Классическая вычислимость и признаки индетерминизма[1]

А. М. Анисов

abstract. Converse to the common belief, the theory of classic deterministic computability allow ambiguous interpretations as it is shown in this paper. It is due to the uncertainty of the theory's fundamental notions of finiteness and natural series. Actually focus is made on effects of the emergence and development of the non-standard analysis on the theory of computability.

Распространенное в научном сообществе представление о вычислении как о жестко детерминированной последовательности шагов, выполняемых согласно точному предписанию (алгоритму), не охватывает всех аспектов идеи вычислимости. Подобно тому как попытки избежать фаталистических выводов из доктрины детерминизма привели к созданию новой ветви логики — многозначной логике, — так и теория вычислимости нуждается в обобщениях, допускающих недетерминированные вычисления. Это важно не только в плане развития самой теории, но, в еще большей степени, в аспекте приложений теории вычислимости к моделированию реальности. В последнее время предпринимаются усилия по созданию вычислительных концепций тех или иных сфер реальности или даже всего универсума в целом. Если при этом пользоваться только стандартной теорией детерминированной вычислимости, картина реальности также окажется полностью детерминированной, со всеми вытекающими отсюда неприятными философскими следствиями. В таких условиях задача построения альтернативных теорий вычислимости, допускающих в той или иной форме элементы индетерминизма, становится особенно актуальной.

Однако в философских исследованиях в нашей стране и за рубежом идея недетерминированной вычислимости не находит

[1]Работа выполнена при поддержке РГНФ, грант № 07-03-00203а.

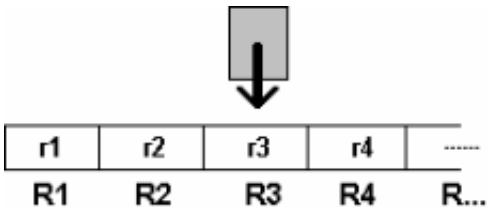
заметного применения. Причина, по-видимому, заключается в том, что философы либо довольствуются традиционной теорией детерминированной вычислимости, либо объявляют решаемые ими задачи принципиально невычислимыми. Так, в философии ИИ уже долгое время идет дискуссия между сторонниками и противниками вычислительного подхода. Первые возлагают надежды на компьютеры нетрадиционной архитектуры (параллельные машины, клеточные автоматы, квантовые компьютеры и др.), вторые не без оснований указывают, что подобные компьютеры не выводят нас за границы классической вычислимости. Например, хотя квантовые вычисления в ряде случаев оказываются эффективнее вычислений по Тьюрингу, они не выходят за границы класса стандартных вычислимых функций. Согласно выдающемуся математику и физику Р. Пенроузу, «все, что в принципе можно получить с помощью квантового компьютера, можно в принципе получить и с помощью соответствующей машины Тьюринга, снабженной генератором случайных чисел» [8, с. 546].

1 Теория классической МНР-вычислимости

Как известно, аксиоматический метод задания теорий состоит в принятии ряда утверждений в качестве аксиом с последующим выведением следствий из них. Теория при таком подходе представляет собой множество высказываний, замкнутое относительно выводимости. Существует альтернативный метод получения теорий, когда исходят не из высказываний, а из некоторого строго заданного набора объектов, с которыми разрешается проводить точно определенные операции. Это так называемый *генетический метод* построения теорий. При построении теорий вычислимости, как правило, используют генетический метод. Мы кратко рассмотрим основные элементы варианта теории классической вычислимости или эффективной вычислимости, изложенной в книге [7], с несущественными отличиями от первоисточника. Эта генетическая теория весьма подходит для наших целей. Исходными объектами здесь являются натуральные числа, а операции с ними осуществляет просто устроенный логический компьютер, все допустимые действия которого будут строго описаны. Множество натуральных чисел (т.е. множе-

ство целых положительных чисел, к которому добавлено число 0) обозначим через **N**.

Идеальный компьютер, с которым нам предстоит работать, называется *машиной с неограниченными регистрами (МНР)*; его также называют *адресной машиной (RAM)*. МНР содержит бесконечное число регистров или ячеек памяти, обозначаемых через $R_1, R_2, R_3 \ldots, R_n, \ldots$ Каждый из регистров R_i в любой момент времени содержит некоторое натуральное число, обозначаемое через переменную r_i. Кроме того, имеется пишущая головка, которая, перемещаясь от регистра к регистру, может изменять содержимое регистра, выполняя некоторые команды или инструкции. Графически МНР изображена на приведенном рисунке.



Команды, или инструкции, выполняемые пишущей головкой, очень просты и сводятся к следующим типам.

Команда обнуления. Для каждого $n > 0$ может быть выполнена команда обнуления $Z(n)$. Команда $Z(n)$ заменяет содержимое регистра R_n на 0 и не затрагивает содержимое других регистров. Действие МНР в ответ на команду $Z(n)$ обозначим через $r_n := 0$ (читается «r_n становится 0» или «r_n присваивается 0»).

Команда прибавления единицы. Для каждого $n > 0$ выполняется команда $S(n)$. Результат применения команды состоит в увеличении на 1 содержимого регистра R_n. Другие регистры не затрагиваются. В этом случае пишем $r_n := r_n + 1$.

Команда переадресации. Для всех $m, n > 0$ имеется команда (m, n). Ответом МНР на эту команду является замена содержимого регистра R_n содержимым регистра R_m, т.е. перенос числа r_m в регистр R_n. Другие регистры (включая R_m) не затрагиваются. Результат (m, n) записываем как $r_n := r_m$. (В действи-

тельности эта команда избыточна и в МНР-вычислениях без нее можно обойтись.)

Команда условного перехода. Для всех $m > 0$, $n > 0$ и $q \geq 0$ имеется команда $J(m, n, q)$. Встретив эту команду в программе, МНР сравнивает содержимое регистров R_m и R_n. Если $r_m = r_n$, то МНР переходит к выполнению команды q; если команда с номером q отсутствует, МНР завершает работу. Если же $r_m \neq r_n$, МНР переходит к выполнению следующей по порядку команды программы (если таковая имеется; в противном случае МНР останавливается).

