УДК 517.11:518.5

В.А. Ганов, В.Р. Карымов

Ограниченные вычисления с оракулами

Исследуются вычисления на машинах с оракулами, работающие с ограничениями на время работы. Главная особенность машин с оракулами – наличие спрашивающих команд, выполнение которых означает нахождение ответа на заданный вопрос. В общем случае вопросы вычисляет машина механически согласно своей программе, а ответ дается машине извне, как значение некоторой нерекурсивной процедуры или функции, называемой *оракулом*. При этом вопросы могут содержать программы других таких же машин, а ответы связаны с поведением этих машин. В результате возникает язык программирования, в котором можно описывать некоторые неалгоритмические процессы.

Например, в работе [1, с. 36] построен специальный оракул F_0, с которым вычислимы гиперарифметические функции, не являющиеся рекурсивными. В частности, оракул F_0 решает так называемую проблему остановки для машин, хорошо работающих с ним же. Известно, что подобная проблема не разрешима для рекурсивных вычислений. Тем не менее создатель общей теории вычислений с оракулами Н.В. Белякин предложил ввести дополнительные ограничения в работе машин с оракулами так, чтобы полученные вычисления можно было бы осуществлять на современных компьютерах. И данную работу можно считать первой попыткой решить эту задачу.

1. Машины Тьюринга весьма далеки от современных вычислительных машин, и потому их использование вносит дополнительные трудности в решении поставленной задачи. Поэтому здесь рассматривается специальная модификация машин Шенфилда из [2]. Такие машины имеют бесконечное число регистров, управляющее устройство и программу. Регистры пронумерованы натуральными числами, и каждый из них может содержать натуральное число, букву или некоторый символ. Пусть x_n обозначает содержимое регистра с номером n. В регистрах 0,1 расположены счетчик команд и счетчик тактов; затем выделяются регистры: для записи программы рассматриваемой машины и ее аргументов; для записи вопросов и ответов машины; для записи результата вычисления. Остальные регистры в начальный момент считаются пустыми.

Управляющее устройство читает содержимое регистров и преобразует его согласно выполняемой команде. Один такт работы машины заключается в исполнении команды, содержащейся в регистре x_0, при этом указывается код следующей выполняемой команды. Количество тактов работы машины ограничено числом t, указанным в счетчике тактов, и при выполнении каждой команды число t уменьшается на 1. При $t = 0$ работа машины прекращается. Тогда, если в счетчике команд указан код *заключительной команды* q_0, то работа машины определена и ее результат записывается в указанных выше регистрах; в противном случае считается, что работа машины не определена и результатом работы получается символ $\varnothing$.

Оракул является некоторым оператором $F(y_1, \dots , y_к)$, значениями аргументов которого могут быть числа и программы машин. Вопрос к оракулу имеет вид числового кортежа ($m_1, \dots , m_k$), составленного из номеров регистров, содержащих значения аргументов оракула. Ответ представляет значение $F(x_{m_1}, \dots , x_{m_k})$, если оно определено. Оракул не входит в программу рассматриваемой машины, и можно считать, что он присоединен к ее специальному входу.

Кроме заключительной команды q_0 рассматриваются две *оперативные команды* $q_1(i, n)$, $q_2(i, n, m)$ и одна *спрашивающая команда* $q_3(n, \bar{u})$, в которых i, n, m – натуральные числа, рассматриваемые как номера регистров; $\bar{u} = (m_1, \dots , m_k)$ – кортеж номеров регистров, содержащих значения аргументов оракула F. Случай $x_1 = 0$ описан выше, и пусть $x_1 > 0$. Тогда

1) команда $q_1(i, n)$ преобразует регистры 0, 1, i следующим образом:

$$x_0 := n,\ x_1 := x_1 - 1,\ x_i := x_i + 1;$$

2) команда $q_2(i, n, m)$ преобразует регистры 0, 1, i следующим образом:

а) если $x_i > 0$, то $x_0 := n, x_1 := x_1 - 1,\ x_i := x_i - 1$;

б) если $x_i = 0$, то $x_0 := m,\ x_1 := x_1 - 1$.

