\lfloor \frac{j-2}{2} \rfloor$ сравнений меньше.

Определим теперь время отыскания команд машины M_2 . Машине M_2 соответствуют две начальные конфигурации P_{k+1}, P_{k+2} . Соответствующие им пути H_{k+1}, H_{k+2} из начальной вершины в конечную имеют длину $k + 1$. В случае 1.1 время отыскания команд при работе над конфигурациями P_{k+1} и P_{k+2} было соответственно равно

$$2(k+1)k + 1 \text{ и } 2(k+1)k + 2.$$

В случае 1.2 время отыскания команд при работе над конфигурацией P_{k+1} будет таким же, как и в случае 1.1, поскольку первая команда в случае 1.2 находится на том же месте, что и в случае 1.1: $2(k+1)k + 1$.

В случае 1.2 время отыскания команд при работе над конфигурацией P_{k+2} будет на $2(k+1) + 2 - j = 2k + 4 - j$ меньше, чем в случае 1.1. Действительно, в случае 1.1 вторая команда машины M_2 находилась на втором месте в списке команд, т. е. до всех команд машины M_1 . Поэтому при работе над конфигурацией P_{k+2} время отыскания команд было равно $2(k+1)k + 2$, т. е. при отыскании каждой из k неконечных команд пути H_{k+2} совершалось $2k$ сравнений и при отыскании конечной команды пути $H_{k+2} - 2$ сравнения. В случае 1.2 вторая команда машины M_2 в списке команд машины M находится на j -ом месте, т. е. после $(j-2)$ команд машины M_1 . Поэтому в случае 1.2 при работе над конфигурацией P_{k+2} время отыскания команд будет равно $2(k+1)(k-1) + j$, т. к. при отыскании каждой из $k-1$ неконечных команд пути H_{k+1} совершалось $2k$ сравнений, а при отыскании k -ой неконечной и конечной команд пути $H_{k+2} - j$ сравнений. Следовательно, в случае 1.2 время отыскания команд при работе над конфигурацией P_{k+2} будет на

$$(2(k+1)k + 2) - (2(k+1)(k-1) + j) = 2(k+1) + 2 - j = 2k + 4 - j$$

меньше, чем в случае 1.1.

Таким образом, в случае 1.2 и при работе машины M_1 , и при работе машины M_2 время отыскания команд будет меньше, чем в случае 1.1.

Случай 2. Одна из двух команд машины M_2 в списке команд машины M следует второй. Если вторая команда машины M_2 в списке команд машины M следует первой, то этот случай совпадает со случаем 1.1.

Пусть вторая команда машины M_2 в списке команд машины находится на j -ом месте ($j \geq 3$). Тогда на первом месте будет находиться одна из команд машины M_1 .

Случай 2.1. Команда машины M_1 , находящаяся на первом месте в списке команд машины M , является неконечной. Другими словами, это будет команда дуги, следующей из вершины i ($1 \leq i \leq k$) в вершину $i + 1$. Эта дуга принадлежит пути H_{k+1} из начальной вершины в конечную, где последняя дуга определяет первую команду машины M_2 . В силу леммы 1 при наибольшем времени отыскания первая команда машины M_2 должна находиться до команды дуги, следующей из вершины i ($1 \leq i \leq k$) в вершину $i + 1$. В случае 2.1 это требование не выполняется. Поэтому в случае 2.1 наибольшее время отыскания команд достигаться не будет.

Случай 2.2. Команда машины M_1 , следующая первой в списке команд машины M , является конечной. Другими словами, это будет команда дуги, следующей из вершины i ($1 \leq i \leq k$) в конечную вершину. Обозначим эту дугу через d . По индуктивному предположению при наибольшем времени отыскания команд дугой d должна быть дуга, исходящая из вершины $i = k$.

Дуга d является последней в пути H_k из начальной вершины в конечную. Длина пути H_k равна k , и этот путь соответствует начальной конфигурации P_k .

Сравним время отыскания команд в случаях 1.1 и 2.2. Для этого определим время отыскания команд машины M_1 с учетом просмотра команд машины M_2 . Машине M_1 соответствует k начальных конфигураций $P_1, P_2, P_3, \dots, P_k$. При работе над этими k конфигурациями машина M_1 просматривает две команды машины M_2 , находящиеся на первом и j -ом местах ($j \geq 3$) в списке команд машины M . По аналогии со случаем 1.1 можно показать, что отыскание первой команды машины M_2 дает дополнительно такое же количество сравнений, как и в случае 1.1: $\frac{1}{2}(k^2 + k)$.