На этом перечень типов МНР-команд закончен. Команды первых трех типов называются *арифметическими.*

Вычисления на МНР-машине начинаются с *начальной конфигурации* — последовательности чисел $r_1, r_2, r_3, \ldots$, содержащихся в соответствующих регистрах $R_1, R_2, R_3 \ldots$ до вычисления. При этом предполагается, что в каждой начальной конфигурации содержится лишь конечное количество отличных от нуля чисел, включая случай, когда таких чисел вообще нет (тогда $r_i = 0$ для всех $i \geq 1$). Отсюда вытекает, что либо во всех регистрах содержатся одни нули, либо существует $n \geq 1$ такое, что в регистре R_n содержится число $r_n \neq 0$, тогда как в бесконечной последовательности регистров $R_{n+1}, R_{n+2}, \ldots$ содержатся одни нули.

МНР-программа — это конечная непустая последовательность команд (инструкций) $I_1, I_2, \ldots, I_n$ перечисленных выше типов.

Примером МНР-программы будет следующий набор команд π^-.

1. $J(1, 2, 6)$
2. $S(2)$
3. $S(3)$
4. $J(1, 2, 6)$
5. $J(1, 1, 2)$
6. $T(3, 1)$

МНР начинает выполнение программы с первой команды (это всегда инструкция I_1). Выполнив очередную команду I_j, если она не последняя и не была командой условного перехода, МНР переходит к выполнению команды I_{j+1}, расположенной ниже в наборе команд. Если выполняемая команда оказалась командой условного перехода $J(r_m, r_n, q)$ и $r_m \neq r_n$, то опять-таки МНР выполняет следующую по порядку команду (если таковая существует). Если же $r_m = r_n$, МНР переходит к выполнению команды с номером q. МНР-вычисление останавливается тогда и только тогда, когда либо нет следующей команды в наборе команд (выполнена последняя инструкция I_n в последовательности $I_1, I_2, \ldots, I_n$), либо при выполнении команды условного перехода требуется перейти на команду, номер которой отсутствует в последовательности инструкций $I_1, I_2, \ldots, I_n$. Например, выполнение команды $J(1, 1, 0)$ обязательно приведет к остановке машины, а выполнение команды $J(1, 1, 1)$ в любой ситуации вернет вычисление к первой инструкции I_1.

В случае остановки вычислений содержимое $r_1, r_2, r_3, \ldots$ регистров $R_1, R_2, R_3 \ldots$ называется *заключительной конфигурацией*. В заключительной конфигурации лишь конечное количество регистров могут содержать ненулевые значения. Разумеется, вычисления не обязательно заканчиваются.

Пусть π — программа и $r_1, r_2, r_3, \ldots$ — начальная конфигурация. Через $\pi(r_1, r_2, r_3, \ldots)$ будем обозначать вычисления по программе π с начальной конфигурацией $r_1, r_2, r_3, \ldots$; запись $\pi(r_1, r_2, r_3, \ldots) \downarrow$ означает, что вычисление π на входе $r_1, r_2, r_3, \ldots$ в конце концов останавливается; запись $\pi(r_1, r_2, r_3, \ldots) \uparrow$ означает, что вычисление π на входе $r_1, r_2, r_3, \ldots$ никогда не останавливается. Часто говорят, что останавливающееся вычисление *сходится*, а никогда не останавливающееся — *расходится*.

Так как начальная конфигурация может, по определению, содержать лишь конечное количество отличных от нуля членов, будем использовать выражение вида $\pi(x_1, x_2, \ldots, x_n)$ с индивидными переменными $x_1, x_2, \ldots, x_n$ для указания на то, что программа π может применяться к *любой* начальной конфигурации, в которой регистры $R_{n+1}, R_{n+2}, \ldots$ содержат одни нули. Отсюда содержание регистров $R_1, \ldots, R_n$ может быть любым, что оправдывает применение в записи индивидных переменных

по натуральным числам. Запись вида $\pi(a_1, a_2, \ldots, a_n)$ означает, что π применяется к *конкретной* начальной конфигурации $r_1, r_2, \ldots, r_n, 0, 0, 0, \ldots$ В этом случае $a_1, a_2, \ldots, a_n$ играют роль индивидных констант. Конечно, должно быть разрешено и смешанное применение индивидных переменных и индивидных констант. Например, $\pi(y, z, b, x, a)$, где (применяя привычные обозначения) y, z, x — переменные, а b, a — константы. Индивидные переменные и индивидные константы назовем *термами*. Для термов будем использовать обозначения $t, t_1, t_2, \ldots, t_n, \ldots$ Это позволяет применять наиболее общую форму записи $\pi(t_1, t_2, \ldots, t_n)$, где каждый терм t_i $(1 \leq i \leq n)$ является либо переменной, либо константой, а порождаемые выражением $t_1, t_2, \ldots, t_n$ начальные конфигурации объеденены общим условием $r_{n+1} = 0, r_{n+2} = 0, \ldots$

Та же самая программа π, т. е. та же самая последовательность команд, может применяться к начальным конфигурациям любой длины. Поэтому при $n \neq m$ записи $\pi(t_1, t_2, \ldots, t_n)$ и $\pi(t_1, t_2, \ldots, t_m)$ правомерны. Это означает, что *начальная конфигурация готовится независимо от программы.*

Предположим, $f(t_1, t_2, \ldots, t_n)$ — функция (тотальная или частичная), определенная на натуральных числах. Точнее, область определения $Dom(f)$ функции $f(t_1, t_2, \ldots, t_n)$ представляет собой декартово произведение множеств $D_1 \times D_2 \times \ldots \times D_n$, где $D_i = \mathbf{N}$, если t_i является индивидной переменной, и $D_i = \{\mathbf{n}\}$, если t_i есть индивидная константа, обозначающая конкретное натуральное число $\mathbf{n}$. Что означает выражение «функция $f(t_1, t_2, \ldots, t_n)$ вычислима на МНР»? Естественно представлять себе вычисление значения $f(r_1, r_2, \ldots, r_n)$ для конкретных натуральных чисел $r_1, r_2, \ldots, r_n$ как выполнение некоторой программы π с начальной конфигурацией $r_1, r_2, \ldots, r_n, 0, 0, 0, \ldots$, т.е вычисление $\pi(r_1, r_2, \ldots, r_n)$. Если такое вычисление останавливается, условимся считать *результатом вычислений* содержимое регистра R_1. Содержимое других регистров тогда будет содержать избыточную информацию. Если $\pi(r_1, r_2, \ldots, r_n) \downarrow$ и после остановки $r_1 = b$, то пишем $\pi(r_1, r_2, \ldots, r_n) \downarrow b$ и говорим, что вычисление $\pi(r_1, r_2, \ldots, r_n)$ *сходится* к b. Если $f(r_1, r_2, \ldots, r_n) = b$ и $\pi(r_1, r_2, \ldots, r_n) \downarrow b$, то ясно, что программа π правильно вычисляет значение функции f на аргументах $r_1, r_2, \ldots, r_n$. В случае,

если f — частичная функция, значение которой не определено на $r_1, r_2, \ldots, r_n$, то программа, вычисляющая f, не должна сходиться к какому-либо числу. Следовательно, должно иметь место $\pi(r_1, r_2, \ldots, r_n) \uparrow$. Дадим теперь точное определение.