3). Пусть вычислен вопрос $\bar{u} = (m_1, \dots , m_k)$ и соответствующее значение оракула $F(x_{m_1}, \dots , x_{m_k})$ определено. Тогда команда $q_3(n, \bar{u})$ преобразует регистры 0, 1, $\bar{b}$, следующим образом:

$$x_0 := n,\ x_1 := x_1 - 1,\ \bar{b} := F(x_{m_1}, \dots , x_{m_k}) .$$

(Здесь $\bar{b}$ – регистры, выделенные для записи ответов оракула). Если значение оракула $F(x_{m_1}, \dots , x_{m_k})$ не определено, то команда $q_3(n, \bar{u})$ не выполнима, и результат работы рассматриваемой машины не определен. В таких случаях считается, что *машина застряла на вопросе* $\bar{u}$.

Стандартным способом вводятся коды указанных команд, при этом считается, что код команды располагается в одном регистре. Программа машины состоит из конечного списка команд и задается картежом следующего вида:

$$(t, m, q_1, q_2, \dots , q_m, k, l_1, \dots , l_k), \qquad (*)$$

где t – ограничение на число тактов; m – число команд; q_i – номера регистров, содержащих коды команд; k – число аргументов; l_j – номера регистров, содержащих значения аргументов машины.

Для краткости записи (∗) обозначаются через $(t, \overline{q}, \overline{l})$ и называются *кодами программ*. Перед началом работы машины с программой $(t, \overline{q}, \overline{l})$ в ее счетчике команд указывается номер регистра, содержащего первую выполняемую команду, в счетчик тактов заносится ограничение t, в регистры $\overline{q}$ и $\overline{l}$ – коды соответствующих команд $\overline{Q}$ и значения аргументов машины $\overline{z}$. Полученная конфигурация называется *инициальной программой* и обозначается через $<t, \overline{Q}, \overline{z}>$.

Работа машины производится по тактам. Один такт работы – это выполнение одной команды, включая спрашивающие команды. При этом возможны три случая:

1) машина приходит в состояние q_0 за допустимое количество тактов;

2) машина получает ответы на все свои вопросы, но не приходит в состояние q_0 за допустимое количество тактов;

3) машина застревает на некотором вопросе.

В первых двух случаях считается, что машина *хорошо работает с оракулом*. Пусть $\overline{B_t}(F)$ – множество всех инициальных программ вида $<t, \overline{Q}, z>$, для которых машины хорошо работают с F и с ограничением t; $B_t(F)$ – множество всех инициальных программ из $\overline{B_t}(F)$, для которых машины останавливаются.

2. В этом разделе указаны основные свойства данных вычислений, представляющих положительные аналоги соответствующих свойств рекурсивных вычислений. Для этого вводятся следующие понятия. Буквой y с некоторыми индексами обозначаются *числовые переменные*, значения которых – натуральные числа. Буквами A, B, *возможно* с некоторыми индексами, обозначаются *операторные переменные*, их значениями являются программы машин. Рассматриваются два типа вычислимых отображений: числовые функции и операторы. Аргументами и значениями числовых функций являются числа, а аргументами и значениями операторов могут быть числа и программы. Для обозначения числовых функций используются буквы f, g и записи вида $f(\overline{A}, \overline{y})$, где $\overline{A}$, $\overline{y}$ – наборы операторных и числовых переменных. Аналогично для обозначения операторов применяются буквы S, T и записи вида $S(\overline{A}, \overline{y})$.

Естественным образом определяются числовые функции $f(\overline{y})$ и операторы $S(\overline{y}, \overline{A})$, *F-вычислимые с ограничением* t на машине W. При этом вводятся обозначения: $\{W\}_t^F(\overline{y}) \cong f(\overline{y})$, $\{W\}_t^F(\overline{y}, \overline{A}) \cong S(\overline{y}, \overline{A})$, соответственно. Код программы машины W называется *F-кодом* соответствующей функции или оператора.

