УДК 517.11:518.5

В.А. Ганов, Л.Л. Смолякова

Моделирование гиперарифметической вычислимости на машинах с ограничениями

Особенность вычислений на машинах с оракулом – в наличии 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 процедуры или функции, называемой *оракулом*. Такие оракулы строятся в рамках аксиоматической системы *ZFC* и не обязаны быть рекурсивными. В результате возникают обобщенные вычисления, которые обладают основными чертами и принципами алгоритмической вычислимости, но выходят далеко за пределы рекурсивности.

В работе [1, с. 36] построен так называемый гиперарифметический оракул F_0, с которым вычислимы в точности гиперарифметические функции. Оракул F_0 решает так называемую проблему остановки для машин, хорошо работающих с ним же. Это позволяет в язык вычислений на машинах с оракулом F_0 включать команды, осуществляющие за один такт некоторые виды бесконечных переборов. Другими словами, возникает математическая модель «бесконечно быстрого компьютера». И было изначально понятно, что кое-что из этого можно перенести на «всамделишное» программирование. В связи с этим Н.В. Белякин выдвинул идею организовать рекурсивный процесс, имитирующий поведение гиперарифметического оракула F_0.

В данной работе сделана попытка реализовать эту идею. Здесь изменяется процедура выполнения спрашивающей команды следующим образом. Пусть машина M вычислила вопрос x о поведении некоторой машины W, тогда M приостанавливает свою основную работу и начинает поиск ответа на этот вопрос. Этот поиск заключается в моделировании работы машины W, в результате которого выясняется необходимая информация. Эта информация передается машине M в качестве ответа на вопрос x, и основная работа M возобновляется. Если такой ответ не получен, то работа M считается неопределенной. Теперь при выполнении спрашивающей команды уже не требуется представлять оракул в виде некоторого устройства, позволяющего "сверхбыстро" выдавать ответ на вычисленный вопрос. Такой ответ получается как результат некоторой вычислительной процедуры.

Рассматриваются вычислительные машины с оракулом, описанные в работе [2]. Поэтому здесь используются основные понятия и обозначения из этой работы. В частности, каждая машина имеет ограничение t на число выполняемых команд. Согласно этому ограничению, машина может выполнять не более чем t команд. Если за это время она не пришла в заключительное состояние, то ее работа прекращается и в качестве результата она выдает символ $\varnothing$. Это означает, что ее работа не определена.

Машины, работающие только с таким видом ограничений, рассматривались в работе [3]. Но при моделировании этих машин на современных компьютерах выясняется необходимость введения еще одного вида ограничений d на уровень задаваемых вопросов. По определению, вопрос $\overline{u}$ имеет уровень 1, если входящая в него машина за допустимое число тактов не задает вопросов, обозначение: $d(\overline{u}) = 1$. Далее, по индукции, пусть вопрос $\overline{u}$ содержит машину W, которая за допустимое число тактов задает вопросы $\overline{u}_1,...,\overline{u}_k$ и получает на них ответы, и пусть уровни $d(\overline{u}_1)$,..., $d(\overline{u}_k)$ определены. Тогда $d(\overline{u}) = max\{d(\overline{u}_1)$,..., $d(\overline{u}_k)\}+1$. Теперь ограничение d означает, что данная машина может задавать только вопросы, уровень которых меньше чем d, в противном случае работа этой машины не определена и считается, что она застряла на этом вопросе.

Программа машины содержит конечное число команд, которые разделяются на оперативные и спрашивающие. Команды первого вида позволяют осуществлять любые рекурсивные процедуры и вычисления (см.: [2]). Для описания спрашивающих команд вводятся следующие понятия.

Так же, как в работе [2], определяются коды программ рассматриваемых машин, обозначаемые кортежами $(t, d, \overline{q}, \overline{l})$, и соответствующие им инициальные программы $<t, d, \overline{Q}, \overline{z}>$, где t, d – ограничения указанных видов. Здесь вместо оракула машины работают с определяемым ниже функционалом $H(\overline{u},T,D)$, где T, D – фиксированные числа. В работе такой машины M, работающей с функционалом $H(\overline{u},T,D)$ согласно инициальной программе $<t, d, \overline{Q}, z>$, возможны три случая:

1) M получает ответы на все задаваемые вопросы и останавливается за t тактов с результатом, отличным от $\varnothing$;

2) M получает ответы на все задаваемые вопросы и останавливается за t тактов с результатом $\varnothing$;

3) M застревает при выполнении спрашивающей команды.

В первых двух случаях считается, что M хорошо работает с функционалом $H(\overline{u},T,D)$. Пусть $\overline{B}_{T,D}(H)$ обозначает множество всех инициальных программ вида $<t, d, \overline{Q}, z>$, для которых $t < T$, $d < D$ и соответствующие машины хорошо работают с функционалом $H(\overline{u},T,D)$; $B_{T,D}(H)$ – множество инициальных программ из $\overline{B}_{T,D}(H)$, для которых имеет место случай 1.