ОПРЕДЕЛЕНИЕ 1. Программа $\pi(t_1, t_2, \ldots, t_n)$ *МНР-вычисляет* $f(t_1, t_2, \ldots, t_n)$, если для всех $< r_1, r_2, \ldots, r_n >$ из $Dom(f)$ и для каждого $b \in \mathbf{N}$ выполняются следующие утверждения: во-первых, $\pi(r_1, r_2, \ldots, r_n) \downarrow b$ тогда и только тогда, когда $f(r_1, r_2, \ldots, r_n) = b$, и, во-вторых, $\pi(r_1, r_2, \ldots, r_n) \uparrow$ тогда и только тогда, когда значение $f(r_1, r_2, \ldots, r_n)$ не определено.

По определению, функция $f(t_1, t_2, \ldots, t_n)$ является *МНР-вычислимой*, если существует программа $\pi(t_1, t_2, \ldots, t_n)$, которая МНР-вычисляет $f(t_1, t_2, \ldots, t_n)$.

Вернемся к первому и пока единственному примеру МНР-программы π^-. Какую функцию она вычисляет? Ответ зависит от длины начальной конфигурации, подаваемой на вход π^-. Оказывается, если положить $\pi^-(x, y)$, то это будет двухместная функция h, заданная следующим образом:

$$h(x,y) = \left\{ \begin{array}{l} x - y, \text{ если } x \geq y \\ \text{в противном случае значение не определено} \end{array} \right\}.$$

Иными словами, эта *частичная* функция осуществляет вычитание на множестве натуральных чисел. Взяв $\pi^-(x)$, получим вычисление *тотальной* тождественной функции $h(x) = x$. Аналогично разбирается случай $\pi^-(x, y, z)$, который снова дает *частичную* функцию $h(x, y, z)$:

$$h(x,y,z) = \left\{ \begin{array}{l} (x - y) + z, \text{ если } x > y \\ z, \text{ если } x = y \\ \text{в противном случае значение не определено} \end{array} \right\}.$$

Однако дальнейшее увеличение длины начальной конфигурации к новым функциям не приводит, так как программа π^- в ходе выполнения затрагивает только три первых регистра. В результате $\pi^-(x, y, z)$ и $\pi^-(x, y, z, x_1, x_2, \ldots, x_n)$ вычисляют одну и ту же функцию, так что получаем либо $h(x, y, z) = h(x, y, z, x_1, x_2, \ldots, x_n)$, либо обе эти функции не определены.

Пусть $R(x_1, x_2, \ldots, x_n)$ — числовой n-местный предикат и $f_R(x_1, x_2, \ldots, x_n, c_1, c_2, \ldots, c_m)$ (здесь $n \geq 1$ и $m \geq 0$) — тотальная числовая n-местная функция[2], определенная следующим образом: для любых натуральных $r_1, r_2, \ldots, r_n$

$$f_R(r_1, r_2, \ldots, r_n, c_1, c_2, \ldots, c_m) = \left\{ \begin{array}{l} 1, \text{ если } R(r_1, r_2, \ldots, r_n) \\ \quad \text{истинно} \\ 0, \text{ если } R(r_1, r_2, \ldots, r_n) \\ \quad \text{ложно} \end{array} \right\}.$$

Функция $f_R(x_1, x_2, \ldots, x_n, c_1, c_2, \ldots, c_m)$ называется *характеристической функцией* предиката $R(x_1, x_2, \ldots, x_n)$. Предикат $R(x_1, x_2, \ldots, x_n)$ называется *разрешимым*, если какая-либо его характеристическая функция $f_R(x_1, x_2, \ldots, x_n, c_1, c_2, \ldots, c_m)$ МНР-вычислима.

Предположим, для предиката $R(x_1, x_2, \ldots, x_n)$ нашлась следующая частичная функция $f_R(x_1, x_2, \ldots, x_n, c_1, c_2, \ldots, c_m)$, оказавшаяся МНР-вычислимой:

$$f_R(r_1, r_2, \ldots, r_n, c_1, c_2, \ldots, c_m) = \left\{ \begin{array}{l} 1, \text{ если } R(r_1, r_2, \ldots, r_n) \\ \quad \text{истинно} \\ \text{не определена, если} \\ \quad R(r_1, r_2, \ldots, r_n) \text{ложно} \end{array} \right\}.$$

Тогда назовем предикат $R(x_1, x_2, \ldots, x_n)$ *полуразрешимым*. В теории МНР-вычислимости доказано, что всякий разрешимый предикат полуразрешим, но обратное неверно.

Наконец, возможен случай, когда предикат $R(x_1, x_2, \ldots, x_n)$ не является ни разрешимым, ни полуразрешимым; в таком случае предикат $R(x_1, x_2, \ldots, x_n)$ *неразрешим*.

Примеры разрешимых, полуразрешимых и неразрешимых предикатов хорошо известны. Так, после надлежащей кодировки (гёделизации) формул и конечных последовательностей формул языков классических исчислений высказываний и предикатов и формальной арифметики предикат на множестве гёделевых номеров формул логики высказываний *Тавтология(x)* окажется

[2]Напомним, что местность предиката или функции определяется числом свободных индивидных переменных.

разрешимым, предикат на множестве гёделевых номеров первопорядковых формул *Общезначима(x)* — полуразрешимым, но не разрешимым, и предикат на множестве гёделевых номеров арифметических формул *Истинна(x)* — неразрешимым.

