S©MR ISSN 1813-3304

СИБИРСКИЕ ЭЛЕКТРОННЫЕ
МАТЕМАТИЧЕСКИЕ ИЗВЕСТИЯ

Siberian Electronic Mathematical Reports
http://semr.math.nsc.ru

Том 19, стр. 144–144 (2022) УДК 510.52
DOI 10.33048/semi.2022.19.xxx MSC 03D15, 68Q09

О ВЫЧИСЛЕНИЯХ НАД УПОРЯДОЧЕННЫМИ КОЛЬЦАМИ

И.В. ЛАТКИН, А.В. СЕЛИВЕРСТОВ

Abstract. We consider generalized register machines over ordered rings with an auxiliary binary operation. In particular, we consider the ring of integers, its infinite Cartesian power, and ultrapowers. The feasibility and computational complexity of some algorithms are discussed. There is also given an example of a non-factorial ring, which is elementarily equivalent to the ring of integers. It is shown that non-deterministic computations with integers can be implemented as computations over the Cartesian power of the ring of integers. It is also possible to model calculations with an oracle using such machines. This provides an algebraic approach to describing some classes of computational complexity. However, this model of computation differs significantly from alternating machines. Moreover, various types of non-deterministic machines are considered.

Keywords: generalized register machine, ring, integral domain, integers, ultrapower, non-deterministic computations, polynomial time, oracle.

1. Введение

В последние десятилетия вопросам вычислимости и сложности вычислений над различными алгебраическими системами посвящено огромное количество работ, одно только перечисление заняло бы несколько страниц текста. При этом круг исследованных проблем также обширен. Из всего разнообразия связанных с вычислимостью тем и проблем в данной работе уделяется внимание только отдельным аспектам.

Во-первых, в качестве основной модели вычислений здесь используются обобщённые регистровые машины (ОРМ) [1, 2, 3], краткое описание которых содержится во втором разделе. Подобные ОРМ зарекомендовали себя как мощное

Latkin, I.V., Seliverstov, A.V. On computations over ordered rings.

Поступила 9 ноября 2022 г., опубликована 31 декабря 2022 г.

средство для изучения сложности вычислений над произвольными алгебраическими структурами, в первую очередь над кольцами и полями. Вычисления ОРМ над полем вещественных чисел $\mathbb{R}$ подобны вычислениям на BSS-машине [4, 5], а в случае линейно упорядоченных ассоциативных и коммутативных колец почти не отличаются от машин над списочной надстройкой Ашаева–Беляева–Мясникова [6] и S-машин Хеммерлинга [7]. Иногда рассматривают и такие машины, время работы которых бесконечно и выражается счётным ординалом [8, 9, 10]. Однако мы предполагаем, что машина, программа которой написана для вычисления какой-то всюду определённой функции, на каждом входе делает лишь конечное число шагов. Второй раздел завершается определением недетерминированных ОРМ и классов $\mathbf{P}$, $\mathbf{DNP}_I$, $\mathbf{DNP}$ и $\mathbf{NP}$.

Во-вторых, в третьем разделе изучается вычислимость некоторых естественных функций, таких как наибольший общий делитель и неполное частное, над упорядоченными кольцами в разных сигнатурах посредством обобщённых регистровых машин (ОРМ). Вполне ожидаемо получается, что различие в сигнатуре рассматриваемого кольца проявляется в различии ответов о вычислимости этих функций. Но в отличие от вычислимых нумерованных систем различие в вычислимости функций может возникать уже при добавлении всего одного константного символа, когда используются ОРМ.

В четвёртом разделе исследуется возможность моделирования обычной тьюринговой вычислимости над кольцом целых чисел $\mathbb{Z}$ посредством ОРМ над декартовой степенью этого кольца. В результате оказывается, что применение ОРМ позволяет успешно моделировать как недетерминированные вычисления, так и вычисления с оракулами над кольцом $\mathbb{Z}$. Второй из этих феноменов неформально объясняется совсем просто: для каждого подмножества кольца в декартовой степени этого кольца имеется элемент, содержащий всю информацию о данном подмножестве, поэтому можно вместо оракула использовать соответствующий ему элемент. Гораздо интереснее объяснение для первого феномена: работа ОРМ над декартовой степенью кольца представляет по сути работу многопроцессорного вычислительного устройства над этим кольцом, то есть параллельные вычисления над кольцом превращаются в вычисление над декартовой степенью.

В пятом разделе обсуждается различие классов $\mathbf{P}$, $\mathbf{DNP}_I$, $\mathbf{DNP}$ и $\mathbf{NP}$ между собой при вычислениях над некоторыми кольцами и полями.

Исследуемые кольца. Мы рассматриваем работу ОРМ над частично упорядоченным ассоциативным и коммутативным кольцом со вспомогательными бинарными операциями вычитания и rest: $(R; 0, +, -, \cdot, \mathrm{rest}, \leqslant)$, поскольку без подобного расширения вычислительные возможности таких машин весьма ограничены. В кольце целых чисел $\mathbb{Z}$ для любого x и для $y \geqslant 2$ значением $\mathrm{rest}(x, y)$ служит остаток от деления x на y из множества $\{0, \ldots, y-1\}$, а для $y \leqslant 1$ полагаем $\mathrm{rest}(x, y) = 0$. Естественно, что в других рассматриваемых кольцах тоже должно выполняться условие

$$\exists z((x = y \cdot z + \mathrm{rest}(x, y)) \wedge (0 \leqslant \mathrm{rest}(x, y) < y))$$

для $y \geqslant 2$, где под двойкой понимается сумма нейтрального элемента по умножению с самим собой, а при нарушении условия $y \geqslant 2$ (в частности, если элемент y несравнимый с аналогом двойки) не обязательно будет выполняться

равенство $\mathrm{rest}(x, y) = 0$. Тем не менее, при корректном определении этой операции, разумно потребовать выполнения более слабого свойства: для любого y верно, что $\mathrm{rest}(x, y) \geqslant 0$ и либо $\mathrm{rest}(x, y) < y$, либо $\mathrm{rest}(x, y)$ несравнимый с y. Такая ситуация возникает, например, при покомпонентном определении остатка в декартовом или прямом произведении колец, в которых эта операция уже определена (см. раздел 4 ниже).

Естественно также, что во всех рассматриваемых кольцах $\mathrm{rest}(x, y) = 0$, когда элемент x делится на y, в частности, когда элемент y обратим.

Заметим, что соблюдая все формальности, следовало бы слово остаток брать в кавычки или использовать вместо него какой-нибудь другой термин, когда оно применяется как название функции $\mathrm{rest}(x, y)$, поскольку эта функция может быть достаточно корректно определена и для не евклидовых колец, как мы это увидим далее.

Хотя во всех исследуемых далее кольцах существует нейтральный элемент по умножению, единица не включена в сигнатуру в явном виде, но когда потребуется, она будет включаться в вычисления в качестве вспомогательного входа.

Как обычно, при натуральном $m > 0$ и $a \in R$ выражение $m \cdot a$ означает сумму элемента a с самим собой m раз.

Отметим, что здесь мы ограничиваемся таким обогащением кольцевой сигнатуры, чтобы в случае совпадения кольца R с кольцом целых чисел $\mathbb{Z}$ получалось консервативное расширение теории $Th(\mathbb{Z})$, так как отношение порядка и деление с остатком определяются над $\mathbb{Z}$ формулами первого порядка. Например, неотрицательное целое число равно сумме четырёх квадратов целых чисел, среди которых могут быть равные. Кроме того, нам достаточно описанного выше тривиального определения операции $\mathrm{rest}(x, y)$ в кольце $\mathbb{Z}$ для случая $y < 2$ ввиду того, что в первую очередь нас интересует моделирование некоторых аспектов обычной тьюринговой вычислимости на натуральных числах средствами обобщённых регистровых машин, в частности, параллельных вычислений — см. раздел 4.

2. Обобщённые регистровые машины (ОРМ)

Напомним вкратце описание работы обобщённой регистровой машины над алгебраической системой $\mathfrak{A} = (A, f_1, f_2, \ldots; T_1, T_2, \ldots; c_0, c_1, \ldots)$ с основным множеством A и заданными на нём операциями f_i местности $k(i)$, предикатами T_i местности $l(i)$ и выделенными элементами c_j. Машина имеет бесконечное множество (рабочих) регистров R_i, содержащих элементы из A, и бесконечно много индексных регистров $I(n)$, содержащих натуральные числа. Константы соответствуют операциям записи соответствующего элемента в регистр. Программы представляют собой конечные списки команд, часть из которых может быть помечена (или пронумерована). Выполняя эти команды, машина может за один шаг копировать элемент из регистра $R_{I(s_1)}$, индексированного содержимым индексного регистра $I(s_1)$, и переслать его в регистр $R_{I(s_0)}$: $R_{I(s_0)} := R_{I(s_1)}$. Также машина может за один шаг применить любую операцию f_j, перечисленную в сигнатуре к элементам, содержащимися в регистрах $R_{I(n_1)}, \ldots, R_{I(n_{k(j)})}$ и записать результат в $R_{I(n_0)}$:

$$R_{I(n_0)} := f_j(R_{I(n_1)}, \ldots, R_{I(n_{k(j)})}).$$

В случае проверки на истинность сигнатурного предиката T_j, применённого к элементам, содержащихся в $R_{I(n_1)}, \ldots, R_{I(n_{l_j})}$, машина переходит в новое состояние в зависимости от его истинности, точнее к выполнению команды с соответствующим номером:

$$m: \quad T_j(R_{I(n_1)}, \ldots, R_{I(n_{l_j})}) \to \mathrm{JUMP}(k), \mathrm{JUMP}(t),$$

здесь m — номер команды, которая равносильна условному оператору

$$\mathit{if} \;\; T_j(R_{I(n_1)}, \ldots, R_{I(n_{l_j})}) \;\; \mathit{then\ goto}\ \mathrm{Lab}(k)\ \mathit{else\ goto}\ \mathrm{Lab}(t),$$

то есть в случае истинности предиката T_j, применённого к элементам, содержащихся в регистрах $R_{I(n_1)}, \ldots, R_{I(n_{l_j})}$, машина переходит к выполнению команды с меткой (номером) k, а в противном случае исполняет команду t.

Над индексными регистрами выполняются обычные операции регистровых машин. В начале работы в нулевом индексном регистре записано число регистров, занятых входными данными, а в остальных индексных регистрах записаны нули. Незанятые входными данными регистры содержат некоторый фиксированный элемент основного множества A, для частично упорядоченного кольца R там естественно записать нули.

