

*В.А. Ганов, В.Р. Карымов***Компьютерная имитация вычислений с оракулами***V.A. Ganov, V.R. Karymov***Computer Simulation of Computations with Oracles**

Рассматриваются вычисления на абстрактных машинах с оракулом. Ставится цель: перенести основные конструкции таких вычислений в язык программирования на современных компьютерах. Для этого в работу машин и процедуру их общения с оракулом вводятся специальные ограничения и исследуются изменения, возникающие в основных принципах программирования.

Ключевые слова: машина с оракулом, проблема остановки, перечислимые множества, арифметические множества.

Данные исследования в полной мере относятся к компьютерным наукам, являющимся ядром теоретических и практических знаний, которые используют в своей работе специалисты в области вычислительной техники, программирования, информационных систем и технологий. Предмет исследования – организация вычислений и принципы программирования на абстрактных вычислительных машинах с оракулом. Описание машин Тьюринга с оракулом возникло в 1930-х гг. почти одновременно с определением рекурсивных вычислений, и наиболее глубокое их исследование осуществлено в 1970–1980-х гг. в работах [1–3]. В программах этих машин предусмотрены так называемые спрашивающие команды, позволяющие осуществлять некоторые виды неалгоритмических шагов. При подходящем выборе оракула такие машины могут «сверхбыстро» просматривать бесконечные последовательности вычислимых объектов и выбирать из них необходимый элемент. Они могут вычислять вопросы о своем собственном поведении или о свойствах вычислимых на них объектах и получать соответствующие ответы от оракула. Наблюдая полученные ответы, машина может изменять свою дальнейшую работу так, чтобы быть более эффективной в достижении некоторой цели. В настоящее время такие возможности данных машин привлекают все большее внимание специалистов компьютерной науки.

Широко распространено мнение, будто вычислительная машина не может мыслить, на том основании, что она (в отличие от человека) всего только воспроизводит рекурсивные процедуры (в пределах своего объема памяти), не понимая, что она делает. С одной стороны, данное мнение выглядит весьма

We consider computations on abstract machines with an oracle. The aim of investigation is to transfer the basic design of such algorithms in a programming language on modern computers. In order to solve this problem the authors bring special limitations in the work of machines and their communication with the oracle and explore the changes that occur in the basic principles of programming.

Key words: machine with the oracle, problem of the stop, enumerable sets, arithmetical sets.

курьезно. Ведь вопросы: сколь нетривиальными могут быть рекурсивные процедуры, какие именно мыслительные действия они способны воспроизвести? – даже не ставятся. Машина есть машина, и потому она не может мыслить по определению.

С другой стороны, указанные особенности машин с оракулом позволяют все более адекватно моделировать различные виды деятельности человека, включая его мышление. И потому возникает мысль создать специальный язык программирования на машинах с оракулом, который был бы максимально приближен к современным языкам программирования и сохранял бы главные свойства таких вычислений. Наиболее естественный подход для решения этой задачи – использовать имеющийся язык программирования на машинах с оракулом и ввести в него подходящие виды ограничений так, чтобы полученные программы можно было бы использовать для реальных вычислительных машин.

Именно такая задача ставится и решается в рассматриваемых исследованиях. В них главное внимание уделяется формализации процедуры общения машины с оракулом и разрабатываются различные виды ограничений в работе таких машин.

1. Вычисления с оракулом и основные принципы программирования. В названных выше статьях оракулы предназначались для машин Тьюринга. Но в связи с поставленной целью удобнее вернуться к адресному программированию. Поэтому определяются специальные виды вычислительных машин с оракулом, программы которых во многом соответствуют программам современных вычислительных машин. Такие машины обладают занумерованным набором регистров (или «ячеек памяти»), которые могут содержать числа и коман-

ды. Номера регистров являются их адресами. Допускается создание новых регистров по ходу вычислений. Имеются команды следующих четырех видов:

а) сложение чисел: указываются адреса слагаемых и адрес для помещения результата;

б) сравнение и условный переход: указываются адреса аргументов и адреса команд, которые надлежит выполнять в зависимости от результата сравнения;

в) обращение к оракулу: указывается адрес вопроса и адрес для полученного ответа (если он будет получен);

г) завершение вычислений: указывается адрес результата.

Для простоты рассуждений считается, что каждый регистр имеет неограниченную емкость, т.е. может содержать сколь угодно большое число или команду со сколь угодно большими адресами. Эти команды отождествляются со своими кодами, являющимися числовыми кортежами, составленными из адресов соответствующих регистров. Кодом программы машины является список кодов ее команд. Вводится понятие инициальной машины – это кортеж, включающий код программы, содержимое регистров и адрес начальной команды.