Значение введенных понятий и конструкций заключается в следующем. Оказалось, что все попытки уточнить понятие вычислимости приводят к эквивалентным результатам. Так, понятие МНР-вычислимости эквивалентно любому другому уточненному понятию вычислимости. В частности, понятию вычислимости на так называемых машинах Тьюринга. Опираясь на этот факт, А.Чёрч выдвинул до сих пор никем не опровергнутый тезис, носящий его имя, согласно которому все, что вычислимо с интуитивной точки зрения, вычислимо и в смысле уточненного понятия вычислимости, и, наоборот, все, что вычислимо в уточненном смысле, вычислимо и интуитивно. Видимо, интуитивное понятие вычислимости сформировалось в математике и логике так, что оно не допускало двусмысленных толкований. Поэтому разнообразные и порой очень не похожие друг на друга уточнения этого понятия и оказались эквивалентными.

Следствием тезиса Чёрча является вывод: любая функция f будет вычислима тогда и только тогда, когда она МНР-вычислима. Таким образом, результаты любого вычисления могут быть повторены на МНР-машине. Например, вычисления любого физически существующего компьютера может повторить (своими средствами) МНР-машина. На самом деле МНР-компьютер мощнее всех существующих в мире компьютеров вместе взятых! Ведь он обладает бесконечной памятью и не имеет никаких ограничений по времени выполнения программ, в отличие от реальных компьютеров, существенно ограниченных по памяти и времени.

Развертывание генетической теории МНР-вычислимости состоит в эффективном построении требуемых объектов и доказательстве утверждений о них. Скажем, приведенные утверждения о связанных с разрешимостью свойствах доказываются в этой теории в виде теорем. И не только в ней. После соответствующих переформулировок аналоги этих теорем доказуемы и в других теориях классической вычислимости, поскольку при всем разнообразии подходов к эффективной вычислимости все они оказались эквивалентными в смысле совпадения определя-

емых в них классов вычислимых функций. А согласно тезису Чёрча вообще все не только имеющиеся, но и будущие теории эффективной вычислимости приведут к этому же классу. Для нас эти общеизвестные в логическом сообществе факты послужат основанием для рассмотрения вопроса о детерминизме лишь в отношении теории МНР-вычислимости, поскольку если тут есть проблемы с детерминизмом, то они так или иначе возникнут и в других эквивалентных ей теориях эффективной вычислимости.

2 Детерминированны ли МНР-вычисления?

Следуя идеям Марио Бунге, выдвинутым в книге [2], определим **детерминизм** как принцип, согласно которому все существующее, во-первых, чем-то *однозначно обусловлено* и, во-вторых, во всем *полностью определенно*. Отсюда **недетерминизм** означает либо неоднозначную обусловленность, либо неполную определенность, или и то, и другое вместе. Крайней формой недетерминизма является **индетерминизм** — допущение существования чего-либо ничем не обусловленного или полностью неопределенного.

МНР-программы $z(t_1, t_2, \ldots, t_n)\downarrow$, останавливающиеся на любом входе (любой начальной конфигурации) длины n, естественно считаются примерами *детерминированных* вычислений, однозначно ведущих к полностью определенной заключительной конфигурации за строго определенное число шагов. Программы, не отанавливающиеся на некоторых начальных конфигурациях, должны быть признаны *недетерминированными*. Порождаемые ими расходящиеся вычисления не определены в отношении их результата. Однако даже в них расходящиеся вычисления (не говоря уже о сходящихся) однозначно обусловлены начальной конфигурацией и конкретным набором команд программы. При той же начальной конфигурации и том же наборе команд на любых двух МНР-машинах будут порождаться одинаковые последовательности шагов вычислений. Различной, с классической точки зрения, может быть разве что скорость работы МНР-компьютеров, и только. Что касается *индетерминизма*, то стандартные представления об эффективной вычислимости полностью исключают этот феномен.

При решении некоторых задач (моделирование зависящих от случайностей физических процессов, творческой деятельности, игровых ситуаций и т.д.) возникает потребность в заканчивающихся недетерминированных вычислениях. На практике предпочитают обходиться лишь имитацией недетерминизма, прибегая к вычислению очередного значения последовательности псевдослучайных чисел, которые в действительности появляются друг за другом однозначно предсказуемым образом.

Рассмотрим пример программы на Паскале, которая псевдослучайным образом осуществляет перестановку множества чисел $0 \ldots 65535$. Последовательно получаем $fatum(0) = 13849$, $fatum(1) = 39022$, $fatum(2) = 64195$,..., $fatum(3915) = 0$,..., $fatum(9541) = 2, \ldots, fatum(33870) = 65535$,..., $fatum(39496) = 1$,..., $fatum(64195) = 7896$ и т.д., что создает впечатление случайности результатов (подробнее см. [4, с. 127–130]), тогда как на самом деле вычисления детерминированы и результаты однозначно предсказуемы. Разумеется, функцию $fatum$ можно воспроизвести средствам и МНР, позаботившись о том, чтобы при $i > 65535$ эта функция (как и в программе на Паскале) не имела значения.

```
var i : word; {word : 0 ... 65535}
function fatum(i : word) : word;
const
    mult = 25173;
    addi = 13849;
    modul = 65536;
begin
    fatum := (mult*i + addi)mod modul;
end;
begin
    WriteLn('Порождение псевдослучайного числа по введенному числу ');
    WriteLn('Введите −1 < i < 65536. Выход за интервал завершает программу ');
Repeat
    WriteLn;
    Readln(i);
    WriteLn(fatum(i));
```

$UntilFalse$;

end.

В рассматриваемом примере и в аналогичных случаях мы сталкиваемся с *мнимым недетерминизмом*. Другое дело, если встроить в вычислительный процесс числа, действительно появляющиеся случайно. В простейшей ситуации можно подбрасывать монету, и, отождествив цифру с нулем, а герб с единицей (или наоборот), получать случайно ноль или единицу и затем вручную вводить полученное число в компьютер. Если указанный ввод влияет на ход последующих компьютерных вычислений, мы получим действительно недетерминированный вычислительный процесс, результат которого заранее не предсказуем. В принципе ничто не мешает встроить в компьютер настоящий генератор случайных, а не псевдослучайных, чисел, автоматизировав тем самым ввод таких чисел в исполняемую программу. Описанную разновидность недетерминизма можно назвать *статистическим недетерминизмом*.