Теперь функционал $H(\overline{u},T,D)$ определяется следующим образом: пусть $\overline{u}$ – кортеж, определяющий инициальную программу <*t, d,* $\overline{Q}$*, z*>, тогда

$$H(\overline{u},T,D)=\begin{cases}0, \text{ если } t<T,\ d<D \text{ и } <t,d,\overline{Q},z>\in B_{T,D}(H);\\ 1, \text{ если } t<T,\ d<D \text{ и } <t,d,\overline{Q},z>\in \overline{B}_{T,D}(H)\backslash B_{T,D}(H);\\ \text{не определено в остальных случаях.}\end{cases}$$

Тогда спрашивающая команда $q_3(n, \overline{u})$ преобразует регистры 0, 1, $\overline{b}$, следующим образом:

$x_0:= n, x_1:= x_1-1, \overline{b}:= H(\overline{u},T,D)$.

Если значение функционала $H(\overline{u},T,D)$ не определено, то команда $q_3(n, \overline{u})$ не выполнима и результат работы рассматриваемой машины не определен. В таких случаях считается, что *машина M застряла на вопросе* $\overline{u}$.

Теорема 1. *Функционал* $H(\overline{u},T,D)$ *– рекурсивный.*

Доказательство. Описывается программа машины *M*, которая на аргументе $\overline{u}$ вычисляет значение $H(\overline{u},T,D)$ только с помощью оперативных команд. Сначала *M* выясняет вид вопроса: верно ли, что $\overline{u}$ – кортеж, определяющий инициальную программу вида <*t, d,* $\overline{Q}$*, z*> и $t < T$, $d < D$? Если эти отношения не выполняются, то работа *M* не определена, в противном случае *M* работает согласно программе <*t, d,* $\overline{Q}$*, z*> до ближайшего вопроса. Если вопросов нет, то *M* в течение *t* тактов выяснит поведение <*t, d,* $\overline{Q}$*, z*> и выдаст 0 или 1.

Пусть машина *M* должна выполнить спрашивающую команду вида $q_3(n_1, \overline{u_1})$. Тогда *M* приостанавливает свою основную работу и выясняет вид нового вопроса: верно ли, что $\overline{u_1}$ – кортеж, определяющий инициальную программу вида <t_1, d_1, $\overline{Q_1}$, z_1> и $t_1 < t$, $d_1 < d$? Если эти отношения не выполняются, то работа *M* не определена, в противном случае *M* работает согласно программе <*t, d,* $\overline{Q}$*, z*> до ближайшего вопроса. Если вопросов нет, то *M* в течение t_1 тактов выяснит поведение <t_1, d_1, $\overline{Q_1}$, z_1> , выдаст ответ 0 или 1 машине <*t, d,* $\overline{Q}$*, z*> на ее вопрос и продолжит моделирование ее работы. Если <t_1, d_1, $\overline{Q_1}$, z_1> вычислила некоторый вопрос $\overline{u_2}$, то описанная процедура повторяется для этого вопроса. В описанной процедуре применяются только оперативные команды и идет уменьшение ограничений: $T > t > t_1 > \ldots, D > d > d_1 > \ldots$. Поэтому эта процедура рекурсивная, и если значение $H(\overline{u},T,D)$ определено, то машина *M* полностью промоделирует работу <*t, d,* $\overline{Q}$*, z*> и выдаст значение $H(\overline{u},T,D)$. Теорема доказана.

Таким образом, для любых ограничений *T,D* вычисления с функционалом $H(\overline{u},T,D)$ являются рекурсивными, и их можно осуществлять на достаточно совершенном компьютере. С другой стороны, схема определения функционала $H(\overline{u},T,D)$ в точности соответствует определению гиперарифметического оракула F_0. Поэтому для вычислений с этим функционалом выполняются следующие аналоги основных свойств гиперарифметических функций.

Из определения функционала $H(\overline{u},T,D)$ следует, что он решает проблему остановки для машин, хорошо работающих с $H(\overline{u},T,D)$ и с ограничениями, меньшими, чем *T, D*.

Пусть запись $\{M\}^H_{t,d}(\overline{x})$ обозначает функцию, которую вычисляет машина *M*, работая с ограничениями *t, d* на аргументах $\overline{x}$ с функционалом $H(\overline{u},T,D)$. Такая функция называется *H-вычислимой с ограничениями t, d*, и код программы машины *M* называется *H-кодом* этой функции. Стандартным образом определяются *H-разрешимые множества и отношения* и *H-перечислимые множества* и их *H-коды*.

Из *H*-разрешимости указанной проблемы остановки следует, что класс *H*-разрешимых множеств равномерно замкнут относительно основных теоретико-множественных операций, включая операции навешивания числовых кванторов на определяющие предикаты этих множеств. Аналогично [2] доказывается существование *H*-вычислимой селекторной функции для непустых *H*-перечислимых множеств. Этот факт влечет равномерную замкнутость класса *H*-перечислимых множеств относительно операций объединения, пересечения и проектирования.

Таким образом, сформулированная во введении проблема имитации вычислений была решена для гиперарифметического оракула. Но более сложные проблемы возникают при имитации вычислений с более сильными оракулами. И, наверное, их решение можно осуществить в рамках арифметики с переменными для чисел разных уровней нестандартности.

Библиографический список

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

2. Ганов, В.А. Ограниченные вычисления с оракулами / В.А. Ганов, В.Р. Карымов // Известия АлтГУ. – Барнаул, 2009. – №1.

3. Ганов, В.А. Имитация гиперарифметической вычислимости рекурсивными функциями / В.А. Ганов, Л.Л. Смолякова // МАК-2008 : мат. XI регион. конф. по математике. – Барнаул, 2008.