Время работы машины полиномиальное, если существует такой многочлен $p(n)$, что если вначале ровно n регистров занято входными данными, то полное число шагов, выполняемых машиной до остановки, ограничено значением многочлена $p(n)$. Задача разрешима за полиномиальное время, если имеется ОРМ, решающая эту задачу за полиномиальное время. Класс $\mathbf{P}$ состоит из всех распознавательных задач (или соответствующих им языков), которые разрешимы за полиномиальное время. Отметим, что это определение сложности вычисления ОРМ естественно в следующем смысле: здесь набор значений аргументов $x_1, \ldots, x_n$ (элементов основного множества системы A) отождествляется со словом $x_1 \ldots x_n$ в алфавите A, таким образом, n — это просто длина входной цепочки, которая распределена по n входным регистрам.

Вычислительная сложность на рассматриваемых машинах не учитывает сложность выполнения отдельных арифметических операций, которые могут быть невычислимыми в обычном смысле. В частности, кольцо R может не быть счётным. Однако здесь учитывается время на операции над индексными регистрами. В случае, когда линейно упорядоченным кольцом R служит поле вещественных чисел $\mathbb{R}$, эти машины тесно связаны с BSS-машинами [4].

Пример 1. Рассмотрим работу обобщённой регистровой машины, вычисляющей над произвольным частично упорядоченным ассоциативным и коммутативным кольцом $(R; 0, +, -, \cdot, \mathrm{rest}, \leqslant)$ со вспомогательными бинарными операциями вычитания и rest функцию $f(x, y) = s \cdot x - y^r$, если $y^r - s \cdot x \leqslant 0$ и $f(x, y) = 0$, в противном случае, при заданных натуральных константах $r \geqslant 1$ и $s \geqslant 1$.

Итак, перед началом работы машины в регистре R_0 записан элемент x, в регистре R_1 содержится элемент y, в остальных рабочих регистрах — элемент 0 кольца R; нулевой индексный регистр $I(0)$ содержит двойку, остальные индексные регистры заняты нулями.

Первые две команды указывают значения индексов регистров, хранящих аргументы x и y; а вторая пара команд загружает в индексные регистры $I(3)$

и $I(4)$ параметры s и r, соответственно:

$$I(1) := 0, \qquad I(2) := 1,$$
$$I(3) := s, \qquad I(4) := r,$$

Заметим, что при работе с индексными регистрами вместо оператора присваивания := часто используют команды с операторами LOAD и STORE.

Далее резервируются номера регистров, в которых будут храниться результаты промежуточных вычислений, а именно, значения $s \cdot x$, y^r и $s \cdot x - y^r$:

$$I(5) := 2, \qquad I(6) := 3, \qquad I(7) := 4$$

Следующая серия команд вычисляет значение выражения $s \cdot x$, используя регистр с индексом 2, хранящимся в $I(5)$:

$R_{I(5)} := R_{I(1)}$ %*поскольку в* $I(1)$ *хранится* 0,
эта команда загружает в регистр с номером 2,
(т. е. в $R_{I(5)}$*) значение переменной* x,
хранящейся в регистре $R_{I(1)} = R_0$

1 : SUB$(3, 1)$ % *уменьшается на единицу*
значение индекса из $I(3)$

JZERO$(3 \to 2)$ %*если значение индекса из* $I(3)$*равно 0,*
то машина переходит к выполнению команды
с меткой 2 (см. ниже), а если нет,
то выполняется следующая команда

$R_{I(5)} := R_{I(5)} + R_{I(1)}$

JUMP(1)

Далее вычисляется значение выражения y^r, используя регистр с индексом 3, хранящимся в $I(6)$:

2 : $R_{I(6)} := R_{I(2)}$ %*так как в* $I(2)$ *хранится* 1,
эта команда загружает в регистр с номером 3,
(т. е. в $R_{I(6)}$*) значение переменной* y,
хранящейся в регистре $R_{I(2)} = R_1$

3 : SUB$(4, 1)$ % *уменьшается на единицу*
значение индекса из $I(4)$

JZERO$(4 \to 4)$ %*если значение индекса из* $I(4)$*равно 0,*
то машина переходит к выполнению команды
с меткой 4 (см. ниже), а если нет,
то выполняется следующая команда

$R_{I(6)} := R_{I(6)} \cdot R_{I(2)}$

JUMP(3)

Напомним, что во всех регистрах, с номерами большими 4 (т. е. начиная с R_5) первоначально были загружены нулевые элементы кольца R, но пока ни один

индексный регистр не содержит чисел больших 4, и поэтому у нас нет пока возможности использовать эти регистры. Восполним этот пробел:

$$4: \qquad I(8) := 5$$

Заканчивается работа машины вычислением значения выражения $y^r - s \cdot x$, сравнением его с нулём (т. е. с $R_{I(8)} = R_5$) и остановкой:

$$\begin{aligned} R_{I(7)} &:= R_{I(6)} - R_{I(5)} \\ R_{I(7)} \leqslant R_{I(8)} &\to \mathrm{JUMP}(6), \mathrm{JUMP}(5) \\ 5: \qquad & \mathrm{HALT}(7) \\ 6: \qquad & \mathrm{HALT}(8), \end{aligned}$$

где команды вида $\mathrm{HALT}(k)$ повелевают машине остановиться и сообщают то, что результат вычислений хранится в регистре $R_{I(k)}$.

Время работы описанной машины ограничено константой, зависящей от чисел r и s, которые служат внутренними параметрами программы, а не частью входных данных.

Замечание 1. Вслед за [1] мы не допускаем для детерминированных ОРМ команд вида $R_{I(k)} := a$, где a — отличный от сигнатурной константы элемент основного множества системы, над которой производятся вычисления. Отсюда следует, что если даже кольцо $\mathbb{Z}$ вкладывается в R и $\mathbb{Z}R$ — его образ, то при некоторых условиях на R и на входные данные $x_1, \ldots, x_n$, в регистрах машины нельзя получить некоторые (или даже никакие) элементы из $\mathbb{Z}R$ отличные от констант, при условии, что они не были заданы изначально. Пример такого сорта кольца и элементов $x_1, \ldots, x_n$ приведён в первых двух абзацах доказательства теоремы 2 ниже.

Но в тоже время в индексных регистрах могут прекрасно вычисляться любые рекурсивные (вычислимые) функции от натуральных чисел, так как в нашем распоряжении имеются функции $\mathrm{ADD}(k, a)$, $\mathrm{SUB}(k, a)$ и $\mathrm{MULT}(k, a)$, позволяющие вычислять, соответственно, результат сложения, вычитания и умножения содержимого индексного регистра $I(k)$ с числом a, а также функция $\mathrm{DIV}(k, a)$, вычисляющая целую часть от деления на a содержимого индексного регистра $I(k)$. Значит, можно находить значения и любых вычислимых функций над кольцом $\mathbb{Z}$, представляя целые числа в виде формальной разности двух натуральных. Разумеется, для этого нужно расширить список команд обобщённых регистровых машин командами вида $\mathrm{HALT}(I(k))$, которые означают, что результат вычислений содержится в индексном регистре $I(k)$, и у нас имеется возможность его прочитать. Однако при этом всё равно остаётся проблема определения, представляет ли элемент, содержащийся в данном регистре $R_{I(n)}$, аналог натурального числа, хранящегося в некотором индексном регистре $I(k)$ (предполагается, что кольцо $\mathbb{Z}$ вкладывается в R, но его подмножеством не является). В разделе 4 мы увидим, что если в качестве дополнительного входа машины разрешить задавать константу 1 или она присутствует в сигнатуре, то эта проблема благополучно разрешима, и в этом случае ОРМ могут вычислять любые рекурсивные (т. е. вычислимые в обычном смысле) функции внутри подкольца $\mathbb{Z}R$, и в том числе, раскладывать элементы из $\mathbb{Z}R$ на простые множители.

Недетерминированные ОРМ. В отличие от машин Тьюринга обобщенные регистровые машины позволяют определить несколько разных классов недетерминированных машин. Первый из этих классов получается, если в программе ОРМ содержатся команды вида

$$\to \mathrm{JUMP}(k_1), \ldots, \mathrm{JUMP}(k_t),$$

которые позволяют машине произвольным образом перейти к выполнению одной из команд с номером, содержащемся в списке $k_1, \ldots, k_t$. Этого же эффекта можно достичь, если позволить машине на каждом недетерминированном шаге записывать в индексный регистр одно из двух значений — либо нуль, либо единицу, ввиду наличия команд $\mathrm{JZERO}(k \to m)$, описанных в примере 1. Назовём такие машины *индексно-недетерминированными* ОРМ.

Когда алгебраическая система с носителем A, над которой производятся вычисления, содержит элементы 0 и 1, можно позволить машине на некоторых шагах работы недетермированно записывать любой из этих элементов также и в рабочие регистры. В обоих этих случаях недетерминированно получаемая информация может быть закодирована конечной последовательностью из нулей и единиц. Поэтому назовём такие машины *бинарно-недетерминированными* ОРМ.

Скажем, что недетерминированная ОРМ допускает данный язык $L \subseteq A^*$ за полиномиальное время, если существует такой многочлен $p(n)$, что для всякой цепочки $x_1 \ldots x_n$ из L, заполнив вначале ровно n регистров машины входными данными $x_1, \ldots, x_n$, можно найти такую ветвь вычислений, при которой полное число шагов, выполняемых машиной до остановки в допускающем состоянии, ограничено значением многочлена $p(n)$; а если $x_1 \ldots x_n \notin L$, то такой ветви вычислений не существует.

Обозначим через $\mathbf{DNP}$ класс множеств, допускаемых бинарно-недетерминированными ОРМ за полиномиальное время, а через $\mathbf{DNP}_I$ — его подкласс, который получается, если мы ограничиваемся применением только индексно-недетерминированных машин. В пятом разделе мы увидим, что эти классы различны для некоторых колец.

Отметим, что неравенство $\mathbf{P} \neq \mathbf{DNP}$ известно для так называемых линейных машин над полем вещественных чисел с операциями сложения, умножения на вещественные константы и проверкой равенства [11, 12, 13]. Также это неравенство выполняется для бесконечных абелевых групп [14], бесконечных булевых алгебр [15], (при любом $n \geqslant 2$) для кольца вещественных $n \times n$ матриц [16] и поля комплексных чисел с дополнительным предикатом для целых чисел [17]. С другой стороны, равенство $\mathbf{P} = \mathbf{DNP}$ достигается для некоторых структур [7, 18].