Статистический недерминизм в действительности не расширяет класс вычислимых функций. Неопределенность имеется лишь в отношении входных данных — начальная конфигурация задается случайным образом. Сами же МНР-вычисления остаются детерминированными, и только в ситуации расхождения вычислений вновь возникает недетерминизм, как об этом уже было сказано. Строго говоря, введение генератора случайных чисел в МНР-компьютер потребует расширения языка МНР-программирования, поскольку возникнет необходимость в командах, считывающих случайные числа и передающих их в программу. Но это несущественное расширение. Момент ввода в исполняемую программу случайного числа или нескольких случайных чисел следует рассматривать как формирование *новой начальной конфигурации*. Иными словами, выполнение программы $\pi(t_1, t_2, \ldots, t_n)$ может сопровождаться появлением случайных чисел только в регистрах $R_1, \ldots, R_n$. Такое появление будет означать переход к обычному детерминированному на каждом очередном шаге вычислению некоторой функции, вплоть до следующего ввода случайного числа. Обобщение понятия начальной конфигурации (например, начальные конфигурации с разрывами, начинающиеся не с первого регистра, и т. д.) приве-

дет только к лишним сложностям, но ничего в плане увеличения вычислительной мощности не даст.

В действительности статистический недетерминизм ближе к детерминизму, чем к индетерминизму. Появление в регистрах случайных чисел позволяет осуществлять хотя бы вероятностные предсказания, и в этом определенное сходство с детерминированными процессами, обеспечивающими возможность однозначных предсказаний. Но изначальная ситуация с формированием начальных конфигураций фактически *индетерминирована*, ибо неопределенна и ничем не обусловлена внутри теории МНР-вычислимости. Фигурально выражаясь, для МНР-компьютера она как бы падает с небес в виде почти что чуда. МНР-вычисление начинается только тогда, когда «чудо» непостижимым для теории образом свершится. Представим наблюдателя, следящего за работой МНР-машины. *После* того как вычисления начались, все действия компьютера детерминированы и потому однозначно предсказуемы наблюдателем. А вот *до* того ситуация недетерминирована. Если начальные конфигурации формируются с использованием генератора случайных чисел, то имеет место статистический недетерминизм и возможность осуществления наблюдателем вероятностных предсказаний. Если же формирование начальных конфигураций не обусловлено даже вероятностно, ситуация с ними полностью не определена и потому вообще не предсказуема. Здесь наблюдатель не может сказать ничего определенного и будет вынужден ждать появления неизвестно какой начальной конфигурации.

Таким образом, классическая трактовка МНР-вычислимости связана с тремя типами недетерминизма. В ней не только принимается недетерминизм в «мягкой» форме неопределенности итогового результата вычислений[3] и случайности появления начальных конфигураций, но и допускается «жесткий» недетерминизм в виде настоящего элемента индетерминизма, имеющего место в силу неопределенности и необусловленности процесса получения начальных конфигураций без генератора случайных чисел. Если бы МНР-машины были воплощены в мате-

[3]Кстати говоря, сложившееся понятие алгоритма как полностью детерминированного предписания по однозначному решению задач исключает возможность неостанавливающихся вычислений.

рии, то им был бы присущ еще один тип недетерминизма, связанный с возможностью неправильной работы: такие компьютеры могли бы завершать вычисления раньше времени, могли бы зависать, неверно выполнять команды и т.д. Однако в теории МНР-вычислимости ничего подобного не может случиться в принципе.

3 МНР-вычислимость в нестандартном универсуме

Под нестандартным универсумом понимается модель арифметики, в которой наряду с обычными конечными имеются бесконечно большие с *внешней* точки зрения натуральные числа. **Бесконечные** объекты — это те, которые не являются конечными. А какие объекты конечны? Имеется два наиболее распространенных определения конечного множества. Согласно первому определению, **конечным**$_1$ называется множество, кардинал которого является натуральным числом. Сами натуральные числа тогда естественно также считать конечными, независимо от того, являются они множествами или имеют другую природу. Согласно второму определению, множество будет **конечным**$_2$, если его нельзя взаимнооднозначно отобразить ни на какое его собственное подмножество.

Применяя первое определение конечного множества *внутри* нестандартного универсума $\mathbf{N}^*$, получаем, что множество натуральных чисел M является конечным, даже если $|M| = N$, где $N \in \mathbf{N}^* \backslash \mathbf{N}$. Будучи элементом теоретико-множественной разности между нестандартным и стандартным универсумами натуральных чисел, такое число N внешне бесконечно, но внутри нестандартного универсума — это обычное конечное натуральное число в ряду всех других натуральных чисел. Более того, множество всех подмножеств 2^M внешне бесконечного множества M с внутренней точки зрения также останется конечным, так как $\left|2^M\right| = 2^N$, где 2^N снова конечное натуральное число из $\mathbf{N}^* \backslash \mathbf{N}$. Если нестандартный универсум $\mathbf{N}^*$ внешне счетен, то внешне счетно и множество M. Тогда множество 2^M внешне несчетно. Тем не менее внутри нестандартного универсума 2^M все равно конечно в силу равенства $\left|2^M\right| = 2^N$ — различие между счетным и несчетным пропадает.

В любом случае ни M, ни 2^M нельзя средствами универсума взаимнооднозначно отобразить ни на какое собственное их подмножество, ибо для $K \subset M$, $K \neq M$ имеем в нестандартном универсуме $|K| = L$, где $L < N$. Тогда и $2^L < 2^M$. Это означает, что первое определение конечного влечет второе, точно так же, как и в стандартном универсуме. Поэтому далее в этом разделе *конечное* будет означать *конечное*$_1$.

Говоря о стандартном и нестандартном универсумах чисел, мы следовали принятому современному словоупотреблению. Однако, если считать универсум натуральных чисел изначальным (а не полученным как производный объект внутри более мощной теории, например, теории множеств), то становится не понятным, что представляет собой стандартный универсум, почему он непременно должен быть изоморфен ординалу ω из аксиоматической теории множеств ZF. С тем же основанием за исходный можно взять универсум $\mathbf{N}^*$.

Как скажется такой выбор на идее МНР-вычислимости? Очевидно, что теперь в регистрах МНР-компьютера может находиться любое число из $\mathbf{N}^*$, что ряд регистров должен быть последовательно пронумерован числами из $\mathbf{N}^*$, что начальные конфигурации могут быть любой стандартной или нестандартной длины, что в командах необходимо разрешить использовать какие угодно числа из $\mathbf{N}^*$, а число самих команд в программах также может измеряться любым числом из $\mathbf{N}^*$. Соответственно число шагов в сходящихся вычислениях должно равняться некоторому числу из $\mathbf{N}^*$, тогда как расходящиеся вычисления будут по числу шагов превышать любое наперед заданное число из $\mathbf{N}^*$.