Оракул – это некоторая функция F , определенная на кодах инициальных машин. Построение этой функции F осуществляется в рамках классической теории множеств. Машина с оракулом работает согласно своей программе и может вычислять вопросы, являющиеся кодами инициальных машин. Ожидаемые ответы – это информация о поведении соответствующей машины или информация о свойстве вычислимого на этой машине объекта.

В частности, возможен так называемый гиперарифметический оракул H , который на вопрос z дает ответ ноль, если инициальная машина с кодом z и оракулом H останавливается с некоторым результатом, и ответ единица, в случае ее бесконечной работы. Процесс построения H использует довольно сложную теоретико-множественную конструкцию, основанную на трансфинитной индукции. H является частичной функцией, поэтому в работе машины с оракулом H возможна особая ситуация застревания, когда H не имеет ответа на заданный вопрос. Подобная ситуация отсутствует в обычном рекурсивном вычислении, и именно ее наличие позволяет моделировать в языке рассматриваемых вычислений разнообразные неалгоритмические процессы и явления. Например, с оракулом H становится распознаваемой проблема остановки машин, не застревающих с H , существует H -вычислимый аналог функции выбора и положительно решается так называемая континуум-гипотеза [1–2].

В общем случае производить фактические вычисления на машинах с произвольным оракулом F невозможно, но тем не менее основные принципы программирования для таких машин совпадают

с принципами обычного программирования. В первую очередь речь идет о принципах замкнутости класса F -разрешимых множеств и класса F -вычислимых функций, относительно основных операций над этими объектами. Далее рассматриваются важные принципы существования универсальной F -вычислимой функции и F -вычислимые аналоги S - m - n -теоремы и теоремы о неподвижной точке. Универсальная функция характеризует вычислительные возможности данного языка программирования, и потому ее существование желательно. Указанная S - m - n -теорема также является естественным утверждением любой теории вычислений, а значение теоремы о неподвижной точке можно пояснить следующим образом. Уже на заре программирования стали применяться такие приемы, как переадресация, согласно которой программа по ходу ее исполнения подвергалась некоторому преобразованию и тем самым сама становилась объектом вычислительной работы наряду с исходными данными. Но еще раньше в теории алгоритмов была доказана теорема о неподвижной точке, которая тоже описывает подобную ситуацию. По этой теореме для заданной вычислимой процедуры G можно построить программу с числовым кодом \bar{e} , осуществление которой для исходных данных \bar{x} сводится к применению G к аргументам e и \bar{x} , т.е. программа e становится аргументом. Выразительный пример: программа вычисляет свой собственный код и совершает над ним заранее предусмотренное вычислительное действие. Понятно, что наличие подобной возможности существенно усиливает рассматриваемую вычислимость.

Кроме того, существенное расширение класса вычислимых объектов происходит в тех случаях, когда для данного оракула F оказываются F -разрешимыми некоторые варианты проблемы остановки для машин, работающих с F .

Как уже отмечалось выше, вычисления с оракулом привлекают все большее внимание специалистов *Computer Science*, особенно в связи с разработкой программистских языков «высокого уровня» (PL/1, Паскаль, Си, C++ и Java) и так называемых языков сценариев [4–7]. Выражение «более высокий уровень» для языка программирования означает, что многие детали обрабатываются в нем автоматически, поэтому программистам при решении соответствующих проблем приходится писать меньше кода. Языки сценариев создавались для «склеивания» мощных готовых вычислимых процедур в предположении, что их программы уже существуют и надо лишь связать их между собой. Они не содержат заведомых ограничений, все компоненты и величины представляются единообразно и могут использоваться в любой ситуации. Элементарные операции в языке сценариев обладают очень богатой функциональностью, и, например, один оператор способен выполнить сотни или даже тысячи машинных команд.

Подобные положения и принципы по определению принимаются в языках программирования машин с оракулом, и это оправдывает указанный интерес к данным вычислениям в настоящее время. Поэтому становится понятной важность указанной выше цели о создании такого программистского языка, который по своим свойствам был бы максимально похож на языки вычислений с оракулами, оставаясь при этом чисто алгоритмическим языком, пригодным для компьютерного пользования. Главное препятствие к достижению этой цели – заложенная в определении рассматриваемых машин возможность бесконечного числа тактов работы, включая число обращений к оракулу. В описываемых здесь исследованиях устранение этого препятствия осуществляется ценой ослабления оракулов путем модификации общения машины с приданным ей оракулом без ущерба для сходства с их классическими прообразами.