Ещё один тип недетерминированных ОРМ получается, если на недетерминированных шагах разрешить запись в регистры произвольного элемента алгебраической системы, над которой определена машина. ОРМ этого типа будем называть *недетерминированными общего вида*. Класс $\mathbf{NP}$ состоит из всех языков (соответствующих задач распознавания), допускаемых недетерминированными ОРМ общего вида за полиномиальное время.

Очевидны включения $\mathbf{P} \subseteq \mathbf{DNP_I} \subseteq \mathbf{DNP} \subseteq \mathbf{NP}$. В пятом разделе мы вернёмся к обсуждению, какие из этих включений собственные для кольца $\mathbb{Z}$, его декартовой степени и ультрастепени, а также для полей рациональных, вещественных и комплексных чисел.

3. Вычисления над ультрастепенью кольца $\mathbb{Z}$

Обозначим через ω множество натуральных чисел, начиная с нуля. Фиксируем некоторый ультрафильтр D, расширяющий фильтр коконечных подмножеств множества ω. В частности, для каждого подмножества множества натуральных чисел либо оно само, либо его дополнение принадлежит ультрафильтру D. Обозначим через U ультрастепень линейно упорядоченного кольца целых чисел $\mathbb{Z}$ над ультрафильтром D. Это линейно упорядоченное кольцо является областью целостности. Более того, в нём корректно определен наибольший общий делитель GCD, поскольку кольцо целых чисел $\mathbb{Z}$ и его ультрастепень $U = \prod_D \mathbb{Z}$ элементарно эквивалентны [19]. Однако U обладает необычными свойствами, невыразимыми в языке первого порядка теории частично упорядоченных колец.

3.1. **Алгебраические свойства ультрастепени U.** Элементами кольца U служат классы эквивалентности бесконечных последовательностей целых чисел $\mathbf{a} = (a_0, a_1, \dots)$. Две последовательности эквивалентны, если они совпадают на множестве индексов, принадлежащем ультрафильтру D. В частности, эквивалентны любые две последовательности, отличающиеся лишь в конечном числе позиций. Операции и отношение порядка в кольце U определяются покомпонентно. Кольцо целых чисел $\mathbb{Z}$ вложено в кольцо U, числу a соответствует класс постоянной последовательности $\mathbf{a} = (a, a, \dots)$. Каждая последовательность, содержащая нулевые элементы, либо эквивалентна последовательности из одних нулей $\mathbf{0} = (0, 0, \dots)$, либо эквивалентна последовательности, в которой нет ни одного нуля. Поэтому кольцо U не содержит делителей нуля. В кольце U ровно два обратимых элемента, а именно, классы каждой из двух постоянных последовательностей $\mathbf{1} = (1, 1, \dots)$ и $-\mathbf{1} = (-1, -1, \dots)$. Класс эквивалентности любой последовательности, состоящей из чисел 1 и -1, служит представителем обратимого элемента, но каждая из них эквивалентна либо $\mathbf{1}$, либо $-\mathbf{1}$, поскольку ультрафильтру принадлежит либо множество индексов единиц, либо множество индексов минус единиц.

Теорема 1. *Область целостности $U = \prod_D \mathbb{Z}$ не является факториальным кольцом, хотя в нём существует наибольший общий делитель любых двух ненулевых элементов $\mathbf{a}$ и $\mathbf{b}$, а также неполное частное от деления элемента $\mathbf{a}$ на $\mathbf{b} \geqslant \mathbf{1}$, то есть такой элемент $\mathbf{q}$, что $\mathbf{a} = \mathbf{b} \cdot \mathbf{q} + \operatorname{rest}(\mathbf{a}, \mathbf{b})$*

Доказательство. Для любого натурального числа m, класс последовательности элементов $\mathbf{c}_m$, k-я координата которых равна целой части $(k-m)$-й степени числа два $\lfloor 2^{k-m} \rfloor$, служит ненулевым и необратимым элементом кольца U, который делится на элемент $\mathbf{2} = (2, 2, \dots)$ без остатка, так как в каждой такой последовательности все координаты чётные. Каждый класс, содержащий какой-то элемент $\mathbf{c}_m$, не разлагается в конечное произведение неприводимых элементов кольца U. Следовательно, кольцо U нефакториальное.

Существование наибольшего общего делителя и неполного частного двух элементов, удовлетворяющих условиям теоремы, вытекает из уже отмеченной [19] элементарной эквивалентности колец $\mathbb{Z}$ и U. Однако можно дать и явное описание этих объектов, а именно, наибольший общий делитель GCD двух элементов, представителями которых служат последовательности $\mathbf{a} = (a_0, a_1, \dots)$ и $\mathbf{b} = (b_0, b_1, \dots)$, равен классу последовательности наибольших общих делителей $\operatorname{GCD}(\mathbf{a}, \mathbf{b}) = (\operatorname{GCD}(a_0, b_0), \operatorname{GCD}(a_1, b_1), \dots)$. И если все $b_i \geqslant 1$,

то неполное частное от деления элемента $\mathbf{a}$ на $\mathbf{b}$ равно

$$((a_0 - \mathrm{rest}(a_0, b_0))/b_0, (a_1 - \mathrm{rest}(a_1, b_1))/b_1, \dots).$$

□

Следствие 1. *Упорядоченное кольцо U не является ни архимедовым, ни плотным, ни евклидовым.*

Доказательство. Действительно, построенный при доказательстве теоремы элемент $\mathbf{c}_0$ не может стать меньше никакой конечной суммы вида $\mathbf{2}^t + \dots + \mathbf{2}^t$. В то же время, для всякого элемента $\mathbf{a}$ из U между ним и $\mathbf{a} + \mathbf{1}$ нет никаких элементов, что следует из элементарной эквивалентности колец $\mathbb{Z}$ и U.

Как известно, любое евклидово кольцо — факториальное, а кольцо U — не такое, следовательно, оно неевклидово. □

С другой стороны, в области целостности U операция $\mathrm{rest}(,)$ ведёт себя во многом одинаково с операцией вычисления остатка в целых числах, поэтому можно называть U слабо евклидовым кольцом.

3.2. **Алгоритмические свойства ультрастепени U.**

Теорема 2. *Наибольший общий делитель и неполное частное от деления одного элемента на другой невычислимы над кольцом $(U; 0, +, -, \cdot, \mathrm{rest}, \leqslant)$ посредством обобщённых регистровых машин.*

Доказательство. Очевидно, что элементы $\mathbf{c} = (2, 2^2, \dots, 2^k, \dots)$ и $\mathbf{1} = (1, 1, \dots)$ равны частным от деления элемента $\mathbf{a} = (2 \cdot 3, 2^2 \cdot 3^2, \dots, 2^k \cdot 3^k, \dots)$ на элемент $\mathbf{e} = (3, 3^2, \dots, 3^k, \dots)$ и на сам $\mathbf{a}$, соответственно.

Индукцией по числу шагов несложно доказывается следующее утверждение. Если на вход обобщённой регистровой машины поданы элементы $\mathbf{a}$ и $\mathbf{e}$, и на каком-то шаге её работы в одном из регистров появился какой-то элемент, то этот элемент обязательно делится и на $\mathbf{3} = (3, 3, 3, \dots)$. Следовательно, этот элемент не может быть равен ни $\mathbf{c}$, ни $\mathbf{1}$, ни вообще какому-либо аналогу целого числа, не кратного $\mathbf{3}$ — именно этот эффект имелся в виду в замечании 1, когда говорилось о невозможности вычислять целые числа посредством обобщённых регистровых машин.

Прежде чем разобраться с вычислимостью наибольшего общего делителя, установим следующее.

Лемма 1. *Пусть на вход обобщённой регистровой машине подаются элементы $\mathbf{x} = (x_0, x_1, \dots)$ и $\mathbf{y} = (y_0, y_1, \dots)$. Тогда если на каком-то шаге вычислений в одном из регистров появился (или изменился) элемент $\mathbf{z}$, то любая его k-я компонента представима как многочлен от степеней чисел x_k и y_k, если оперировать с x_k и y_k как будто это независимые переменные, а не числа, вида*

$$z_k = \sum_t l_t(k) \cdot x_k^t + \sum_s n_s(k) \cdot y_k^s + \sum_{i,j} m_{i,j}(k) \cdot x_k^i \cdot y_k^j. \tag{1}$$

В каждой из этих трёх сумм число слагаемых конечно и не зависит от k, хотя при этом для некоторых k часть из коэффициентов $l_t(k)$, $n_s(k)$, и $m_{i,j}(k)$ может равняться нулю, а при других k нет; например, может быть, что во вторую сумму входят лишь четыре слагаемых при $s = 2, 3, 6, 7$ и

$n_2(5) = n_2(6) = \ldots = n_2(15) = n_7(20) = n_7(51) = 0$, *но при всех других* k *ни* $n_2(k)$, *ни* $n_7(k)$ *не равны нулю, точно также как* $n_3(k)$ *и* $n_6(k)$ *при всех* k.

Доказательство. Индукция по количеству шагов вычисления. Перед началом вычислений после нулевого шага в регистрах имеется всего два элемента $\mathbf{x}$ и $\mathbf{y}$, имеющие нужный вид.

Пусть после нескольких шагов вычисления в регистрах машины записаны некоторые элементы, у которых каждая компонента имеет вид (1). Понятно, что операции сложения, вычитания и умножения, применённые к любой паре этих элементов, дают (после приведения подобных членов) элемент с компонентами, имеющими тот же вид. При этом количество слагаемых в каждой из трёх сумм в k-й компоненте результата не превосходит суммы количеств слагаемых в соответствующих суммах k-х компонент операндов при сложении и вычитании. При умножении количество слагаемых в первых двух суммах у k-й компоненты результата — не больше произведения количеств слагаемых в соответствующих суммах k-х компонент операндов, а число слагаемых в третьей сумме k-й компоненты результата не превосходит числа $T_1 \cdot S_2 + T_2 \cdot S_1 + T_1 \cdot J_2 + J_1 \cdot T_2 + J_1 \cdot J_2$, где T_i,S_i и J_i $(i = 1, 2)$ — количества слагаемых в первой, второй и третьей сумме, соответственно, у k-х компонент двух сомножителей.