Отсюда следует, что расходящееся вычисление «внутри» натурального ряда $\mathbf{N}^*$ невозможно, например, невозможно вычисление с бесконечным числом шагов, соответствующих стандартному ряду $1, 2, 3, \ldots$ Всякое МНР-вычисление либо завершится на некотором шаге $n \in \mathbf{N}^*$, либо продолжится за границы любого $n \in \mathbf{N}^*$. Иными словами, ряд $\mathbf{N}^*$ надлежит рассматривать как настоящий ряд натуральных чисел, а вычислимость на нем должна осуществляться без всяких индетерминированных аномалий. Это позволяет сохранить все ранее данные определения, в том числе связанные с проблемой разрешимости.

Конечно, не все так просто. Предвижу недоумения, вызванные вопросом: как все-таки проводимое последовательно, шаг за шагом (начиная с первого), вычисление может «выскочить» за границы стандартного универсума $\mathbf{N}$ и оказаться в области $\mathbf{N}^*\backslash\mathbf{N}$? Нет ли здесь явного индетерминизма, подобного индетерминизму тезиса о творении из ничего? Уклончиво отвечу, что постулированная МНР-вычислимость в универсуме $\mathbf{N}^*$ со столь необычными свойствами, по-видимому, все же не ведет к противоречиям, если принимать необходимые меры предосторожности.

Например, нет никаких оснований сомневаться, что если программа $\pi(r_1, r_2, \ldots, r_n)$ для любых $r_1, r_2, \ldots, r_n \in \mathbf{N}$ останавливается за некоторое $m \in \mathbf{N}$ шагов, то все ее действия в нестандартном универсуме будут точно такими же, как и в стандартном. А если на вход такой программы π подать числа из $\mathbf{N}^*\backslash\mathbf{N}$ — что будет тогда? Или, если вычисления по программе $\pi(r_1, r_2, \ldots, r_n)$ для конкретных $r_1, r_2, \ldots, r_n \in \mathbf{N}$ расходятся в стандартном универсуме $\mathbf{N}$, то будут ли они расходиться и в нестандартном универсуме $\mathbf{N}^*$?

Обстоятельное обсуждение теории МНР-вычислимости в нестандартных универсумах находятся за рамками данной работы. Ограничимся лишь одним аспектом этой теории, касающимся проблемы совместимости стандартных и нестандартных вычислений. Будем рассматривать только такие МНР-программы, длина которых равна n для некоторого $n \in \mathbf{N}$. Иными словами, это будут программы *стандартной длины*. Примем следующую *аксиому совместимости*.

АКСИОМА. *Для всякой програмы π стандартной длины и любых $r_1, r_2, \ldots, r_n \in \mathbf{N}$, если $\pi(r_1, r_2, \ldots, r_n) \uparrow$ в $\mathbf{N}$, то $\pi(r_1, r_2, \ldots, r_n) \uparrow$ в $\mathbf{N}^*$.*

Аксиома совместимости утверждает, что всякое расходящееся при данном входе вычисление в стандартном универсуме останется расходящимся и в нестандартном универсуме при том же входе. Это весьма правдоподобное утверждение, но можно ли его доказать? Как уже отмечалось, симметричное высказывание «Для всякой програмы π стандартной длины и любых $r_1, r_2, \ldots, r_n \in \mathbf{N}$, если $\pi(r_1, r_2, ..., r_n) \downarrow$ в $\mathbf{N}$, то $\pi(r_1, r_2, ..., r_n) \downarrow$ в $\mathbf{N}^*$» в доказательстве не нуждается, настолько оно очевидно.

Назовем предикат $P(x_1, x_2, \ldots, x_n) \subset \mathbf{N} \times \mathbf{N} \times \ldots \times \mathbf{N}$ (n раз) *стандартным*, если утверждение $< r_1, r_2, \ldots, r_n > \notin P$ означает, что $r_1, r_2, \ldots, r_n \in \mathbf{N}$. Проще говоря, на вход стандартного предиката разрешено подавать только n-ки стандартных чисел.

Между МНР-вычислимостью в стандартном и нестандартном универсумах имеется значительное различие в вычислительной мощи, как показывает следующая теорема.

ТЕОРЕМА 2. *Всякий полуразрешимый в* $\mathbf{N}$ *стандартный предикат* $P(x_1, x_2, \ldots, x_n)$ *разрешим в* $\mathbf{N}^*$.

Доказательство. Рассмотрим МНР-программу $\pi_P(x_1, x_2, \ldots, x_n, c_1, c_2, \ldots, c_m)$, вычисляющую в $\mathbf{N}$ соответствующую частичную функцию $f_P(x_1, x_2, \ldots, x_n, c_1, c_2, \ldots, c_m)$ (напомним, что $n > 0$ и $m \geq 0$). Это будет стандартная программа, имеющая стандартную длину, скажем $l \in \mathbf{N}$, в которой все константы начальных конфигураций (если они вообще имеются) являются стандартными числам и $c_1, c_2, \ldots, c_m \in \mathbf{N}$. Преобразуем программу π_P следующим образом. Во-первых, освободим от операций регистр R_{n+m+1}. Для этого в командах из π_P увеличим на единицу все упоминания о регистрах, если их номер $i \geq n + m + 1$. Получим программу π_P^*. Во-вторых, заменим в π_P^* все ссылки q в командах условного перехода на $3q$: $J(i, j, q)$ заменяется командой $J(i, j, 3q)$. Например, если в π_P^* имелась команда $J(i, j, 10)$, то в π_P^{**} она должна быть заменена командой $J(i, j, 30)$. Возникнет промежуточная программа π_P^{**}. В-третьих, найдем наибольший номер mm упоминаемого в программе π_P^{**} регистра. Затем перед каждой командой из π_P^{**} вставим две: либо $J(n + m + 1, mm + 1, 3l + 2)$ и $S(mm + 1)$, если $n + m + 1 \neq mm + 1$, либо $J(n + m + 1, mm + 2, 3l + 2)$ и $S(mm + 2)$, если $n + m + 1 = mm + 1$. В любом случае годится второй вариант $J(n + m + 1, mm + 2, 3l + 2)$ и $S(mm + 2)$, им и воспользуемся. В результате число команд вырастет втрое и станет равным $3l$. Обозначим полученную программу через π_P^{***}. Наконец, в-четвертых, допишем к концу π_P^{***} еще две команды: $3l + 1$. $J(1, 1, 0)$ и $3l + 2$. $Z(1)$. Это и будет итоговая программа π_P^{****}.

