Доклады Академии наук СССР
1969. Том 189, № 4

УДК 519.95

КИБЕРНЕТИКА И ТЕОРИЯ РЕГУЛИРОВАНИЯ

М. И. ДЕХТЯРЬ

О НЕВОЗМОЖНОСТИ ЭЛИМИНАЦИИ ПОЛНОГО ПЕРЕБОРА ПРИ ВЫЧИСЛЕНИИ ФУНКЦИЙ ОТНОСИТЕЛЬНО ИХ ГРАФИКОВ

(Представлено академиком П. С. Новиковым 21 IV 1969)

В работах С. В. Яблонского (1) и Ю. И. Журавлева (2) рассматриваются некоторые специальные задачи, при решении которых предположительно невозможно обойтись без сложных алгоритмов типа «перебор всех вариантов». Обоснование подобных предположений проводится ими в терминах «правильных алгоритмов» (1) и «локальных алгоритмов конечного ранга» (2). В работе (3) Б. А. Трахтенброт рассмотрел другой подход к делу, основанный на оценке сложности вычислений посредством сигнализирующих функций, и в этих терминах анализировал некоторые аспекты задачи С. В. Яблонского. Этот же подход применяется в настоящей статье для установления неэлиминируемости перебора при решении другой модельной задачи, а именно: задачи вычисления функции относительно ее графика. Предварительная ее формулировка (точные определения см. в пп. 1^0, 2^0) заключается в следующем. Пусть A задумал некоторую функцию (быть может, неэффективную), значения которой B пытается угадать; при этом ему разрешается задавать вопросы вида «$f(y) = z$?», на которые A обязан давать правильные ответы («да» или «нет»). Стратегия «полного перебора» для B заключается в том, что он для каждого x последовательно спрашивает «$f(x) = 0$?», «$f(x) = 1$?» и т. д. до получения положительного ответа. В каких случаях и в каком смысле невозможна лучшая стратегия (т. е. перебор не элиминируем)?

Ясно, что оценивать сложность стратегии только числом задаваемых вопросов не разумно, ибо при этом не учитывается сложность самих вопросов (можно было бы, например, для эффективной функции f ограничиться всегда одним вопросом «$f(x) = y$?», где y — заранее вычисленное значение функции в точке x).

В пп. 1^0, 2^0 приведены точные определения понятий «стратегия угадывания», «полный перебор» и их сложности в терминах вычислений на машинах Тьюринга с «оракулом» и сложности вычислений. В пп. 3^0, 4^0 рассматриваются две ситуации (теоремы 1, 2), подтверждающие гипотезу Б. А. Трахтенброта о неэлиминируемости полного перебора.

1^0. Для вычисления функций будем использовать специальные машины Тьюринга. Внешний алфавит этих машин A удобно рассматривать как произведение двух алфавитов: $A = A_1 \times A_2$, где $A_1 = \{\wedge, 1, 0\}$ является алфавитом «нижнего этажа» ленты, а некоторый конечный алфавит A_2 является алфавитом «верхнего этажа» ленты. Каждая машина имеет конечное множество состояний $Q = \{q_0, q_1, \ldots, q_s\}$, в котором выделено подмножество $\tilde{Q} = \{q_{i_1}, q_{i_2}, \ldots, q_{i_r}\}$ $(r < s)$ вопросительных состояний ($\tilde{Q}$ может быть пусто). Для каждого вопросительного состояния $q_\alpha \in \tilde{Q}$ в программе машины имеются специальные команды вида

$$q_\alpha a \to q_i,\ q_j \quad (a \in A;\ q_i,\ q_j \in Q). \tag{1}$$

При вычислении функции f на некоторой машине $\mathfrak{M}$ к машине присоединяется «f-оракул», который «знает» график функции и для любой пары чисел (z, y) «умеет» отвечать на вопрос: равна ли функция f в точке y числу z («$f(y) = z$?»). Работа машины $\mathfrak{M}$ проходит следующим образом: в начальный момент на нижнем этаже ленты записан аргумент x (в двоичной системе); головка машины находится в ячейке с записью младшего разряда x в состоянии q_1. Затем машина работает в соответствии со своей программой до тех пор, пока впервые не перейдет в некоторое вопросительное состояние q_α. Выполнение команды вида (1) зависит от f-оракула. Предположим, что к этому моменту запись на нижнем этаже ленты имеет вид

$$N \wedge z \wedge y \wedge M, \tag{2}$$

где N и M — произвольные слова в алфавите A_1; z и y — двоичные числа, причем головка машины обозревает одну из ячеек с записью z. Такую ситуацию f-оракул воспринимает как вопрос «$f(y) = z$?». Ответ на вопрос машины дается в следующей форме: a) если $f(y) = z$, то f-оракул переводит машину в состояние q_i; b) если $f(y) \neq z$, то машина переводится в состояние q_j. Если же к моменту выполнения команды (1) запись на ленте не имеет вид (2), то машина остается в состоянии q_α и в дальнейшем никаких действий не совершает.