Рассмотрим вычисление остатка двух элементов $\widehat{\mathbf{z}} = (\widehat{z}_0, \widehat{z}_1, \ldots)$ и $\tilde{\mathbf{z}} = (\tilde{z}_0, \tilde{z}_1, \ldots)$, у которых k-е компоненты имеют вид (1). Поскольку все операции производятся по-компонентно, то любая k-я компонента у $\mathrm{rest}(\widehat{\mathbf{z}}, \tilde{\mathbf{z}})$ либо равна нулю, либо имеет вид $\widehat{z}_k - q_k \cdot \tilde{z}_k$ для подходящего числа q_k, которое, вообще говоря, существенно зависит от k. Опять, после приведения подобных членов в $\widehat{z}_k - q_k \cdot \tilde{z}_k$ получается выражение вида (1). Отметим, что только при вычислении остатка могут появиться коэффициенты $l_t(k)$, $n_s(k)$ и $m_{i,j}(k)$, реально зависящие от k. □

Вернёмся к вопросу о том, каким образом можно вычислить наибольший общий делитель d двух элементов x и y какого-либо кольца. Поскольку этот d — делитель обоих данных элементов, то можно попробовать собрать его из более мелких частей элементов x и y, сначала выделив их одинаковые «простейшие» (может быть и разложимые) сомножители, а затем перемножая эти сомножители. Но при доказательстве теоремы 1 и первой части этой мы увидели, что в общем случае невозможно найти такие «простейшие» множители даже в том случае, когда они существуют.

Можно также попробовать искать наибольший общий делитель d, постепенно уменьшая элементы x и y. Надежда на это есть, поскольку в U имеется операция $\mathrm{rest}(,)$, которая ведёт себя во многом одинаково с операцией вычисления остатка в целых числах, и можно попытаться модернизировать известный алгоритм Евклида, хотя ультрастепень U и не евклидово кольцо.

Имеются также и комбинированные методы, сочетающие оба этих способа, они применяются для ускорения работы алгоритма Евклида над кольцом целых чисел. Например, для нахождения наибольшего общего делителя целых чисел x и y при $x > y$ их предварительно представляют в виде $x = x_1 \cdot 2^s + x_2$ и $y = y_1 \cdot 2^s + y_2$, где $x_2 < 2^s$, $y_2 < 2^s$ и $x_1 \leqslant (y_1)^2$ — см., например, подраздел 8.10 в [20]. Но чтобы применить этот модернизированный метод в кольце U, нужно сначала получить элемент вида $\mathbf{2}^s$, имея в регистрах лишь два произвольных элемента $\mathbf{x}$ и $\mathbf{y}$. Однако при доказательстве первой части теоремы мы убедились, что обобщённая регистровая машина над кольцом U не может

получить в общем случае из элементов $\mathbf{x}$ и $\mathbf{y}$ ни элемент $\mathbf{1}$, ни элементы вида $\mathbf{2}^s$, ни многие другие целые числа.

Итак, комбинированные способы и разложение на множители неприменимы над ультрастепенью U, поскольку она — нефакториальное кольцо. Кроме того, как мы видели в уже доказанной части теоремы, что если даже у элемента этого кольца имеются простые (в обычном смысле) делители, то нет алгоритма для их нахождения, например, нельзя вычислить элемент $\mathbf{2} = (2, 2, 2, \dots)$, имея лишь элемент $\mathbf{a}$ из доказательства первой части.

Поймём теперь, что в общем случае алгоритм Евклида в ультрастепени U не срабатывает. Для этого рассмотрим пару элементов $\mathbf{a} = (2, 3, 5, 8, 13, 21, \dots)$ и $\mathbf{b} = (1, 2, 3, 5, 8, 13, \dots)$, где $a_{k+2} = a_{k+1} + a_k$ и $b_{k+2} = b_{k+1} + b_k$. Компоненты этих элементов представляют собой два усечённых ряда чисел Фибоначчи, сдвинутых друг относительно друга на одну позицию, т. е. $a_{k+2} = b_{k+3} = b_{k+2} + b_{k+1} = b_{k+2} \cdot 1 + a_k$ и $b_{k+2} = a_{k+1} = a_k + a_{k-1} = a_k \cdot 1 + b_k$.

Отсюда получаем, что классический алгоритм Евклида даёт для этой пары элементов $\mathbf{a}$ и $\mathbf{b}$ следующие остатки: $\mathbf{r}_1 = \operatorname{rest}(\mathbf{a}, \mathbf{b}) = (0, 1, 2, 3, 5, 8, 13 \dots)$, $\mathbf{r}_2 = \operatorname{rest}(\mathbf{b}, \mathbf{r}_1) = (0, 0, 1, 2, 3, 5, 8, 13 \dots)$, $\mathbf{r}_3 = \operatorname{rest}(\mathbf{r}_1, \mathbf{r}_2) = (0, 0, 0, 1, 2, 3, 5, 8, \dots)$ и т. д. Каждый из этих остатков получается из предыдущего сдвигом на одну позицию вправо и дописыванием нуля в освободившейся крайней левой компоненте. Таким образом, за конечное число делений нулевой остаток получить не возможно, и вдобавок получается, что при любом k выполнено

$$\operatorname{GCD}(a_{k+2}, b_{k+2}) = \operatorname{GCD}(a_{k+1}, b_{k+1}) = \dots = \operatorname{GCD}(3, 2) = \operatorname{GCD}(2, 1) = 1.$$

Следовательно, $\operatorname{GCD}(\mathbf{a}, \mathbf{b}) = \mathbf{1}$.

Подойдём теперь к задаче с другой стороны, чтобы убедиться в невозможности обойтись без какой-нибудь разновидности алгоритма Евклида. Предположим, что наибольший общий делитель элементов $\mathbf{x}$ и $\mathbf{y}$ — элемент $\mathbf{d} = (d_0, d_1, \dots)$ найден, и для всех k выполнено $x_k > y_k \geqslant 1$, а значит и $d_k \geqslant 1$. Хорошо известно, что для подходящих целых чисел $u(k)$ и $v(k)$ верно равенство $u(k) \cdot x_k + v(k) \cdot y_k = d_k$. Эти множители Безу $u(k)$ и $v(k)$ находятся неоднозначно, а именно, они все имеют вид $u(k) = u_0(k) + s(k) \cdot (y_k/d_k)$, $v(k) = v_0(k) - s(k) \cdot (x_k/d_k)$, при произвольных $s(k) \in \mathbb{Z}$ и однозначно определяемых основных (базовых) множителях Безу $u_0(k)$ и $v_0(k)$ таких, что $|u_0(k)| < y_k$, а $|v_0(k)| < x_k$. Эти базовые множители могут быть найдены посредством расширенного алгоритма Евклида, производимого по классической схеме.

Таким образом, задача о вычислении $\operatorname{GCD}(\mathbf{x}, \mathbf{y})$ сводится к тому, чтобы на каком-то шаге вычисления появился элемент $\mathbf{z}$, у которого для бесконечного множества индексов k компонента z_k имела бы следующий вид

$$[u_0(k) + s(k) \cdot (y_k/d_k)] \cdot x_k + [v_0(k) - s(k) \cdot ((x_k/d_k)] \cdot y_k, \tag{2}$$

т. е. правые части равенств (1) могли бы быть преобразованы к виду (2).

Итак, всякий способ вычисления наибольшего общего делителя обобщёнными регистровыми машинами над ультрастепенью U является, по сути, некоторой модификацией алгоритма Евклида, при условии, что этот способ отличен от разложения на множители. □

Картина существенно меняется, когда имеется возможность использовать константу 1, как показывают следующие утверждения. Подчеркнём, что модель вычислимости здесь никак себя не проявляет, будь то вычислимость, задаваемая некоторым абстрактным вычислительным устройством, вроде обобщённых регистровых машин, или вычислимость, заданная подходящей нумерацией.

Пусть в области целостности R с нестрогим линейным порядком $\leqslant$ определена вычислимость таким образом, что имеются алгоритмы для вычисления сложения, вычитания и умножения, а также вычислимыми являются константа 0 и отношение порядка. В нижеследующих утверждениях предполагается выполнение всех этих свойств в кольце R.

Утверждение 1. *Из наличия алгоритма для нахождения неполного частного любых двух элементов a и $b \neq 0$ следует существование алгоритма для вычисления остатка от деления всякого элемента a на любой элемент $b \geqslant 1$.*

Доказательство. Действительно, если неполное частное от деления a на $b \geqslant 1$ равно q, то, очевидно, $\mathrm{rest}(a, b) = a - b \cdot q$. □

Утверждение 2. *Наоборот, наличие алгоритма для вычисления функции* rest*, описанной во введении, и возможность вычислять элемент* 1 *(или наличие его в сигнатуре) влечёт существование алгоритмов для выяснения обратимости любых ненулевых элементов кольца и нахождения неполного частного во многих случаях.*

Доказательство. Очевидно, что элемент $b > 0$ из R обратим тогда и только тогда, когда $\mathrm{rest}(1, b) = 0$, а если $b < 0$, то нужно вычислить $\mathrm{rest}(1, -b)$.

Предположим, что каким-то образом можно вычислять единицу и остаток от деления всякого элемента a на любой $b \neq 0$ (или имеется способ нахождения их номеров). Чтобы вычислить неполное частное q от деления данного элемента $a > 0$ на известный элемент $b > 1$, вычислим элемент

$$c = a - \mathrm{rest}(a, b)$$

и рассмотрим три основных случая.

Если $c < b^2 - b$ и элемент $b - 1$ необратимый, то, учитывая $c = b \cdot q$, получаем $q < b - 1$ и $c = (b - 1)q + q$. Следовательно, $q = \mathrm{rest}(c, b - 1)$.

Когда $b^2 < c$, тогда $b < q$. Но элемент $c \cdot b + 1$ всё же больше, чем q, так как $b > 1$, а значит и $b^2 > 1$; если к тому же этот элемент необратимый, то $q = \mathrm{rest}(-c^2, c \cdot b + 1)$, поскольку $-c^2 = (c \cdot b + 1) \cdot (-q) + q$.

При выполнении условия $b^2 - b < c < b^2$, равносильного $b - 1 < q < b$, имеем $-c = (b + 1) \cdot (-q) + q$ и $q = \mathrm{rest}(-c, b + 1)$ при необратимом элементе $b + 1$.

В крайних случаях $c = b^2 - b$ или $c = b^2$ понятно, что $q = b - 1$ или $b = q$, соответственно.