Программа π_P^{****} останавливается на любом входе $x_1, x_2, \ldots,$

$x_n, c_1, c_2, \ldots, c_m, c_{m+1}$. Действительно , из построений видно, что программа $\pi_P^{****}(x_1, x_2, \ldots, x_n, c_1, c_2, \ldots, c_m, c_{m+1})$ осуществляет вычисления по исходной программе π_P, используя разве что другие регистры. Операции условного перехода $J(n+m+1, mm+2, 3l+2)$ и увеличения на единицу $S(mm+2)$ содержимого регистра R_{mm+2} на эти исходные вычисления никак не влияют, если не достигнуто равенство $r_{n+m+1} = r_{mm+2}$. Если равенство $r_{n+m+1} = r_{mm+2}$ получено, переход к последней команде $Z(1)$ с номером $3l+2$ обнулит первый регистр и закончит вычисления. В противном случае продолжится выполнение вычислений в соответствиии с предписаниями π_P. Если эти вычисления закончатся раньше, чем будет получено равенство $r_{n+m+1} = r_{mm+2}$, в первом регистре будет содержаться единица. Таким образом, либо вычисление остановится из-за $\pi_P(x_1, x_2, \ldots, x_n, c_1, c_2, \ldots, c_m) \downarrow$, либо остановится из-за достижения равенства $r_{n+m+1} = r_{mm+2}$. Но указанное равенство будет непременно достигнуто в том случае, если имеет место $\pi_P(x_1, x_2, \ldots, x_n, c_1, c_2, \ldots, c_m) \uparrow$.

В стандартном универсуме $\mathbf{N}$ вычисления по программе $\pi_P^{****}(x_1, x_2, \ldots, x_n, c_1, c_2, \ldots, c_m, c_{m+1})$ не приведут к чему-то интересному. Другое дело нестандартный универсум $\mathbf{N}^*$. Достаточно взять в качестве константы c_{m+1} *нестандартное число* $N \in \mathbf{N}^* \backslash \mathbf{N}$, чтобы получить разрешающую предикат $P(x_1, x_2, \ldots, x_n)$ процедуру. Если n-ка $< r_1, r_2, \ldots, r_n > \in P$, то $r_1, r_2, \ldots, r_n \in \mathbf{N}$ (так как предикат P стандартен) и вычисление $\pi_P^{****}(r_1, r_2, \ldots, r_n, c_1, c_2, \ldots, c_m, N)$ закончится с результатом $r_1 = 1$ на каком-то шаге k, где $k \in \mathbf{N}$. Если же $< r_1, r_2, \ldots, r_n > \notin P$, то вновь $r_1, r_2, \ldots, r_n \in \mathbf{N}$ по причине стандартности P и $\pi_P(r_1, r_2, \ldots, r_n, c_1, c_2, \ldots, c_m) \uparrow$ в $\mathbf{N}$. В силу аксиомы совместимости $\pi_P(r_1, r_2, \ldots, r_n, c_1, c_2, \ldots, c_m) \uparrow$ верно и для $\mathbf{N}^*$. Однако вычисление $\pi_P^{****}(r_1, r_2, \ldots, r_n, c_1, c_2, \ldots, c_m, N)$ в этом случае закончится по достижению равенства $N = r_{mm+2}$, причем с $r_1 = 0$. Это и дает МНР-вычислимость искомой характеристической функции $f_P(x_1, x_2, \ldots, x_n, c_1, c_2, \ldots, c_m, N)$ стандартного предиката $P(x_1, x_2, \ldots, x_n)$, что и требовалось. Q.E.D.

В частности, в нестандартном универсуме будет разрешим упоминавшийся выше классически полуразрешимый, но не разрешимый предикат на множестве гёделевых номеров первопорядковых формул *Общезначима(x)*.

4 МНР-вычислимость в уличенном универсуме

Термин «уличенный универсум» (a witnessed universe) взят из книги [3] в переводе с английского А.Г. Драгалина. В самом общем смысле он означает большой, но все же конечный в классическом измерении универсум, который изнутри рассматривается как бесконечный. Ситуация обратная в сравнении с нестандартным универсумом: в последнем внешне бесконечное внутри оказывается конечным, а в уличенном универсуме, наоборот, внешне конечное предстает как внутренне бесконечное. Построим конкретный уличенный универсум в форме арифметики с конкретными конечными и бесконечными числами.

Пусть $L_A = \{+, \times, ^+, 0\}$ — язык арифметики. Будем говорить, что формула *арифметическая*, если все ее дескриптивные символы принадлежат языку L_A. Следующие аксиомы обычны.

1. $0 \neq x^+$ (число 0 не имеет предшествующего элемента);
2. $x^+ = y^+ \to x = y$ (функция $^+$ взаимнооднозначна);
3. $x + 0 = x$;
4. $x + y^+ = (x + y)+$ (A3 и A4 — рекурсивное определение сложения через 0 и $^+$);
5. $x \times 0 = 0$;
6. $x \times y^+ = (x \times y) + x$ (A5 и A6 — рекурсивное определение умножения через $0, ^+$ и $+$);
7. $(A(0)\ \&\ \forall x(A(x) \to A(x^+))) \to \forall x A(x)$, где A — арифметическая формула (схема аксиом индукции; пункт об арифметичности необходим, как будет видно из дальнейшего).

Следующие аксиомы сформулированы в расширенном языке $L = L_A \cup \{F, T, f\}$, где F и T — одноместные предикатные символы для выражения неарифметических свойств «быть конечным (финитным) числом» и «быть бесконечным (трансфинитным) числом», а f — функция с достаточно быстрым ростом значений.