В случае 1.1 команда дуги d в списке команд машины M находилась на четвертом месте. Поэтому в случае 1.1 время отыскания команд при работе над конфигурацией P_k было равно $2(k + 1)(k - 1) + 4$. В случае 2.2 время отыскания команд при работе над конфигурацией P_k будет равно $2(k + 1)(k - 1) + 1$, поскольку команда дуги d в списке команд находится на первом месте. Таким образом, при работе над конфигурацией P_k время отыскания команд в случае 2.2 уменьшается на 3 единицы по сравнению со случаем 1.1.

При работе над конфигурациями $P_1, P_2, \dots, P_{k-1}$ время отыскания команд в случае 2.2 будет, очевидно, таким же, как и в случае 1.1. Следовательно, в случае 2.2 по сравнению со случаем 1.1 при работе машины M_1 время отыскания команд будет меньше на 3 единицы.

Определим теперь время отыскания команд машины M_2 . Машине M_2 соответствуют две начальные конфигурации P_{k+1}, P_{k+2} . Соответствующие им пути имеют длину $k + 1$. В случае 1.1 время отыскания команд при работе над конфигурациями P_{k+1} и P_{k+2} было соответственно равно

$$2(k + 1)k + 1 \text{ и } 2(k + 1)k + 2.$$

В случае 2.2 время отыскания команд при работе над конфигурацией P_{k+2} будет таким же, как и в случае 1.1, поскольку первая команда машины M_2 в случае 2.2 находится на том же втором месте, что и в случае 1.1: $2(k + 1)k + 2$.

В случае 2.2 время отыскания команд при работе над конфигурацией P_{k+1} будет равно:

а) $2(k+1)k+3$, если $j=3$. Действительно, в отличие от случая 1.1 вторая команда машины M_2 находится не на первом месте, а на третьем. А так как в случае 1.1 время отыскания команд было равно $2(k+1)k+1$, то в случае 2.2 время отыскания при работе машины M_2 будет на 2 единицы больше, чем в случае 1.1.

Итак, в случае 2.2 по сравнению со случаем 1.1 время отыскания команд при работе машины M_1 уменьшилось на 3 единицы, а при работе машины M_2 увеличилось на 2. Разность этих величин равна 1, т. е. общее время работы машины M в случае 2.2 уменьшилось на 1 единицу по сравнению со случаем 1.1.

б) $2(k+1)(k-1)+j$, если $j>3$. Это значит, в случае 2.2 при работе машины M_2 время отыскания команд будет на

$$(2(k+1)k+1) - (2(k+1)(k-1)+j) = 2(k+1) + 1 - j = 2k + 3 - j$$

меньше, чем в случае 1.1.

Итак, в случае 2.2 по сравнению со случаем 1.1 время отыскания команд и при работе машины M_1 , и при работе машины M_2 уменьшилось.

Случай 3. Одна из двух команд машины M_2 в списке команд машины M находится на j -ом месте ($j \geq 3$). Если вторая команда машины M_2 в списке команд машины M следует первой или второй, то этот случай совпадает соответственно со случаем 1.2 и случаем 2.

Пусть вторая команда машины M_2 в списке команд машины M находится на s -ом месте ($s \geq 3$). Пусть для определенности $s > j$. Тогда $s > 3$.

Из условия рассматриваемого случая следует, что в списке команд машины M на первом и втором местах находятся команды машины M_1 . По индуктивному предположению это будут команды дуг, выходящих из вершины k .

Случай 3.1. На первом и втором местах находятся команды дуг, исходящих из вершины k и входящих соответственно в вершину $k+1$ и конечную вершину.

По аналогии со случаем 1.2 можно показать, что в случае 3.1 время отыскания команд при работе:

– машины M_1 будет на $\lfloor \frac{j-2}{2} \rfloor + \lfloor \frac{s-2}{2} \rfloor$ меньше, чем в случае 1.1;

– машины M_2 будет на $(2k+2-j) + (2k+2-s)$ меньше, чем в случае 1.1.