2. Вычисления с ограничением на время работы. Сначала в работу машины с оракулом F вводится ограничение t на время работы. Если за t тактов машина не пришла в заключительное состояние, то ее работа считается бесконечной. В этом случае указанные принципы программирования изменяются незначительно. Существует F -вычислимая универсальная функция, выполняются F -вычисляемые аналоги S - m - n -теоремы и теоремы о неподвижной точке, выполняются указанные принципы замкнутости соответствующих классов множеств и функций. Конструируемые в этих утверждениях объекты остаются по-прежнему вычислимыми с оракулом F и с некоторым новым ограничением t_1 , но это ограничение вычисляется рекурсивно по t . Полная проблема остановки и ее естественные варианты для машин, работающих с ограничением t , не являются F -разрешимыми. И только в случае фиксированного ограничения t можно построить оракул F_t , решающий эту проблему для машин, не застревающих при работе с F_t [8].

Исследование данного вида ограничений позволило получить некоторые результаты, относящиеся к известным проблемам классической теории алгоритмов. Например, в [9, с. 58] доказывается существование частично рекурсивной функции, которую нельзя продолжить всюду определенной рекурсивной функцией. В данном исследовании доказано следующее утверждение, связанное с этим фактом.

Теорема 1. *Частично рекурсивная функция может быть продолжена до всюду определенной рекурсивной функции тогда и только тогда, когда она вычислима с некоторым ограничением на число тактов работы.*

Далее, в свое время в теории алгоритмов много внимания уделялось понятию сложности вычислений [10]. Выше отмечалось, что вычисления с оракулом позволяют «сверхбыстро» осуществлять некоторые процедуры, и поэтому вопрос о сложности таких вычислений не ставился. Но введение огра-

ничений на число тактов работы впервые позволяет ввести следующий аналог так называемых сигнализирующих функций.

Определение 1. Пусть функция $f(x)$ вычислима с оракулом F и с ограничением t . Числовая функция $h(t)$ называется *сигнализирующей* для $f(x)$, если существует машина w , которая вычисляет эту функцию с ограничением $h(t)$, но без оракула.

В ходе данных исследований для фиксированного числа t и индукцией по n строится специальная последовательность оракулов $F_t^1, F_t^2, \dots, F_t^n, \dots$, каждый из которых распознает проблему остановки для машин, работающих с предыдущим оракулом и с ограничением t . Следующее утверждение демонстрирует применение нового понятия к вычислениям с этими оракулами.

Теорема 2. *Для каждого числа n сигнализирующей функцией для F_t^n -вычисляемых функций является $h(t) = t^{n+1}$.*

Для дальнейшей характеристики вычислений с ограничением на время работы привлекаются алгоритмические конструкции, связанные с арифметическими и гиперарифметическими множествами. Для каждого вида таких множеств имеются две эквивалентные формы определения: первая использует кванторные формы с рекурсивными предикатами; вторая – вычисления относительно специального вида оракулов. В данных исследованиях в эти определения вводится ограничение на число тактов работы соответствующих машин.

Определение 2. Множество S называется *арифметическим относительно ограничения t* , если оно может быть выражено в виде:

$$S = \{x / (Q_1 y_1) \dots (Q_m y_m) R(x, y_1, \dots, y_m)\},$$

где \bar{Q}_i – кванторы \forall или \exists ; $R(x, y_1, \dots, y_m)$ – предикат, являющийся разрешимым на машине, работающей без оракула и с ограничением t .

Индукцией по n определяется следующая последовательность множеств-оракулов:

$$E_0, E_1, \dots, E_n, \dots,$$

где $E_0 = \emptyset$, и E_{n+1} получается из E_n с помощью так называемой джамп-операции:

$$E_{n+1} = \{ \langle z, x \rangle / \{z\}^{E_n}(x) \text{ определено} \}.$$

Теорема 3. *Множество S является арифметическим относительно ограничения t тогда и только тогда, когда оно является разрешимым относительно какого-нибудь оракула E_n с некоторым ограничением t_1 , при этом данное утверждение выполняется равномерно.*

Таким образом, замена рекурсивной вычислимости на вычисление с ограничением на время сохраняет эквивалентность этих форм определения арифметических множеств. В дальнейшем исследовании доказывается, что такая же согласованность имеет место на примере определения гиперарифметических множеств [11].

3. Вычисления с ограничением на диалог машины с оракулом. Здесь рассматриваются ограничения на так называемую глубину задаваемых вопросов и количество вопросов. Вводится необычный способ выполнения спрашивающей команды. Если машина w вычислила вопрос w_1 , то она приостанавливает свою работу до получения ответа. С этого момента начинается моделирование работы машины w_1 на тех же условиях, что и у машины w . Может случиться, что и машина w_1 вычислила некоторый вопрос w_2 , тогда и ее работа приостанавливается до получения ответа. Пусть, например, работа w_2 полностью промоделирована, тогда однозначно будет определен ответ на вопрос w_2 , этот ответ передается машине w_1 и возобновляется ее дальнейшая работа. Если работа w_1 также закончена и определен ответ на вопрос w_1 , то этот ответ передается машине w и возобновляется ее дальнейшая работа. Чтобы такие вычисления можно было осуществлять на обычных компьютерах, предполагается, что имеется дополнительное ограничение на глубину задаваемых вопросов. При таком соглашении спрашивающие команды можно выполнять без оракула путем подходящих вычислительных процедур. Тем самым вычисления с оракулом можно моделировать на обычных вычислительных машинах. Доказывается, что такие вычисления будут ограничены, но для них без существенных изменений сохраняются указанные выше принципы программирования и некоторые специфические свойства вычислений с оракулом [12].