При $a < 0$ поступаем почти симметричным образом. □

Теорема 3. *Вычисление неполного частного от деления любого элемента* **a** *на элемент* $\mathbf{b} > 0$ *производится подходящей обобщённой регистровой машиной над кольцом* $(U; 0, +, -, \cdot, \mathrm{rest}, \leqslant)$ *за время, ограниченное константой, если на вход машины подавать не только эти элементы, но также и запись элемента* **1** *в регистре.*

Доказательство. Напомним, что в кольце U ровно два обратимых элемента $\mathbf{1}$ и $-\mathbf{1}$, и из того, что $\mathbf{b} > \mathbf{0}$ и $\mathbf{b} \neq \mathbf{1}$ следует, что $\mathbf{b} \geqslant \mathbf{2}$, поэтому полностью применим алгоритм, описанный при доказательстве утверждения 2. □

Замечание 2. Введённое при доказательстве утверждения 2 ограничение $b > 1$ представляется существенным, а именно, при $0 < b < 1$ случай $b^2 < c$ оказывается самым сложным.

Действительно, чтобы при $c < b^2 - b$ была возможность делить на $b - 1 < 0$, можно определить, что неполное частное от деления элемента $x > 0$ на $y < 0$ равно неполному частному от деления элемента $-x$ на $-y$.

Если $0 < b < 1$, $b^2 < c$ и было найдено такое , что $m \cdot b^2 \geqslant 1$, то получается неравенство $m{\cdot}c{\cdot}b{+}1 = q{\cdot}m{\cdot}b^2{+}1 > q$, и поскольку $-m{\cdot}c^2 = (m{\cdot}c{\cdot}b{+}1){\cdot}(-q){+}q$, то $q = \mathrm{rest}(-m \cdot c^2, m \cdot c \cdot b + 1)$ при необратимом $m \cdot c \cdot b + 1$. Но что делать, когда такого m нет? И как вообще узнать о том, что оно есть?

Таким образом, поиск неполного частного для произвольной области целостности R представляет собой крайне сложную задачу, когда $0 < b < 1$. На этот вывод наводит изучение делимости в кольце U_2, которое получено как ультрастепень кольца рациональных чисел, знаменатели которых в несократимой записи суть степени двойки

$$\mathbb{Q}(2) = \{m/2^t \mid m \in \mathbb{Z}, t \in \omega\},$$

по ультрафильтру D из начала этого раздела. Определим неполное частное q от деления числа x на $y > 0$ в кольце $\mathbb{Q}(2)$ как число вида $s/2$, где s — наименьшее целое со свойством $y \cdot (s+1)/2 \geqslant x$, а $\mathrm{rest}(x, y) = x - y \cdot q$. Рассмотрим в кольце U_2 множество элементов $\mathbf{b}(M)$ вида $(t_0/2^{r_0}, t_1/2^{r_1}, \ldots, t_k/2^{r_k}, \ldots)$, где последовательность $M = (\langle t_0, r_0\rangle, \langle t_1, r_1\rangle, \ldots, \langle t_k, r_k\rangle, \ldots)$ пар натуральных чисел имеет единственное ограничение $(\forall k \in \omega)(t_k \leqslant 2^{r_k})$. Складывается впечатление, что общего алгоритма для вычисления посредством обобщённых регистровых машин неполного частного от деления элементов $\mathbf{a} \geqslant \mathbf{2}$ на произвольный элемент вида $\mathbf{b}(M) \leqslant \mathbf{1}$ над кольцом $(U_2; 0, +, -, \cdot, \mathrm{rest}, \leqslant)$ не существует, даже тогда, когда на вход машины наряду с элементами $\mathbf{a}$ и $\mathbf{b}(M)$ подаётся элемент $\mathbf{1}$.

3.3. **Сложность вычисления наибольшего общего делителя.** Принятая для обобщённых регистровых машин оценка вычислительной сложности оказывается неудобной, когда на вход подаётся одно число, в этом случае сложность с любой оценивающей функцией f совпадает с константной, точнее равна $O(f(1))$. Хотя при работе с многочленами, рациональными функциями или матрицами эта оценка лучше соответствует обычному понятию сложности, поскольку при их задании вводятся несколько чисел.

Но в общем случае полиномиально ограниченное число арифметических операций нельзя выполнить за полиномиальное время на обычных машинах Тьюринга из-за возникновения неожиданно больших чисел.

Пример 2. Рассмотрим вычисление наибольшего общего делителя (в кольце $\mathbb{Q}[x]$) двух многочленов с целыми коэффициентами от одной переменной на обобщённой регистровой машине над кольцом $\mathbb{Z}$. Пусть каждый многочлен задан набором коэффициентов, включая нулевые. Тогда запись одного многочлена степени d занимает $d{+}1$ регистров. Алгоритм Евклида требует линейного от суммы степеней числа операций. Однако возникающие на промежуточных шагах коэффициенты могут иметь очень большую длину записи [21, 22, 23, 24, 25],

что значительно увеличивает время вычислений при использовании многоленточных машин Тьюринга.

Чтобы преодолеть эту трудность, для вычисления наибольшего общего делителя многочленов применяются другие алгоритмы. Метод, основанный на вычислении субрезультантов, предложил Дж. Сильвестр (J.J. Sylvester). Потом этот метод улучшали Вальтер Габихт (Walter Habicht) [24] и Алкивиадис Акритас (Alkiviadis Akritas) [25]. Более эффективен алгоритм, который предложил Уильям Браун (William Brown) [26].

4. Вычисления над декартовой степенью кольца $\mathbb{Z}$

Перейдём к вычислениям над декартовой степенью $\mathbb{Z}^\omega$ кольца целых чисел, с покомпонентным определением сигнатурных операций и отношения порядка. Это кольцо имеет мощность континуума. Далее отождествим кольцо целых чисел $\mathbb{Z}$ с образом диагонального вложения в $\mathbb{Z}^\omega$, когда целое число m отождествляется с постоянной последовательностью. Кроме очевидного наличия делителей нуля и несравнимых элементов, у кольца $\mathbb{Z}^\omega$ имеются и другие существенные отличия от кольца U. Например, в кольце $\mathbb{Z}^\omega$ для любого элемента $\mathbf{c} = (c_0, c_1, c_2, \ldots)$ между ним и элементом $\mathbf{c} + \mathbf{1} = (c_0+1, c_1+1, c_2+1, \ldots)$ имеется бесконечно много попарно несравнимых друг с другом элементов. Тем не менее, порядок в кольце $\mathbb{Z}^\omega$ всё же неплотный, так как между двумя элементами этого кольца, у которых проекции на все множители, кроме одного, одинаковые, а особая координата второго элемента на единицу больше соответствующей проекции у первого, ничего нет, но первый элемент меньше второго. В этом кольце также наблюдается эффект, отмеченный во введении: остаток от деления на элемент, у которого проекции на собственную часть множителей — минус единицы, а остальные проекции — положительные, может быть несравнимым с делителем.

Говоря неформально, мы покажем, что, подавая обобщённой регистровой машине над $\mathbb{Z}^\omega$ на вход числа из $\mathbb{Z}$, можно моделировать параллельные вычисления с ограниченным обменом между процессорами. Выигрыш может быть достигнут, если позволить машине использовать внутренние параметры из $\mathbb{Z}^\omega$, которые не принадлежат кольцу $\mathbb{Z}$, как дополнительные входы. Среди таких дополнительных входов-параметров, наряду с элементом $\mathbf{1} = (1, 1, \ldots)$, будем использовать элемент, обозначаемый через $\mathbf{d} = (0, 1, 2, \ldots) \in \mathbb{Z}^\omega$, проекция которой на k-й декартов множитель равна k.

Замечание 3. В дальнейшем для краткости, рассматривая вычисления над кольцами $\mathbb{Z}^\omega$ или U, мы будем говорить, что на вход машины подан постоянный элемент $\mathbf{a}$ (или элемент-константа), подразумевая элемент вида $(a, a, \ldots)$ или класс эквивалентности, в котором он лежит, на основании следующей леммы. Именно об этом говорилось в конце замечания 1. В формулировке леммы говорится о двоичной системе счисления, однако понятно, что её доказательство проходит для позиционной системы с любым основанием.

Лемма 2. *Пусть в одном из регистров ОРМ имеется элемент* $\mathbf{1} = (1, 1, \ldots)$. *Тогда по имеющейся записи элемента* $\mathbf{k} = (k, k, \ldots) \in \mathbb{Z}^\omega, U$ *можно найти запись в индексных регистрах представления числа* k *в двоичной системе счисления за сублинейное время от величины* k. *Наоборот, если в индексных*

регистрах имеется запись представления целого числа k в двоичной системе счисления или в одном из индексных регистрах записано само это число, то можно за линейное время от числа индексных регистров, содержащих цифры числа k или от величины k, соответственно, вычислить элемент $\mathbf{k} = (k, k, \ldots) \in \mathbb{Z}^\omega$.

Доказательство. Покажем, что вычисление записи двоичного представления натурального числа k в индексных регистрах по данному элементу $\mathbf{k}$ кольца $\mathbb{Z}^\omega$ или U осуществимо за $O(\log k)$ шагов. Действительно, вычисление неполного частного от деления элемента $\mathbf{t} \leqslant \mathbf{k}$ на $\mathbf{2} = \mathbf{1} + \mathbf{1}$ и сравнение остатка от этого деления с элементом $\mathbf{1}$ (чтобы узнать какое число записывать в очередной индексный регистр) осуществимо за время, ограниченное некоторой константой, согласно алгоритму, описанному при доказательстве утверждения 2. Этот алгоритм здесь применим, так как все необходимые вычисления производятся внутри образа кольца $\mathbb{Z}$ при его диагональном вложении в $\mathbb{Z}^\omega$, т. е. в некоторой области целостности. Однако можно поступить по-иному, чтобы узнать неполное частное от деления $\mathbf{k}$ на $\mathbf{2}$, для этого достаточно сделать следующее. Подряд сравниваем число $\mathbf{k}$ со степенями $\mathbf{2}^t$ до тех пор, пока не найдётся значение, превосходящее $\mathbf{k}$; число шагов при этом равно $t \leqslant 1 + \log_2(1 + k))$; затем находим двоичное представление числа k — это сумма некоторых степеней $\mathbf{2}^t$. Но число таких сравнений само равно $O(\log_2 k)$, поэтому второй способ немного более трудоёмкий. Обратное утверждение очевидно. □

Пример 3. Рассмотрим критерий простоты числа, который основан на малой теореме Ферма: целое число $p \geqslant 2$ простое тогда и только тогда, когда для каждого $x \in \mathbb{Z}$ выполнено равенство $x^p \equiv x \pmod p$. Этот критерий лежит в основе вероятностного теста Рабина–Миллера для проверки простоты натурального числа в рамках обычной тьюринговой вычислимости, когда для достаточно большого количества натуральных чисел проверяется сравнение $x^p \equiv x \pmod p$. Однако использование ОРМ над кольцом $\mathbb{Z}^\omega$, позволяет создать уже детерминированный тест для такой проверки.