Таким образом, в случае 3.1 и при работе машины M_1 , и при работе машины M_2 время отыскания команд будет меньше, чем в случае 1.1.

Случай 3.2. На первом и втором местах находятся команды дуг, исходящих из вершины k и входящих соответственно в конечную вершину и в вершину $k+1$. Аналогично предыдущему случаю показывается, что в случае 3.2 и при работе машины M_1 , и при работе машины M_2 время отыскания команд будет меньше, чем в случае 1.1.

Теорема 2 доказана.

Следствие 3. Для элементарной машины Тьюринга наибольшее среднее время отыскания команд по всем начальным конфигурациям равно

$$\frac{n^3 + 2n^2 - n + 1}{n+1} = n^2 + n - \frac{2n-1}{n+1}.$$

Следствие 4. Для элементарной машины Тьюринга время отыскания команд по всем начальным конфигурациям T удовлетворяет условию

$$n^2 + 2n \leq T \leq n^3 + 2n^2 - n + 1.$$

Следствие 5. Для элементарной машины Тьюринга среднее время отыскания команд по всем начальным конфигурациям удовлетворяет условию

$$n + \frac{n}{n+1} \leq T_{\text{ср.}} \leq n^2 + n - \frac{2n-1}{n+1}.$$

Заклучение

Таким образом, для элементарных машин Тьюринга определен порядок следования команд, при котором время их отыскания будет наибольшим. Кроме того, для элементарных машин указана наибольшая сумма времени отыскания команд по всем начальным конфигурациям. С учетом результатов, полученных в [10], указаны верхняя и нижняя границы времени отыскания команд для элементарных машин Тьюринга по всем начальным конфигурациям, а также такие же границы для среднего времени.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Марченков, С. С. Сложность алгоритмов и вычислений / С. С. Марченков, В. Л. Матросов // Итоги науки и техники. Сер.: Теория вероятностей. Мат. статистика. Теорет. кибернетика. – 1979. – Т. 16. – С. 103–149.
2. Слисенко, О. А. Сложностные задачи теории вычислений / О. А. Слисенко // Успехи мат. наук. – 1981. – Т. 36, вып. 6 (222). – С. 21–103.
3. Крупский, В. Н. Введение в сложность вычислений / В. Н. Крупский. – М. : Факториал Пресс, 2006. – 217 с.
4. Минский, М. Вычисления и автоматы / М. Минский. – М. : Мир, 1971. – 281 с.
5. Рогожин, Ю. В. Семь универсальных машин Тьюринга / Ю. В. Рогожин // Мат. исслед. – 1982. – № 6. – С. 76–90.
6. Ясюкович, Э. И. Разработка алгоритма и программного обеспечения детерминированной машины Тьюринга / Э. И. Ясюкович // Весн. МДУ імя А. А. Куляшова. – 2017. – № 2. – С. 13–22.
7. Мозговой, М. В. Классика программирования: алгоритмы, языки, автоматы, компиляторы. Практический подход / М. В. Мозговой. – СПб. : Наука и техника, 2006. – 320 с.
8. Будько, А. Е. О некоторых свойствах оптимального порядка следования команд в программах машин Тьюринга / А. Е. Будько // Нанопроектирование, технология, компьютерное моделирование (НПТКМ-2019) : материалы 18-го Междунар. симпозиума по новым приложениям высоких технологий, посвящ. памяти В. А. Пальмова, Брест, 24–27 сент. 2019 г. – Брест : БрГУ, 2019. – С. 19–20.
9. Будько, А. Е. О двух классах машин Тьюринга / А. Е. Будько // Докл. АН БССР. – 1985. – № 9. – С. 792–793.
10. Будько, А. Е. О порядке следования команд в программах элементарных машин Тьюринга / А. Е. Будько // Вучон. зап. Брэсц. дзярж. ун-та імя А. С. Пушкіна. – 2020. – Вып. 16, ч. 2. – С. 7–12.
11. Будько, А. Е. О порядке следования команд в программах полных древо-видных машин Тьюринга / А. Е. Будько // Весн. МДУ імя А. А. Куляшова. Сер. В. – 2021. – № 1. – С. 19–26.
12. Будько, А. Е. Об одном свойстве следования команд в программах машин Тьюринга / А. Е. Будько, А. В. Богдан // Вычислительные методы, модели и образовательные технологии : сб. материалов Респ. науч.-практ. конф., Брест, 22 окт. 2021 г. / Брест. гос. ун-т им. А. С. Пушкина ; под общ. ред. Д. В. Грицука. – Брест : БрГУ, 2021. – С. 81–82.
13. Будько, А. Е. О порядке следования команд в программах одноцикловых машин Тьюринга / А. Е. Будько // Весн. МДУ імя А. А. Куляшова. Сер. В. – 2022. – № 1. – С. 14–22.