Получив ответ f-оракула на первый вопрос, машина $\mathfrak{M}$ продолжает работать согласно своей программе. В процессе работы она вновь может обращаться к f-оракулу с вопросами вида «$f(y) = z$?» и получает ответ в указанной выше форме.

Пусть $f(x)$ — произвольная функция. Обозначим через D_f область ее определения.

Определение. Машина $\mathfrak{M}$ вычисляет функцию $f(x)$ относительно графика функции $f(x)$, если, начав работу с f-оракулом в начальной конфигурации, она: 1) для любого $x \in D_f$ через конечное число тактов придет в значительное состояние q_0, и запись на нижнем этаже ленты в этот момент имеет вид

$$N \wedge z \wedge M, \tag{3}$$

где N, M — произвольные слова в алфавите A_1; z — двоичное число, причем $z = f(x)$ и головка машины находится в одной из ячеек с записью z; 2) для любого $x \notin D_f$ либо никогда не остановится, либо заключительная конфигурация не имеет вид (3).

Через $t_{\mathfrak{M}f}(x)$ обозначим число таков работы $\mathfrak{M}$ при вычислении f в точке x. Если в процессе работы с f-оракулом в точке x машина никогда не перейдет в заключительную конфигурацию (3), где $z = f(x)$, то полагаем по определению $t_{\mathfrak{M}f}(x) = \infty$.

В дальнейшем будем говорить, что машина $\mathfrak{M}$ вычисляет функцию $f(x)$, если $\mathfrak{M}$ вычисляет $f(x)$ относительно графика $f(x)$.

2^0. Существует много машин, реализующих полный перебор. В этом пункте мы зафиксируем одну из них $\mathfrak{M}_0$, которая является самой «быстрой» (см. лемму 2) среди таких машин. Алфавит A_2 верхнего этажа ленты состоит у $\mathfrak{M}_0$ из единственного символа $*$. Начав работу с аргументом x на ленте, головка $\mathfrak{M}_0$ проходит запись x справа налево, оставляет одну ячейку пустой, а в следующей печатает $\overset{*}{0}$ и переходит в вопросительное состояние. f-Оракулу задается вопрос: «$f(x) = 0$?». При отрицательном ответе 0 заменяется на 1 и задается вопрос «$f(x) = 1$?» и т. д. Символ $*$ на верхнем этаже отмечает ячейку с младшим разрядом вопроса. Для любого $n > 0$ вопрос «$f(x) = n$?» задается машиной $\mathfrak{M}_0$, когда ее головка находится в ячейке с записью крайней справа единицы в двоичном представлении числа n. Если n нечетно, то это будет отмеченная ячейка. Тогда для перехода к вопросу «$f(x) = (n+1)$?» головка $\mathfrak{M}_0$ движется справа налево, заменяя все единицы на нули до тех пор, пока впервые не попадет в ячейку, не содержащую 1. В этой ячейке записывается 1, и машина переходит в вопросительное состояние. Задается вопрос «$f(x) = (n+1)$?».

Если n четно, то ячейка, в которой находится головка $\mathfrak{M}_0$ в момент вопроса «$f(x) = n$?», не содержит $*$. Тогда для записи следующего вопроса головка сдвигается вправо до отмеченной ячейки, заменяет в ней 0 на 1 и задает вопрос «$f(x) = (n+1)$?». Машина $\mathfrak{M}_0$ останавливается, когда получает положительный ответ f-оракула на свой вопрос. Легко оценить время работы машины $\mathfrak{M}_0$.

Лемма 1. *Пусть f — произвольная функция, определенная в точке x. Тогда* $t_{\mathfrak{M}_0 f}(x) < 8[f(x)+1] + \log_2 x$.

Пусть f — произвольная функция, $\mathfrak{M}$ — одна из машин. Предположим, что при вычислении f в некоторой точке x машина $\mathfrak{M}$ реализует полный перебор, т. е. задает последовательно вопросы «$f(x) = 0$?», «$f(x) = 1$?» и т. д. Следующая лемма показывает, что $\mathfrak{M}$ реализует перебор не быстрее, чем машина $\mathfrak{M}_0$.

Лемма 2. *Имеет место одно из следующих двух утверждений:*

1) *машина $\mathfrak{M}$ при вычислении f в точке x задает все вопросы за то же время, что машина $\mathfrak{M}_0$ при вычислении $f(x)$;*

2) *существует такое число n, что вопрос «$f(x) = n$?» задается машиной $\mathfrak{M}_0$ раньше, чем машиной $\mathfrak{M}$, а все вопросы «$f(x) = i$?» ($i = 0, 1, \ldots$ $\ldots, n-1$) задаются машинами $\mathfrak{M}_0$ и $\mathfrak{M}$ одновременно.*

Вообще говоря, есть много машин, работа которых во всех или почти во всех точках совпадает с работой машины $\mathfrak{M}_0$. Естественно не отличать такие машины от $\mathfrak{M}_0$. Машину $\mathfrak{N}$ назовем эквивалентной машине $\mathfrak{M}_0$, если во всех точках, за исключением, быть может, конечного числа, $\mathfrak{N}$ реализует полный перебор за то же время, что и машина $\mathfrak{M}_0$ ($\mathfrak{N} \sim \mathfrak{M}_0$).