Вместо перебора чисел x из $\mathbb{Z}$ можно запустить обобщённую регистровую машину над $\mathbb{Z}^\omega$ на независящих от входа $\mathbf{p}$ последовательностях $\mathbf{d} = (0, 1, 2, 3, \ldots)$ и $\mathbf{1}$. Целое число $p \geqslant 2$ простое тогда и только тогда, когда

$$\mathbb{Z}^\omega \models \mathbf{d}^p \equiv \mathbf{d} \,(\mathrm{mod}\, \mathbf{p}).$$

Проверка этого условия завершается за конечное число шагов над $\mathbb{Z}^\omega$. В самом деле, остаток от деления на $\mathbf{p} \in \mathbb{Z}^\omega$ вычисляется за один шаг; для этого в сигнатуре предусмотрен функциональный символ rest. Возведение в степень $p \in \omega$ требует $O(\log p)$ умножений, если нам известно это натуральное число. Но поскольку нам дано лишь $\mathbf{p} \in \mathbb{Z}^\omega$, то предварительно мы ищем число p, используя элемент $\mathbf{1}$, встроенные в машину операции и часть индексных регистров для хранения цифр в двоичном представлении числа p, как это описано в доказательстве леммы 2. На это тратится тоже $O(\log p)$ действий.

При этом на вход подаётся только три элемента $\mathbf{p}$, $\mathbf{1}$ и $\mathbf{d}$. А число шагов зависит от значения числа p и может быть сколь угодно большим. Поэтому работа ОРМ не завершается за полиномиальное время относительно количества входных регистров, которых всего только три. Но время работы машины — линейное по отношению к величине числа p.

Отметим, что задача об определении простоты элемента может быть очень сложной в смысле вычислимости по Тьюрингу, даже в случае вычислимой области целостности с однозначным разложении на множители. Например, в работе [27] строится кольцо $A(Q)$ с этими свойствами, содержащее в качестве подкольца кольцо целых чисел $\mathbb{Z}$, в котором при любом вычислимом представлении указанная задача имеет сложность вхождения в наперёд заданное Π^0_2-множество Q.

Напомним, что множество X принадлежит классу $\mathcal{NP}$ (в рамках обычной вычислимости по Тьюрингу), если существует такой алгоритм и многочлены $f(n)$ и $g(n)$, что для каждого элемента $x \in X$ длины n существует сертификат y (то есть набор некоторых параметров, создаваемых как правило не детерминировано, которые позволяют подтвердить принадлежность x множеству X) длины $m \leqslant f(n)$, при котором этот алгоритм допускает пару $\langle x, y\rangle$ за время не превосходящее $g(n+m)$, а для каждого $x \notin X$ такого сертификата не существует. Множество (или язык) X соответствует задаче распознавания. Это определение легко переносится и на обобщённые регистровые машины и при этом получается те классы **NP**, **DNP** и $\mathbf{DNP}_I$, что описаны в конце второго раздела. Но в этом разделе мы будем рассматривать класс $\mathcal{NP}$ в обычном смысле и во избежание недоразумений будем использовать разные шрифты для этих классов.

Множество X из класса $\mathcal{NP}$ называется $\mathcal{NP}$-полным, если каждое множество из класса $\mathcal{NP}$ сводится по Карпу к множеству X. Примером служит множество таких линейных диофантовых уравнений от многих переменных, что каждое из этих уравнений имеет некоторое $(0,1)$-решение [28, 29, 30]. Коэффициентами уравнений служат обычные целые числа. Эту задачу можно интерпретировать и следующим образом. Можно ли среди нескольких целых чисел, которые задаются в качестве коэффициентов диофантова уравнения, выбрать такие, что их сумма равна данному числу — противоположному к свободному члену уравнения? Поэтому для краткости, мы будем называть задачу распознавания множества X *задачей о сумме подмножества*. Строго говоря, следовало бы говорить о мультимножестве и его подмножестве, поскольку среди коэффициентов диофантова уравнения могут быть равные.

Теорема 4. *Задача о сумме подмножества над $\mathbb{Z}$ разрешима за детерминированное полиномиальное время на обобщённой регистровой машине над $\mathbb{Z}^\omega$, использующей элементы* $\mathbf{1}$ *и* $\mathbf{d}$.

Доказательство. Машина получает на вход набор постоянных элементов $\mathbf{a}_0$, $\mathbf{a}_1, \ldots, \mathbf{a}_n$ из кольца $\mathbb{Z}^\omega$, которые соответствуют целым коэффициентам линейного уравнения $a_0 + a_1x_1 + \cdots + a_nx_n = 0$ и элементы $\mathbf{1}$ и $\mathbf{d}$. Число регистров, занятых входом, равно $n+3$. Это число $n+3$ записано в нулевом индексном регистре. Работа машины состоит из трёх этапов. Реализация и сложность каждого из них описывается ниже.

Сначала машина вычисляет n различных простых натуральных чисел p_1, $\ldots$, p_n, точнее соответствующие элементы из кольца $\mathbb{Z}^\omega$, а также запись в $O(n \log n)$ индексных регистрах цифр при двоичном представлении этих чисел.

Потом машина вычисляет n элементов $\mathbf{b}_1, \ldots, \mathbf{b}_n$ из $\mathbb{Z}^\omega$, для каждого из которых все проекции на декартовы множители равны 0 или 1. При этом каждая

из 2^n комбинаций из n нулей и единиц должна (бесконечное число раз) реализоваться как набор проекций $(b_{1,k}, \ldots, b_{n,k})$ элементов $\mathbf{b}_1, \ldots, \mathbf{b}_n$ на некоторый множитель с номером k.

Наконец, вычисляется значение $\mathbf{c} = (\mathbf{a}_0 + \mathbf{a}_1 \cdot \mathbf{b}_1 + \cdots + \mathbf{a}_n \cdot \mathbf{b}_n)^2$ и за один шаг проверяется условие $\mathbf{1} \leqslant \mathbf{c}$, означающее положительность проекции элемента $\mathbf{c}$ на каждый множитель. При выполнении этого условия вход отвергается, и вход принимается, когда ответ отрицательный.

Оценим сложность этого алгоритма. При $x \geqslant 17$ количество простых чисел на отрезке $[1, x]$ удовлетворяет неравенству [31]

$$\pi(x) \geqslant \frac{x}{\ln x}.$$

Поэтому для выбора простых чисел $p_1 = 2$, $p_2 = 3$, $\ldots$, p_n и соответствующих им постоянных элементов достаточно проверить $O(n \log n)$ чисел. Проверка каждого из них выполняется за время, ограниченное многочленом от n, при использовании теста, описанного в примере 3. Разумеется, проверку на простоту этих чисел за полиномиальное время можно осуществить и непосредственно либо перебирая в индексных регистрах числа, не превосходящие испытываемого числа $t \leqslant O(n \log n)$, а затем восстанавливая постоянный элемент $\mathbf{t}$, либо сразу испытывая элемент $\mathbf{t}$. Но этот способ непосредственной проверки немного более трудоёмкий.

Для каждого индекса $k \leqslant n$ вычислим $\mathbf{b}_k = \mathrm{rest}(\mathbf{d}^{p_k - 1}, \mathbf{p_k})$, где через $\mathbf{d}$ обозначен фиксированный элемент $(0, 1, 2, \ldots) \in \mathbb{Z}^\omega$. Проекция k-го элемента $\mathbf{b}_k$ на m-й множитель равна нулю, когда p_k делит m, иначе она равна единице. В частности, $\mathbf{b}_1 = (0, 1, 0, 1, 0, 1, \ldots)$ и $\mathbf{b}_2 = (0, 1, 1, 0, 1, 1, 0, \ldots)$. В силу Китайской теоремы об остатках, любой набор нулей и единиц реализуется как набор проекций $(b_{1,t}, \ldots, b_{n,t})$ элементов $\mathbf{b}_1, \ldots, \mathbf{b}_n$ на некоторый множитель с номером t. Например, при любом $\mathbf{b}_k$ его проекции $b_{k,0} = 0$ и $b_{k,1} = 1$. Поэтому проекции на нулевой и первый множители дают набор из нулей и набор из единиц, соответственно. Наконец, вычисление $\mathbf{c} = (\mathbf{a}_0 + \mathbf{a}_1 \cdot \mathbf{b}_1 + \cdots + \mathbf{a}_n \cdot \mathbf{b}_n)^2$ выполняется за $O(n)$ операций. □

Из теоремы 4 не следует, что за полиномиальное время разрешима задача, аналогичная задаче о сумме подмножества, когда коэффициентами уравнений служат элементы из кольца $\mathbb{Z}^\omega$.

Неформальное объяснение теоремы 4 состоит в том, что недетерминированное вычисление над $\mathbb{Z}$ превращается в параллельное вычисление на неограниченном числе копий кольца $\mathbb{Z}$, которыми служат проекции декартовой степени на множители. Такая модель соответствует многопроцессорному вычислительному устройству с ограниченным обменом данными между процессорами, что существенно отличает эту модель от альтернирующих машин.

Отметим трудную задачу, которую не удаётся решить, используя обобщённые регистровые машины над $\mathbb{Z}^\omega$. Непонятно, можно ли за полиномиальное время, используя $\mathbf{d} = (0, 1, 2, \ldots)$, найти число решений задачи о сумме подмножества. Или хотя бы проверить за полиномиальное время, что это число решений равно наперёд угаданному числу.

Если позволить использовать не только элемент $\mathbf{d}$, проекции которого легко вычислимы, но и произвольные наперёд заданные элементы, то можно реализовать вычисление с оракулом.