8. $F(0)$ (разумеется, 0 — конечное число);

9. $F(x) \to F(x^+)$ (если x конечно, то и следующее за x число x^+ также конечно; поскольку А9 — не арифметическая формула, нельзя индукцией доказать, что все числа конечны);

10. $T(x) \to T(x^+)$ (аналогично А9: если x бесконечно, то и x^+ бесконечно);

11. $F(x) \to \neg T(x)$ (конечные числа не являются бесконечными; по контрапозиции $\neg\neg T(x) \to \neg F(x)$ и снятию двойного отрицания имеем $T(x) \to \neg F(x)$, т.е. бесконечные числа не конечны);

12. $F(x) \vee T(x)$ (каждое число либо конечно, либо бесконечно);

13. $f(0) = 0^{++++++++++} = 10$;

14. $f(x^+) = f(x)^{10}$, где $f(x)^{10}$ равно по определению $f(x) \times f(x) \times f(x) \times f(x) \times f(x) \times f(x) \times f(x) \times f(x) \times f(x) \times f(x)$ (заметим, что нам не потребовалось общее определение степени x^y);

15. $T(f(10))$ ($f(10)$ — конкретное бесконечное число).

Построенная теория противоречива, но противоречие достигается за огромное число шагов, которое в нашей аксиоматике считается бесконечным. Так как доказательство — это конечная последовательность формул[4], классическое выведение противоречия не является доказательством в нашем смысле, поэтому в этом понимании данная теория является непротиворечивым расширением арифметики.

Заменяя аксиомы 13-15 утверждением

13' . $\exists x(T(x))$,

получим классически непротиворечивую теорию. Действительно, пусть N — бесконечное число, существование которого утверждается, т.е. $T(N)$. Поскольку теоремами являются $\neg T(0)$, $\neg T(1), \ldots, \neg T(n), \ldots$, имеем теоремы $0 < N, 1 < N, \ldots, n < N \ldots$

[4]Отметим, что формулы, оперирующие бесконечными числами, могут оставаться конечными в смысле нашей теории, как это имеет место в аксиоме 15.

Но для каждого n цепочка утверждений $0 < N, 1 < N, \ldots, n < N$ непротиворечива. Значит, по теореме компактности, непротиворечива и вся теория. Эта теория, однако, будет ω-противоречивой ввиду теоремы $\neg\forall x \neg T(x)$.

Число $f(10)$ — это единица с десятью миллиардами нулей. Даже записать такое число вручную в одиночку невозможно. Но это возможно с помощью современной вычислительной техники. Однако на самом деле здесь мы принимаем сокращенную запись числа $f(10)$. Собственно говоря, выражение «$f(10)$» — не что иное, как еще большее сокращение (в отличие от предыдущего, легко реализуемое вручную) записи рассматриваемого числа. Каноническое представление $f(10)$ в виде $0^{+++\ldots+++}$ невозможно, поскольку потребует значков $^+$ больше, чем частиц в Метагалактике (в которой, по уверениям физиков, содержится примерно 10^{80} — 1 с 80-ю нулями — частиц ([9, с. 113])).

Да и времени для канонического представления потребуется несусветно много. В самом деле, в году примерно 31536000 секунд. Наша Метагалактика, опять же по уверениям физиков, существует примерно 10 миллиардов лет (особая точность нам в этом случае не нужна). Стало быть, от так называемого «Большого взрыва» прошло что-то около 315360×10^{12} секунд (315 квадрильонов 360 триллионов секунд). Даже если бы мы с момента этого взрыва располагали устройством, записывающим по 10^{12} (по триллиону) значков $^+$ в секунду (что само по себе невероятно), мы имели бы на сегодняшний день лишь $315360 \times 10^{12} \times 10^{12} = 315360 \times 10^{24}$ значков, то есть в десятичной записи — число 315360 с последующими 24 нулями. А нам необходимо получить десять миллиардов нулей! Сколько же еще ждать? Эти рассуждения показывают, что построенная здесь арифметика связана с космологией.

Космологические ограничения должны быть наложены и на теорию МНР-вычислимости в уличенном универсуме. В классическом случае допускалась двоякого рода бесконечность: актуально бесконечное число регистров и потенциальная бесконечность числа шагов вычислений и помещаемых в регистры чисел. Аналогичным образом, уличенная теория МНР-вычислимости в рассматриваемом конкретном варианте предполагает *внутренне актуально бесконечную* ленту регистров $R_1, R_2, R_3, \ldots$

$R_{f(10)}$. Потенциальная бесконечность чисел и шагов вычислений моделируется их *физической реализуемостью*: числа до 10^{80} будут физически реализуемы. Быть может, и гораздо большие числа, скажем, 10^{1000}, тоже найдут физическую интерпретацию. Но нет никакой надежды записать в регистры начальную конфигурацию с актуально бесконечными числами или добраться в ходе МНР-вычислений до таких чисел, хотя в нашем уличенном универсуме они все же существуют. Обозначим описанную модель уличенного универсума через **U**.

Сравнивая, как и в случае нестандартной вычислимости, уличенную вычислимость с классической, приходим к очевидному выводу, что МНР-вычислимость в **U** значительно слабее классической. Для подтверждения сказанного приведем следующую теорему.

ТЕОРЕМА 3. *Всякий разрешимый в* **N** *предикат* $P(x_1, x_2, \ldots, x_n)$ *неразрешим в* **U**.

Доказательство. Ясно, что числа в окрестности $f(10)$, не говоря уже о больших числах, даже не могут быть записаны в регистры уличенного МНР-компьютера. Q.E.D.

Например, разрешимое в **N** свойство гёделевых номеров формул логики высказываний *Тавтология(x)* будет неразрешимым в **U** по причине неограниченного роста этих номеров в **N**. Тем не менее МНР-вычислимость в **U** превосходит все мыслимые возможности реальных компьютеров.

Литература

[1] *Анисов А.М.* Вычислительная метамодель реальности и проблема истины // Логические исследования. Вып. 13. М., 2006.
[2] *Бунге М.* Причинность. Место принципа причинности в современной науке. М., 1962.
[3] *Вопенка П.* Математика в альтернативной теории множеств. М., 1983.
[4] *Грогоно П.* Программирование на языке Паскаль. М., 1982.
[5] *Девис М.* Прикладной нестандартный анализ. М., 1980.
[6] *Драгалин А.Г.* Конструктивная теория доказательств и нестандартный анализ. М., 2003.
[7] *Катленд Н.* Вычислимость. Введение в теорию рекурсивных функций. М., 1983.
[8] *Пенроуз Р.* Тени разума: в поисках науки о сознании. Москва — Ижевск, 2005.
[9] *Хокинг С.* От большого взрыва до черных дыр. М., 1990.