Math-Net.Ru

Общероссийский математический портал

С. С. Марченков, О сложности класса $\mathcal{E}^2$ Гжегорчика, *Дискрет. матем.*, 2010, том 22, выпуск 1, 5–16

DOI: <https://doi.org/10.4213/dm1081>

Использование Общероссийского математического портала Math-Net.Ru подразумевает, что вы прочитали и согласны с пользовательским соглашением
<http://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 3.85.73.92

26 апреля 2023 г., 09:47:36

О сложности класса $\mathcal{E}^2$ Гжегорчика

© 2010 г. С. С. Марченков

Определяется серия абстрактных вычислительных устройств, в терминах которых можно охарактеризовать класс $\mathcal{E}^2$ иерархии Гжегорчика. Для каждого из этих устройств устанавливается время вычисления функций из класса $\mathcal{E}^2$.

Работа выполнена при поддержке Российского фонда фундаментальных исследований, проект 09–01–00701.

В 1953 г. А. Гжегорчик [11] предложил ставшую впоследствии широко известной классификацию множества примитивно рекурсивных функций. Из всех классов $\mathcal{E}^n$ этой классификации особо выделяются классы $\mathcal{E}^2$ и $\mathcal{E}^3$. Класс $\mathcal{E}^3$ — это класс функций, элементарных по Кальмару, а класс $\mathcal{E}^2$, еще более элементарный, чем класс $\mathcal{E}^3$, вполне можно считать ключевым для всей классификации. Так, класс $\mathcal{E}^2$ содержит большинство арифметических функций, используемых в широкой математической практике (исключение составляют, пожалуй, некоторые функции экспоненциального роста). Класс $\mathcal{E}^2$ имеет различные машинные описания, и эти описания легко распространяются на большие классы иерархии Гжегорчика. Наконец, класс $\mathcal{E}^2$ имеет конечный базис по суперпозиции, и этот результат также нетрудно перенести на классы $\mathcal{E}^n$ при $n > 2$. Однако уже для классов $\mathcal{E}^0$ и $\mathcal{E}^1$ ни положительные, ни отрицательные результаты о конечной базисуемости пока не получены.

В этой статье мы хотим обратить внимание на сложность вычисления функций, образующих класс $\mathcal{E}^2$ Гжегорчика. Как правило, под сложностью вычисления функций понимают временную либо емкостную сложность вычисления для надлежаще выбранного типа вычислительных устройств. Так сложилось, что для получения основных результатов по классу $\mathcal{E}^2$ использовались машины Минского либо вычислительные устройства, близкие к машинам Минского [6, 7]. Другие вычислительные устройства в этих целях практически не рассматривались. Между тем, за прошедшие 40 лет были предложены новые вычислительные устройства, пригодные для изучения класса $\mathcal{E}^2$, и кроме того, обнаружилось, что можно с успехом применять некоторые уже известные устройства.

Ниже мы определим ряд вычислительных устройств, с использованием которых можно задать класс $\mathcal{E}^2$ Гжегорчика и установим, что для каждого из этих устройств время вычисления функций из класса $\mathcal{E}^2$ либо не превосходит некоторого полинома от значений аргументов, либо, что то же самое, — некоторой экспоненты от длины двоичной записи аргументов. Часть из этих результатов уже известна в теории рекурсивных функций, другие результаты публикуются впервые. В заключение статьи обсуждается один результат С. А. Волкова [3], который связан с классом $\mathcal{E}^2$ и его описанием в терминах машин Минского.

Мы начнем с индуктивного определения класса $\mathcal{E}^2$, предложенного А. Гжегорчиком, и его упрощенного задания с помощью монотонной ограниченной рекурсии.