Первое свойство связано с последовательной композицией программ. Доказывается, что суперпозиция числовых функций и операторов, F-вычислимых с некоторыми ограничениями, будет также F-вычислимой с некоторым ограничением s равномерно по F-кодам и ограничениям, рассматриваемых функций и операторов.

Второе свойство – существование так называемой *универсальной машины* U_k и рекурсивного оператора $h^k(t, P)$ таких, что для любой программы P машины, F-вычисляющей с ограничением t k-местную числовую функцию от $\overline{y}$, верно соотношение:

$$\{U_k\}_{h^k(t,P)}^F(P, \overline{y}) \cong \{P\}_t^F(\overline{y}).$$

Затем формулируются и доказываются F-вычислимые аналоги S-m-n-теоремы и теоремы о неподвижной точке. Эти свойства аналогичны соответствующим утверждениям из [1, с. 21].

3. Далее указаны некоторые свойства, которые не имеют аналогов в теории алгоритмов.

Теорема 1. *Для любого всюду определенного оракула F существует числовой F-вычислимый оператор, который не является F-вычислимым ни с каким ограничением.*

Доказательство использует так называемый диагональный метод и указанный выше аналог теоремы о неподвижной точке.

Следствие 1. *Существует частично рекурсивная функция, которая не является вычислимой ни с каким ограничением.*

В [3, с. 58] приводится пример частично рекурсивной функции, которую нельзя продолжить общерекурсивной функцией. Рассматриваемые здесь машины работают с некоторыми ограничениями, поэтому верно следующее утверждение.

Теорема 2. *Если оракул F является всюду определенным, то любую функцию, F-вычислимую с некоторым ограничением, можно продолжить тотальной F-вычислимой функцией.*

Следствие 2. *Любую частично рекурсивную функцию, вычислимую с некоторым ограничением, можно продолжить общерекурсивной функцией.*

Следующее утверждение является обратным для предыдущей теоремы.

Теорема 3. *Если любую числовую функцию, F-вычислимую с некоторым ограничением, можно продолжить всюду определенной F-вычислимой функцией, то оракул F можно продолжить тотальной F-вычислимой функцией.*

Доказательство использует тот факт, что любой оракул F является F-вычислимым с некоторым ограничением.

Проблема остановки рассматриваемых машин сводится к выяснению F-вычислимости характеристических операторов множеств $B_t(F)$.

Теорема 4. *Для любого оракула F существует число t_0, начиная с которого характеристические операторы множеств $B_t(F)$ не являются F-вычислимыми ни с каким ограничением.*

Доказательство аналогично доказательству теоремы 6 из [1, 36].

В случае частичного оракула F возникает качественно новая ситуация, когда машина застревает на некотором вопросе. В таком случае вводится следующий оператор $H(t, P, z)$: если P – программа инициальной машины $< t, \bar{Q}, z >$, применяемой к одному аргументу z, то

1) $H(t, P, z) = 0$, при $< t, \bar{Q}, z > \in B_t(F)$;

2) $H(t, P, z) = 1$, при $< t, \bar{Q}, z > \in \bar{B}_t(F) \setminus B_t(F)$; и в остальных случаях $H(t, P, z)$ не определено.

Если этот оператор является F-вычислимым, то говорят, что *оракул F решает проблему остановки для машин, хорошо работающих с F и с ограничением t*. Используя разработанные в [1] методы, для каждого t строится (в рамках системы *ZFC*) оракул F_t такой, что соответствующий оператор $H(t, P, z)$ является F_t-вычислимым. Оракул F_t является ограниченным аналогом гиперарифметического оракула F_0, указанного во введении.

Библиографический список

1. Ганов, В.А. Общая теория вычислений с оракулами / В.А. Ганов, Н.В. Белякин. – Новосибирск, 1989.

2. Морозов, А.С. Машины Шёнфилда / А.С. Морозов. – Новосибирск, 1996.

3. Роджерс, Х. Теория рекурсивных функций и эффективная вычислимость / Х. Роджерс. – М., 1973.