Теорема 5. *Задача распознавания целых чисел, принадлежащих фиксированному непустому множеству $X \subset \mathbb{Z}$ разрешима за конечное время на обобщённой регистровой машине над $\mathbb{Z}^\omega$, использующей элемент $\mathbf{1}$ и элемент $\mathbf{f}$, определяемый множеством X.*

Доказательство. Фиксируем сюръективное отображение $\nu : \omega \to X$. Зададим проекцию элемента $\mathbf{f}$ на k-й декартов множитель равной $\nu(k)$. Машина получает на вход три элемента $\mathbf{1}$, $\mathbf{f}$ и $\mathbf{n}$, соответствующий числу $n \in \mathbb{Z}$. Это число принадлежит множеству X тогда и только тогда, когда нарушается условие $\mathbf{1} \leqslant (\mathbf{n} - \mathbf{f})^2$. Это условие проверяется за конечное число операций над $\mathbb{Z}^\omega$. □

5. Недетерминированные вычисления над некоторыми кольцами и полями

Напомним, что задача допустима посредством недетерминированной ОРМ, если для всякого частного случая задачи найдётся такая цепочка вычислений машины, которая даёт утвердительный ответ, если он присущ данному частному случаю, а в противном случае такой цепочки нет.

Существенным обстоятельством при доказательстве теорем 7–9 является то, что при вычислениях на ОРМ (как детерминированных, так и не детерминированных) полиномиальное время работы совпадает с константным, когда на вход машины подаётся только один элемент — см. подраздел 3.3 и раздел 2.

5.1. **Вычисления над кольцами с операцией** rest**.**

Теорема 6. *Класс $\mathbf{DNP}_I$ — собственный подкласс класса* **DNP** *при вычислениях над ультрастепенью $(U, 0, +, -, \cdot, \mathrm{rest}, \leqslant)$.*

Доказательство. Рассуждения, приведённые в первых двух абзацах доказательства теоремы 2, дословно проходят и для индексно-недетерминированных ОРМ. Таким образом, задача о нахождении неполного частного от деления элемента $\mathbf{a}$ на элемент $\mathbf{b} > \mathbf{0}$ над упорядоченным кольцом U не разрешима с использованием таких машин. С другой стороны, бинарно-недетерминированная машина может уже на первом недетерминированном шаге записать в каком-то регистре элемент $\mathbf{1}$. После проверки того, что вновь появившийся элемент — это именно $\mathbf{1}$, а не $\mathbf{0}$ (напомним, что других элементов бинарно-недетерминированные ОРМ записывать не могут), эта машина уже полностью детерминировано может найти искомое неполное частное, согласно тереме 3.

Теперь сформулируем распознавательную задачу. Даны три элемента $\mathbf{a}$, $\mathbf{b} > 0$ и $\mathbf{c} > 0$ кольца U, требуется определить делится ли неполное частное от деления $\mathbf{a}$ на $\mathbf{b}$ — элемент $\mathbf{q}$, на $\mathbf{c}$. Для положительного ответа на этот вопрос необходимо, чтобы элемент $\mathbf{a}_1 = \mathbf{a} - \mathrm{rest}(\mathbf{a}, \mathbf{b})$ делился на $\mathbf{c}$, т. е. когда $\mathrm{rest}(\mathbf{a}_1, \mathbf{c}) = 0$. Если это так, элемент $\mathbf{q}$ может как делиться на $\mathbf{c}$, так может и не делиться, даже тогда, когда $\mathbf{b}$ не делится на $\mathbf{c}$. Узнать какой из этих случаев для элемента $\mathbf{q}$ имеет место невозможно, используя индексно-недетерминированные ОРМ, поскольку в общем случае он невычислим посредством таких машин, а значит, остаётся неизвестным, о нём лишь известно, что его произведение с $\mathbf{b} > 0$ равно $\mathbf{a}_1$ и потому он меньше $\mathbf{a}_1$. В этой связи напомним, что кольцо U нефакториальное и неевклидово, в частности, может быть, что никакая степень элемента $\mathbf{b}$ не превосходит элемента $\mathbf{a}_1$.

В то же время, согласно теореме 3 бинарно-недетерминированная машина может вычислить неполное частное $\mathbf{q}$ за время ограниченное константой и после проверки равенства $\mathrm{rest}(\mathbf{q}, \mathbf{c}) = 0$ дать определённый ответ. □

Отметим, что при доказательстве этой теоремы был установлен более сильный факт по сравнению с формулировкой теоремы: существуют задачи, у которых для любого частного случая положительный или отрицательный ответ может быть найден бинарно-недетерминированными ОРМ за фиксированное число шагов, но эти ответы невозможно получить для некоторых частных случаев ни за какое время, используя индексно-недетерминированные машины.

Теорема 7. *Класс* **DNP** *— собственный подкласс класса* **NP** *при вычислениях над ультрастепенью* $(U, 0, +, -, \cdot, \mathrm{rest}, \leqslant)$*, а также над* $(U, 0, 1, +, -, \cdot, \mathrm{rest}, \leqslant)$*, кольцом целых чисел и его декартовой степенью.*

Доказательство. Рассмотрим следующую задачу. Дан постоянный элемент $\mathbf{a} > 0$, требуется понять, имеется ли такой элемент $\mathbf{k} > 0$, что $\mathbf{a}$ делится на $\mathbf{k}^2$, но не делится на $\mathbf{k}^3$ (в случае кольца $\mathbb{Z}$ подразумеваем, что эти элементы — просто целые числа a и k, записанные в рабочих регистрах, при этои они могут благополучно сосуществовать с их записями в индексных).

Сначала убедимся, что для ультрастепени U такой элемент-сертификат $\mathbf{k}$, если он имеется для данного $\mathbf{a}$, должен быть постоянным. Действительно, число a имеет конечное количество всевозможных делителей. Пусть $d_1, d_2, \ldots, d_n$ — это все такие делители числа a, которые встречаются в качестве проекций на декартовы множители у элемента $\mathbf{k}$ и такие, что каждый d_j^2 делит a, но его не делит d_j^3. Поскольку предполагается, что $\mathbf{k}$ является подтверждающим сертификатом для рассматриваемой задачи, то этот список делителей не пуст. Более того, множество индексов проекций у элемента $\mathbf{k}$, на которых записано хоть одно число из этого списка принадлежит ультрафильтру D. Когда $n = 1$, тогда элемент $\mathbf{k}$ постоянный. Предположим, что $n > 1$. Пусть теперь $I(d_j))$ — множество тех индексов проекций, в которых записан делитель d_j. Из определения сразу вытекает, что эти множества попарно не пересекаются. Если множество $I(d_1)$ лежит в ультрафильтре D, тогда $\mathbf{k}$ — постоянный элемент. А когда $I(d_1)$ — не подмножество в D, тогда его дополнение — подмножество в D, и значит, объединение всех остальных $I(d_j)$ принадлежит ультрафильтру и можно множество $I(d_1)$ исключить из рассмотрения. Продолжая далее укорачивать список индексных множеств приходим к ситуации, когда в нём остаётся только одно множество.

Теперь покажем, что в случае декартвой степени $\mathbb{Z}^\omega$ среди всех подтверждающих сертификатов этой задачи, если они существуют для данного $\mathbf{a}$, обязательно найдётся постоянный элемент. Рассмотрим любой из этих сертификатов $\mathbf{k}$. Пусть $d_1, d_2, \ldots, d_n$ — это все такие делители числа a, которые встречаются в качестве проекций на декартовы множители у элемента $\mathbf{k}$ и такие, что каждый d_j^2 делит a. С другой стороны, любая координата сертификата $\mathbf{k}$ должна быть равна одному из этих d_j, и среди последних обязательно найдётся такой d_s, у которого d_s^3 не делит число a. Тогда $\mathbf{d_s} = (d_s, d_s, d_s, \ldots)$ — искомый постоянный сертификат.

Проведённый анализ позволяет построить бинарно-недетерминированную ОРМ, допускающую рассматриваемую задачу для любого из этих колец. Это, например, можно реализовать следующим образом.

Вначале машина заполняет единицей один из индексных регистров. Затем она недетерминированным образом заполняет нулями и единицами ещё несколько следующих индексных регистров. Количество заполняемых индексных регистров тоже определяется недетерминировано, например, так. Перед заполнением очередного индексного регистра, начиная со второго (в первом заполняемом уже стоит единица), машина вначале недетерминировано пишет там либо нуль, либо единицу; если записался нуль, то машина прекращает заполнение и переходит к детерминированному окончанию, считая предыдущий регистр последним заполненным, а когда записалась единица, тогда машина повторно недетерминировано пишет там нуль или единицу и переходит к заполнению следующего индексного регистра. Разумеется, при этом вычислении в одном, специально отведённом для этого, индексном регистре, хранится номер первого заполненного регистра, а в следующем — либо номер последнего, либо количество заполненных индексных регистров. Набор нулей и единиц в заполненном таким недетерминированным образом массиве индексных регистров рассматривается как двоичная записью некоторого числа $k > 0$.

Затем машина детерминированно вычисляет элементы $\mathbf{k}^2$ и $\mathbf{k}^3$ и сравнивает остатки $\mathrm{rest}(\mathbf{a}, \mathbf{k}^2)$ и $\mathrm{rest}(\mathbf{a}, \mathbf{k}^3)$ с нулём. Если нужный элемент $\mathbf{k}$ существует, то найдётся такая ветвь вычислений машины, которая обнаруживает его, а именно, после подтверждения того, что $\mathrm{rest}(\mathbf{a}, \mathbf{k}^2) = 0$ и $\mathrm{rest}(\mathbf{a}, \mathbf{k}^3) > 0$, эта машина допускает вход $\mathbf{a}$. При этом на вычисление величин $\mathbf{k}^2$ и $\mathbf{k}^3$ с использованием двоичной записи числа k требуется $O(\log k)$ операций и на создание самой записи в индексных регистрах нужно примерно столько же действий. Поскольку на вход подаётся всего один элемент, а время работы машины зависит от величины числа k, то описанный процесс не является полиномиальным.

Вполне возможно, что бинарно-недетерминированная ОРМ может находить подтверждающий сертификат $\mathbf{k}$ и каким-то иным способом при условии его существования, но в любом случае время вычисления будет зависеть от числа k и потому будет переменной величиной, а не фиксированной.

В то же время, недетерминированная машина общего вида может сразу записать в одном из рабочих регистров элемент $\mathbf{k}$, а затем осуществить проверку, истинности утверждений, что $\mathrm{rest}(\mathbf{a}, \mathbf{k}^2) = 0$ и $\mathrm{rest}(\mathbf{a}, \mathbf{k}^3) > 0$. □

5.2. **Вычисления над полями.** Поскольку все ненулевые элементы любого поля суть обратимые, то согласно описанию свойств функции rest из первого раздела, эта функция тождественно равна нулю, и поэтому нет смысла включать её в сигнатуру.