Обозначим через $I_i^n(x_1, \dots, x_i, \dots, x_n)$ селекторную функцию (проекцию), совпадающую со значениями переменной x_i . Пусть I есть множество всех селекторных функций I_i^n , $1 \leq i \leq n$, $n = 1, 2, \dots$. Положим

$$x \div y = \max(x - y, 0).$$

Будем говорить, что всюду определенная функция $f(x_1, \dots, x_n, y)$ получена из всюду определенных функций $g(x_1, \dots, x_n)$, $h(x_1, \dots, x_n, y, z)$, $j(x_1, \dots, x_n, y)$ с помощью операции ограниченной рекурсии, если выполняются соотношения

$$\begin{aligned} f(x_1, \dots, x_n, 0) &= g(x_1, \dots, x_n), \\ f(x_1, \dots, x_n, y + 1) &= h(x_1, \dots, x_n, y, f(x_1, \dots, x_n, y)), \\ f(x_1, \dots, x_n, y) &\leq j(x_1, \dots, x_n, y). \end{aligned} \quad (1)$$

Класс $\mathcal{E}^2$ Гжегорчика — это наименьший класс функций, который содержит исходные функции 0 , $x + 1$, $x \cdot y$, I , замкнутый относительно операций суперпозиции и ограниченной рекурсии.

Индукцией по построению функций в классе $\mathcal{E}^2$ несложно показать, что всякая функция из $\mathcal{E}^2$ мажорируется подходящим полиномом с натуральными коэффициентами. Более того, нетрудно понять, что при любом естественном способе вычисления любую функцию из $\mathcal{E}^2$ можно вычислить за время, также ограниченное некоторым полиномом с натуральными коэффициентами (полином рассматривается от аргументов исследуемой функции).

Будем говорить, что ограниченная рекурсия (1) является монотонной ограниченной рекурсией, если получаемая в результате функция $f(x_1, \dots, x_n, y)$ монотонно не убывает по переменной y .

Обозначим через $\mathcal{E}_{\text{мон}}^2$ наименьший класс функций, который содержит исходные функции

$$x + 1, \quad x \div y, \quad x \cdot y, \quad I \quad (2)$$

и замкнут относительно операций суперпозиции и монотонной ограниченной рекурсии.

Предложение 1. *Справедливо равенство*

$$\mathcal{E}^2 = \mathcal{E}_{\text{мон}}^2.$$

Доказательство. Хорошо известно [9, 11], что функция $x \div y$ принадлежит классу $\mathcal{E}^2$. Поэтому $\mathcal{E}_{\text{мон}}^2 \subseteq \mathcal{E}^2$. Установим обратное включение.

Суперпозициями функций системы (2) нетрудно получить функции

$$0, \quad \text{sg}(x), \quad \overline{\text{sg}}(x), \quad x + y, \quad (3)$$

где

$$\begin{aligned} \text{sg}(x) &= \begin{cases} 0, & \text{если } x = 0, \\ 1 & \text{в противном случае,} \end{cases} \\ \overline{\text{sg}}(x) &= \begin{cases} 1, & \text{если } x = 0, \\ 0 & \text{в противном случае.} \end{cases} \end{aligned}$$

Имея в классе $\mathcal{E}_{\text{мон}}^2$ функции (2) и (3), можно заключить [9, 11], что класс $\mathcal{E}_{\text{мон}}^2$ замкнут относительно операции разбора случаев по предикатам. Нас далее будут интересовать лишь предикаты вида

$$\begin{aligned} g_1(x_1, \dots, x_n) &< g_2(x_1, \dots, x_n), \\ g_1(x_1, \dots, x_n) &\leq g_2(x_1, \dots, x_n). \end{aligned}$$

В частности, если функция f определяется соотношениями

$$f(x_1, \dots, x_n) = \begin{cases} f_1(x_1, \dots, x_n), & \text{если } g_1(x_1, \dots, x_n) \leq g_2(x_1, \dots, x_n), \\ f_2(x_1, \dots, x_n) & \text{в противном случае,} \end{cases}$$

то с использованием функций (2), (3) ее можно задать так:

$$\begin{aligned} f(x_1, \dots, x_n) &= f_1(x_1, \dots, x_n) \cdot \overline{\text{sg}}(g_1(x_1, \dots, x_n) \div g_2(x_1, \dots, x_n)) \\ &\quad + f_2(x_1, \dots, x_n) \cdot \text{sg}(g_1(x_1, \dots, x_n) \div g_2(x_1, \dots, x_n)). \end{aligned}$$

Применяя эти соображения, определим в классе $\mathcal{E}_{\text{мон}}^2$ функции $[x/y]$ и $\text{rm}(x, y)$, где

$$\begin{aligned} [x/y] &= \begin{cases} \text{целой части от деления } x \text{ на } y, & \text{если } y > 0, \\ x, & \text{если } y = 0, \end{cases} \\ \text{rm}(x, y) &= \begin{cases} \text{остатку от деления } x \text{ на } y, & \text{если } y > 0, \\ x, & \text{если } y = 0. \end{cases} \end{aligned}$$