REFERENCES

1. Marchienkov, S. S. Slozhnost' algoritmov i vychislenij / S. S. Marchenkov, V. L. Matrosov // Itogi nauki i tiekhniki. Sier.: Tieorija vierojatnostiej. Mat. stat. Tieoriet. kibernetika. – 1979. – T. 16. – S. 103–149.
2. Slisienko, O. A. Slozhnostnyje zadachi tieorii vychislenij / O. A. Slisenko // Uspekhi mat. nauk. – 1981. – T. 36, vyp. 6 (222). – S. 21–103.
3. Krupskij, V. N. Vviedienie v slozhnost' vychislenij / V. N. Krupskij. – M. : Factorial Press, 2006. – 217 s.
4. Minskij, M. Vychislenija i avtomaty / M. Minskij. – M. : Mir, 1971. – 281 s.
5. Rogozhin, Yu. V. Siem' univiersal'nykh mashin T'juringa / Yu. V. Rogozhin // Mat. isslied. – 1982. – № 6. – S. 76–90.
6. Yasiukovich, E. I. Razrabotka algoritma i programmogo obiespiechienija determinirovannoj mashiny T'juringa / E. I. Yasiukovich // Viesn. MDU imia A. A. Kuliashova. – 2017. – № 2. – S. 13–22.
7. Mozgovoj, M. V. Klassika programmirovaniya: algoritmy, jazyki, avtomaty, kompiliatory. Prakticheskij podhod / M. V. Mozgovoj. – SPb. : Nauka i Tekhnika, 2006. – 320 s.
8. Bud'ko, A. Ye. O niekotorykh svojstvakh optimal'nogo poriadka sledovaniya komand v programmakh mashin T'juringa / A. Ye. Bud'ko // Nanoprojektirovanije, tiekhnologija, komp'juternoje modelirovanije (NPTKM – 2019) : materialy 18-go miezhdunar. simpoziuma po novym prilozhenijam vysokikh tiekhnologij, posviashch. pamiaty V. A. Pal'mova, Briest, sientiabr' 2019. – Briest : BrGU, 2019. – S. 16–18.
9. Bud'ko, A. Ye. O dvukh klassakh mashin T'juringa / A. Ye. Bud'ko // Dokl. AN BSSR. – 1985. – № 9. – S. 792–793.
10. Bud'ko, A. Ye. O poriadkie sledovaniya komand v programmakh eliemientarnykh mashin T'juringa / A. Ye. Bud'ko // Vuchon. zap. Bresc. dziarzh. un-ta imia A. S. Pushkina. – 2020. – Vyp. 16, ch. 2. – S. 7–12.
11. Bud'ko, A. Ye. O poriadkie sledovaniya komand v programmakh polnykh drievo-vidnykh mashin T'juringa / A. Ye. Bud'ko // Viesn. MDU imia A. A. Kuliashova. Sier. B. – 2021. – № 1. – S. 19–26.
12. Bud'ko, A. Ye. Ob odnom svojstvie sledovaniya komand v programmakh mashin T'juringa / A. Ye. Bud'ko, A. V. Bogdan // Vychislitel'nyje mietody, modeli i obrazovatel'nyje tiekhnologii : sb. materialov Riesz. nauch.-pract. konf., Briest, 22 okt. 2021 g. / Briest. gos. un-t im. A. S. Pushkina ; pod obshch. ried. D. V. Gricuka. – Briest: BrGU, 2021. – S. 80–81.
13. Bud'ko, A. Ye. O poriadkie sledovaniya komand v programmakh odnociklovykh mashin T'juringa / A. Ye. Bud'ko // Viesn. MDU imia A. A. Kuliashova. Sier. B. – 2022. – № 1. – S. 14–22.

Рукапіс паступіў у рэдакцыю 20.04.2022