Далее вводится ограничение на количество задаваемых вопросов оракулу. Ясно, что если машина с оракулом останавливается с некоторым результатом, то она задает лишь конечное число вопросов. Поэтому данное ограничение выглядит искусственным и не следует ожидать, что оно внесет какие-нибудь существенные изменения в класс вычислимых объектов. Но особенность такого ограничения в том, что задаваемые вопросы касаются поведения машин, которые также могут задавать только огра-

ниченное число вопросов, в противном случае оракул на таких вопросах не определен. В результате происходит уменьшение области определения оракула и, следовательно, уменьшение класса функций, вычислимых с таким оракулом.

Первые исследования вычислений с таким ограничением выделяют новый вид алгоритмической проблемы – проблему распознавания конечности числа вопросов оракулу. Показывается, что для любого оракула F эта проблема эквивалентна проблеме остановки инициальных машин, работающих с F . Поэтому полная формулировка этой проблемы также не является F -разрешимой. Но если оракул F – частичная числовая функция, то возникает возможность определить ограниченный вариант этой проблемы для машин, не застревающих с F . Строится оракул F_1 , который непосредственно решает проблему распознавания конечности числа вопросов для машин, хорошо работающих с F_1 . Доказывается, что класс F_1 -разрешимых множеств не выходит за пределы класса арифметических множеств.

Затем фиксируется некоторое число s , и рассматриваются вычисления на машинах с оракулом, при работе которых разрешается задавать не более чем s вопросов. Ясно, что такие вычисления включают обычные рекурсивные вычисления, но, с другой стороны, при подходящем оракуле F класс функций, F -вычислимых с таким ограничением, шире, чем множество рекурсивных функций. Используя метод построения гиперарифметического оракула H , строится оракул F_2 , который решает проблему остановки для машин, работающих с данным видом ограничений и не застревающих с F_2 . Оракул F_2 слабее H , но он обладает аналогами наиболее важных свойств H -вычислимых функций. В частности, существуют F_2 -вычислимые селекторные функции $\nu_s(w_1, w_2)$ и $\mu_s(z)$, которые впервые были построены для гиперарифметической вычислимости [12–13].

Некоторые из рассматриваемых здесь вопросов обсуждаются в [14].

Библиографический список

1. Ганов В.А. Обобщенно-конструктивный континуум // Сиб. мат. журн. – 1973. – Т. 14, №5.
2. Белякин Н.В. Вычисления с оракулами // Труды ИМ СОАН СССР. – 1989. – Т. 12.
3. Ганов В.А., Белякин Н.В. Общая теория вычислений с оракулом. – Новосибирск, 1989.
4. Wall L., Christiansen T., Schwartz R. Programming Perl. 2nd ed. – O'Reilly & Associates, Sebastopol, CA, 1996.
5. Lutz M. Programming Python. – O'Reilly & Associates, Sebastopol, CA, 1996.
6. O'Hara R., Gomberg D. Modern Programming Using REXX. – Prentice Hall, Englewood Cliffs. – 1988. – №1.
7. Ousterhout J. Td and Tk Toolkit. – Addison-Wesley, Reading, MA, 1994.
8. Ганов В.А., Карымов В.Р. Ограниченные вычисления с оракулами // Известия АлтГУ. – 2009. – №1(61).
9. Роджерс Х. Теория рекурсивных функций и эффективная вычислимость. – М., 1972.
10. Хартманис Ю., Хопкрофт Дж. Обзор теории сложности вычислений // Кибернетический сборник. – М., 1974. – Вып. 14.
11. Карымов В.Р. Арифметическая и гиперарифметическая вычислимость относительно вычислений с ограничениями // Известия АлтГУ. – 2010. – №1(65).

12. Карымов В.Р. Вычисления с оракулом и фиксированным ограничением на число вопросов // Известия АлтГУ. – 2010. – №1/2(65).

13. Ганов В.А. Арифметическая и гиперарифметическая вычислимость относительно вычислений с ог-

раничениями // Известия АлтГУ. – Барнаул, 2010. – №1(65).

14. Белякин Н.В., Гайлит Е.В. Рекурсивная имитация гиперарифметической вычислимости // Философия, математика, лингвистика: мат. науч. конф. – СПб., 2009.