Полагая

$$\begin{aligned} h(x, y, 0) &= 0, \\ h(x, y, z+1) &= \begin{cases} z+1, & \text{если } y(z+1) \leq x, \\ h(x, y, z) & \text{в противном случае,} \end{cases} \\ h(x, y, z) &\leq z, \end{aligned}$$

получим, что

$$\begin{aligned} [x/y] &= h(x, y, x), \\ \text{rm}(x, y) &= x \div [x/y] \cdot y. \end{aligned}$$

Теперь сведем ограниченную рекурсию (1) к монотонной рекурсии. С этой целью добавим новую переменную w , которая будет компенсировать возможное убывание функции f по переменной y . Набор переменных $x_1, \dots, x_n$ далее для краткости обозначим через $\tilde{x}$. Положим

$$\begin{aligned} f'(\tilde{x}, w, 0) &= \begin{cases} g(\tilde{x}), & \text{если } g(\tilde{x}) < w, \\ 0 & \text{в противном случае,} \end{cases} \\ f'(\tilde{x}, w, y+1) &= \begin{cases} h(\tilde{x}, y, [f'(\tilde{x}, w, y)/w]) + w, & \text{если } h(\tilde{x}, y, [f'(\tilde{x}, w, y)/w]) < w, \\ 0 & \text{в противном случае,} \end{cases} \\ f'(\tilde{x}, w, y) &\leq (y+1)w. \end{aligned}$$

Если теперь выбрать полином $P(\tilde{x}, y)$ с натуральными коэффициентами, который удовлетворяет соотношению

$$\max\{f(\tilde{x}, 0), f(\tilde{x}, 1), \dots, f(\tilde{x}, y)\} < P(\tilde{x}, y),$$

то получим, что

$$f(\tilde{x}, y) = \text{rm}(f'(\tilde{x}, P(\tilde{x}, y), y), P(\tilde{x}, y)).$$

Предложение доказано.

Как отмечалось, любую функцию из класса $\mathcal{E}^2$ можно вычислить за полиномиальное (относительно значений аргументов) время. Одним из вычислительных устройств, которые в этом смысле удобны для вычисления функций класса $\mathcal{E}^2$, являются машины Минского [5, 10].

Машина Минского представляет собой вариант нестирающих машин Тьюринга. Машина Минского $\mathcal{M}$ имеет конечное число односторонних лент, содержимое которых не меняется в процессе вычисления. Концевые клетки лент содержат символ 1, все остальные клетки — символ 0. На каждой из лент находится одна читающая головка. В процессе работы машины $\mathcal{M}$ на каждом шаге вычисления любая из головок может независимым образом сдвигаться на одну клетку влево, вправо или оставаться в той же клетке. Функционирование машины $\mathcal{M}$ определяется программой, которая состоит из конечного числа команд вида

$$a_1 \dots a_k q_i \rightarrow q_j d_1 \dots d_k,$$

где k — число лент машины $\mathcal{M}$, $a_1, \dots, a_k$ — символы 0 или 1, обозреваемые головками машины $\mathcal{M}$ на лентах, и $d_1, \dots, d_k$ — символы движения головок на лентах ($d_1, \dots, d_k \in \{-1, 0, 1\}$). Программа машины $\mathcal{M}$ организована так, что головки машины $\mathcal{M}$ не могут сходить с концевых клеток лент.

Машина $\mathcal{M}$ вычисляет (частичную) функцию $f(x_1, \dots, x_n)$, если $k \geq n$, в начальный момент вычисления первые n головок машины $\mathcal{M}$ находятся в клетках с номерами $x_1, \dots, x_n$ (концевые клетки имеют номер 0), а в заключительный момент вычисления (если значение $f(x_1, \dots, x_n)$ определено) первая головка находится в клетке с номером $f(x_1, \dots, x_n)$.

В [7] (см. также [8]) доказано следующее утверждение.