При доказательстве следующих двух теорем по сути обыгрывается тот простой факт, что при вычислениях над полями рациональных, вещественных или комплексных чисел бинарно-недетерминированная ОРМ не может получить в регистрах никаких иных чисел кроме рациональных, если ей на вход поданы только рациональные числа.

Теорема 8. *Классы* **DNP** *и* **NP** *различны при вычислениях посредством ОРМ над полем рациональных чисел как с линейным порядком* $(\mathbb{Q}, 0, 1, +, -, \cdot, \leqslant)$*, так и без него* $(\mathbb{Q}, 0, 1, +, -, \cdot)$.

Доказательство. Рассмотрим такую задачу: определить извлекается ли из данного рационального числа $a > 0$ квадратный корень. Предположим, что

для некоторого $a \in \mathbb{Q}$ его квадратный корень — рациональное число. Это означает, что для подходящих целых чисел r и s верно равенство $a = (r/s)^2$. Поэтому имеется бинарно-недетерминированная ОРМ, допускающая вход a. Эта машина может действовать следующим способом. Она вначале недетерминировано создаёт в индексных регистрах двоичные записи чисел r и s, как это описано при доказательстве теоремы 7, а затем проверяет верность равенства $a \cdot s^2 = r^2$ и в зависимости от его истинности допускает или отвергает вход. Время работы этой машины — сублинейное относительно величины $r + s$.

Время работы любой бинарно-недерминированной ОРМ для решения рассматриваемой задачи также должно меняться с ростом суммы $r + s$, следовательно, время не может быть полиномиальным.

Однако недетерминированная ОРМ общего вида уже на первом недетерминированном шаге вычислений может записать в одном из регистров некоторое рациональное число k и убедившись, что $k^2 = a$, допустить вход a. □

Поля действительных и комплексных чисел рассматриваются с дополнительным одноместным предикатом Int, выделяющим целые числа, а именно, в сигнатуре $(0, 1, +, -, \cdot, \mathrm{Int})$, При этом в поле вещественных чисел $\mathbb{R}$ можно ещё добавить линейный порядок $\leqslant$.

Теорема 9. *Класс* **DNP** *собственный подкласс в* **NP** *при вычислениях посредством ОРМ над полями действительных чисел* $(\mathbb{R}, 0, 1, +, -, \cdot, \mathrm{Int})$ *и* $(\mathbb{R}, 0, 1, +, -, \cdot, \leqslant, \mathrm{Int})$*, а также над полем комплексных чисел* $(\mathbb{C}, 0, 1, +, -, \cdot, \mathrm{Int})$.

Доказательство. Немного изменим задачу из доказательства предыдущей теоремы: рациональное ли число — квадратный корень для данного положительного $a \in \mathbb{Q}$?

Предположим, что для некоторого $a \in \mathbb{Q}$ существуют целые чисел r и s, для которых верно равенство $a = (r/s)^2$. Тогда опять, для любой бинарно-недерминированной ОРМ число шагов вычисления пары целых чисел (r, s), подтверждающего сертификата для этого частного случая задачи, — переменная величина, зависящая от $r+s$, и поэтому не является полиномиальным. Хотя имеется небольшое достоинство вычислений, генерирующих двоичные записи чисел r и s в индексных регистрах — не используется предикат Int.

Подобная генерация для недетерминированной ОРМ общего вида не нужна. Она может сразу недетерминировано записать в рабочих регистрах какие-то два числа r_1 и s_1, убедиться, что они целые, а затем после проверки истинности равенства $a \cdot s_1^2 = r_1^2$ допустить или отвергнуть вход a. □

Отметим, что доказательство обеих теорем этого подраздела не изменится, если в сигнатуры исследуемых полей ввести операцию деления, доопределив, что результат деления на нуль равен, например, нулю.

References

[1] E. Neumann, P. Pauly, *A topological view on algebraic computation models*, Journal of Complexity, **44** (2018), 1–22. https://doi.org/10.1016/j.jco.2017.08.003

[2] A.V. Seliverstov, *Heuristic algorithms for recognition of some cubic hypersurfaces*, Programming and Computer Software, **47** (2021), 50–55. https://doi.org/10.1134/S0361768821010096

[3] A.V. Seliverstov, *Binary solutions to large systems of linear equations*, Prikladnaya Diskretnaya Matematika, no. 52 (2021), 5–15. https://doi.org/10.17223/20710410/52/1

[4] L. Blum, M. Shub, S. Smale, *On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines*, Bulletin of the American Mathematical Society, **21**:1 (1989), 1–46. https://doi.org/10.1090/S0273-0979-1989-15750-9

[5] L. Blum, F. Cucker, M. Shub, S. Smale, *Complexity and Real Computation*, Springer, New York, 1998.

[6] I.V. Ashaev, V.Ya. Belyaev, A.G. Myasnikov, *Toward a Generalized Computability Theory*, Algebra and Logic, **32**:4 (1993), 185–205. https://doi.org/10.1007/BF02261744

[7] A. Hemmerling, *Computability of string functions over algebraic structures*, Mathematical Logic Quarterly **44**:1 (1998), 1–44. https://doi.org/10.1002/malq.19980440102

[8] P. Koepke, A.S. Morozov, *Characterizations of ITBM-computability. I*, Algebra and Logic, **59**:6 (2021), 423–436. https://doi.org/10.1007/s10469-021-09622-2

[9] P. Koepke, A.S. Morozov, *Characterizations of ITBM-computability. II*, Algebra and Logic, **60**:1 (2021), 26–37. https://doi.org/10.1007/s10469-021-09625-z

[10] M. Carl, *Taming Koepke's Zoo II: Register machines*, Annals of Pure and Applied Logic, **173**:3 (2022), 103041. https://doi.org/10.1016/j.apal.2021.103041

[11] K. Meer, *A note on a $P \neq NP$ result for a restricted class of real machines*, Journal of Complexity **8**:4 (1992), 451–453. https://doi.org/10.1016/0885-064X(92)90007-X

[12] P. Koiran, *Computing over the reals with addition and order*, Theoretical Computer Science **133**:1 (1994), 35–47. https://doi.org/10.1016/0304-3975(93)00063-B

[13] F. Cucker, M. Matamala, *On digital nondeterminism*, Mathematical Systems Theory **29** (1996), 635–647. https://doi.org/10.1007/BF01301968

[14] C. Gaßner, *The P-DNP problem for infinite Abelian groups*, Journal of Complexity **17**:3 (2001), 574–583. https://doi.org/10.1006/jcom.2001.0583

[15] M. Prunescu, *$P \neq NP$ for all infinite Boolean algebras*, Mathematical Logic Quarterly **49**:2 (2003), 210–213. https://doi.org/10.1002/malq.200310020

[16] A. Rybalov, *On the P-NP problem over real matrix rings*, Theoretical Computer Science **314** (2004), 281–285. https://doi.org/10.1016/j.tcs.2003.11.022

[17] A.N. Rybalov, *Relativizations of the $P = NP$ problem over the complex number field*, Siberian Electronic Mathematical Reports **1** (2004), 91–98.

[18] A. Hemmerling, *P=NP for some structures over the binary words*, Journal of Complexity **21**:4 (2005), 557–578. https://doi.org/10.1016/j.jco.2005.02.001

[19] C.C. Chang, H.J. Keisler, *Model Theory*, Elsevier, 1990.

[20] A.V. Aho, J.E. Hopcroft, and J.D. Ullman, *The design and analysis of computer algorithms*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1976.

[21] P.E. Alaev, V.L. Selivanov, *Fields of algebraic numbers computable in polynomial time. I*, Algebra and Logic, **58**:6 (2020), 447–469. https://doi.org/10.1007/s10469-020-09565-0

[22] P.E. Alaev, V.L. Selivanov, *Fields of algebraic numbers computable in polynomial time. II*, Algebra and Logic, **60**:6 (2022), 349–359. https://doi.org/10.1007/s10469-022-09661-3

[23] A. Sinhababu, T. Thierauf, *Factorization of polynomials given by arithmetic branching programs*, Computational complexity, **30**:15 (2021), 1–47. https://doi.org/10.1007/s00037-021-00215-0

[24] W. Habicht, *Eine Verallgemeinerung des Sturmschen Wurzelzählverfahrens*, Commentarii Mathematici Helvetici, **21** (1948), 99–116. https://doi.org/10.1007/BF02568028

[25] A.G. Akritas, *Elements of Computer Algebra with Applications*, John Wiley and Sons, NY, 1989.

[26] W.S. Brown, *The subresultant PRS algorithm*, ACM Transactions on Mathematical Software, **4**:3 (1978), 237–249. https://doi.org/10.1145/355791.355795

[27] D.D. Dzhafarov, J.R. Mileti, *The Complexity of Primes in Computable Unique Factorization Domains*, Notre Dame Journal of Formal Logic, **59**:2 (2018), 139–156. https://doi.org/10.1215/00294527-2017-0024

[28] K. Koiliaris, C. Xu, *Faster pseudopolynomial time algorithms for subset sum*, ACM Transactions on Algorithms, **15**:3 (2019), 40. https://doi.org/10.1145/3329863

[29] A.V. Seliverstov, *On binary solutions to systems of equations*, Prikladnaya Diskretnaya Matematika, no. 45 (2019), 26–32. https://doi.org/10.17223/20710410/45/3

[30] T. Alon, N. Halman, *Strongly polynomial FPTASes for monotone dynamic programs*, Algorithmica, **84** (2022), 2785–2819. https://doi.org/10.1007/s00453-022-00954-8

[31] P. Dusart, *Explicit estimates of some functions over primes*, The Ramanujan Journal, **45** (2018), 227–251. https://doi.org/10.1007/s11139-016-9839-4

Ivan Vasilyevich Latkin
D. Serikbayev East Kazakhstan Technical University,
Protozanov Street, 69, Ust-Kamenogorsk, 070004, The Republic of Kazakhstan
E-mail address: lativan@yandex.kz

Alexandr Vladislavovich Seliverstov
Institute for Information Transmission Problems
of the Russian Academy of Sciences,
Bolshoy Karetny, 19, Moscow, 127051, Russia
E-mail address: slvstv@iitp.ru