3⁰. Для достаточно широких классов функций существуют машины, вычисляющие их во всех точках быстрее машины $\mathfrak{M}_0$. Такими являются, например, все монотонные функции, функции с «редким», легко перечислимым множеством значений, функции, вычислимые на обычных машинах Тьюринга (без f-оракулов) с временной сигнализирующей $t_f(x) <$ $< f(x)$. К последнему классу относятся, в частности, все многочлены. Следующее утверждение показывает, что имеются и такие функции, которые нельзя во всех точках вычислить быстрее полного перебора.

Теорема 1. *Существует общерекурсивная функция (о.р.ф.) $f(x)$ такая, что для любой машины $\mathfrak{N}$, $\mathfrak{N} \not\sim \mathfrak{M}_0$, имеет место неравенство $t_{\mathfrak{N}f}(x) >$ $> t_{\mathfrak{M}_0 f}(x)$ в бесконечном числе точек x.*

Доказательство этой теоремы проводится диагональными методами и использует лемму 2.

Примечание 1. Из любой о.р.ф. $g(x)$, изменяя ее на «редком» множестве, можно получить функцию $f(x)$, удовлетворяющую утверждению теоремы 1. Обозначим через $\lambda fg(y)$ число тех значений аргумента x, $x \leqslant y$, для которых $f(x)$ отлична от $g(x)$. Пусть $\varepsilon(y)$ — произвольная неубывающая, неограниченная о.р.ф. Тогда для данной о.р.ф. $g(x)$ существует о.р.ф. $f(x)$, удовлетворяющая теореме 1 и такая, что $\lambda fg(y) \leqslant \varepsilon(y)$ для всех y.

Примечание 2. Теорему 1 нельзя усилить, так как для каждой функции $f(x)$ существует машина $\mathfrak{M}_f$, которая в бесконечном числе точек x вычисляет $f(x)$ быстрее машины $\mathfrak{M}_0$. Разумеется, если $f(x)$ удовлетворяет теореме 1, то в другом бесконечном множестве точек машина $\mathfrak{M}_0$ вычисляет $f(x)$ быстрее машины $\mathfrak{M}_f$.

4⁰. В этом пункте будем рассматривать время вычисления функций с точностью до постоянных множителей (по порядку). Из леммы 1 следует, что на вычисление достаточно «больших» функций f полный перебор тратит по порядку $f(x)$ тактов. Существуют функции, для которых эта оценка является наилучшей.

Теорема 2. *Для произвольной о.р.ф. $g(x)$ существует о.р.ф. $f(x)$ такая, что:*

1) $\forall x[f(x) > g(x)]$;

2) *всякая машина $\mathfrak{N}$ вычисляет $f(x)$ не быстрее (по порядку) машины $\mathfrak{M}_0$*, т. е.

$$\forall \mathfrak{N}\ \exists\, c(\mathfrak{N})\ \forall x [t_{\mathfrak{N}f}(x) \geqslant c(\mathfrak{N}) \cdot f(x)],$$

где $c(\mathfrak{N})$ — константа, зависящая лишь от машины $\mathfrak{N}$.

Утверждение теоремы 2 можно уточнить, если ввести в рассмотрение число вопросов, задаваемых машиной $\mathfrak{N}$ при вычислении функции f. Существуют такие функции, что если при их вычислении $\mathfrak{N}$ задает по порядку меньше вопросов, чем машина $\mathfrak{M}_0$, то время, затраченное машиной $\mathfrak{N}$, будет значительно больше времени машины $\mathfrak{M}_0$.

Теорема 2'. *Пусть $h(x, y)$ — произвольная о.р.ф., не убывающая по y. Тогда для любой о.р.ф. $g(x)$ существует о.р.ф. $f(x)$, $f(x) > g(x)$ во всех точках, со следующим свойством: для всякой машины $\mathfrak{N}$ существует константа $\bar{c}(\mathfrak{N})$, такая, что в произвольной точке x имеет место одно из двух утверждений:*

1) *машина $\mathfrak{N}$ при вычислении f в точке x задает f-оракулу не менее $\bar{c}(\mathfrak{N}) \cdot f(x)$ вопросов;*

2) $t_{\mathfrak{N}f}(x) \geqslant h(x, f(x))$.

Ясно, что при $h(x, y) \geqslant y$ функции, удовлетворяющие теореме 2', удовлетворяют также теореме 2.

В заключение автор благодарит Б. А. Трахтенброта за постановку задачи и полезные замечания.

Новосибирский
государственный университет

Поступило
17 IV 1969

ЦИТИРОВАННАЯ ЛИТЕРАТУРА

[1] С. В. Яблонский, Сборн. Проблемы кибернетики, 2, 75 (1959). [2] Ю. И. Журавлев, Там же, 8, 5 (1962). [3] Б. А. Трахтенброт, Алгебра и логика, Семинар, 4, 5, 79 (1965).