Предложение 2. *Класс $\mathcal{E}^2$ состоит в точности из всех всюду определенных функций, которые можно вычислить на машинах Минского за полиномиальное (относительно значений аргументов) время.*

Существует несколько модификаций машины Минского и соответственно имеется несколько утверждений, аналогичных предложению 2. Мы рассмотрим только одну модификацию — машину Минского без внутренней памяти (сокращенно машина Минского БВП). Из самого названия видно, что машина БВП не имеет внутренних состояний. Функция переходов машины БВП (которую в данном случае правильнее было бы назвать функцией движений) имеет вид

$$\{0, 1\}^k \rightarrow \{-1, 0, 1\}^k,$$

где входной и выходной векторы размерности k интерпретируются так же, как в случае обычной машины Минского. Другое отличие машины БВП от машины Минского состоит в том, что некоторые векторы из множества $\{0, 1\}^k$ объявляются заключительными, и

машина БВП завершает работу, как только обозреваемый ее k головками набор входных значений становится заключительным вектором.

Техническая сторона моделирования обычной машины Минского $\mathcal{M}$ машиной Минского БВП $\mathcal{M}'$ довольно проста: если машина $\mathcal{M}$ имеет r внутренних состояний, то машина $\mathcal{M}'$ по сравнению с машиной $\mathcal{M}$ имеет дополнительно $l = \lceil \log_2 r \rceil$ лент, на которых записываются (в двоичном коде) номера состояний моделируемой машины $\mathcal{M}$. Можно условиться, например, что кодом начального состояния машины $\mathcal{M}$ является единичный вектор длины l (напомним, что концевые клетки лент машины $\mathcal{M}$ содержат символ 1), а заключительными векторами машины $\mathcal{M}'$ являются все векторы длины $k + l$, у которых последние l разрядов образуют код заключительного состояния машины $\mathcal{M}$. Дальнейшие технические детали моделирования машины $\mathcal{M}$ машиной $\mathcal{M}'$ очевидны и мы опускаем их описание.

Для машин Минского БВП справедливо следующее утверждение.

Предложение 3. *Класс $\mathcal{E}^2$ состоит в точности из всех всюду определенных функций, которые можно вычислить на машинах Минского без внутренней памяти за полиномиальное время.*

Еще один тип вычислительных устройств для описания класса $\mathcal{E}^2$ (а также многих других классов) предложен в работе [1] (см. также [9]). Эти вычислительные устройства носят название стековые регистровые машины (сокращенно машины SRM).

Память машины SRM состоит из входных регистров, которые будем обозначать $x_1, \dots, x_n$, регистров стека $t_0, t_1, \dots, t_k$, рабочего регистра r и нулевого регистра 0. В каждом регистре, за исключением регистра 0, на каждом шаге работы машины может находиться любое число $x \geq 0$. В нулевом регистре постоянно содержится число 0.

Работа машины определяется программой, которая состоит из конечного числа команд. Каждая команда в программе снабжена некоторой меткой, при этом разные команды помечены различными метками. Команды могут быть одного из следующих трех типов:

- (1) $r := z$, где z имеет вид x_i, t_j, r или 0;
- (2) $t_i := t_i + 1$;
- (3) $z_1 = z_2$, то переходим к L_i , иначе к L_j , где z_1, z_2 имеют вид x_p, t_q, r или 0, а L_i, L_j — метки программы.

Команда (1) носит название команды засылки. При ее выполнении содержимое регистра z засылается в регистр r . Содержимое всех регистров, отличных от регистра r , при этом не меняется. После выполнения этой команды машина переходит к выполнению следующей (согласно нумерации меток) команды.

Команда (2) называется командой приращения. При ее выполнении содержимое регистра t_i увеличивается на 1. Одновременно содержимое всех регистров t_j , где $j < i$, обращается в 0. Содержимое остальных регистров, отличных от регистров $t_0, \dots, t_i$, не меняется. После выполнения команды (2) машина переходит к выполнению следующей команды программы.

Команда (3) является командой перехода. При ее выполнении происходит проверка равенства содержимого регистров z_1, z_2 , и машина переходит либо к команде с меткой L_i , либо к команде с меткой L_j . При выполнении команды (3) содержимое всех регистров остается неизменным.

Программа машины SRM представляет собой список команд C_i , помеченных различными метками L_i : $(L_1 : C_1; L_2 : C_2; \dots; L_s : C_s)$. Программы машин SRM подчиняются

двум ограничениям. Во-первых, последняя команда программы не есть команда перехода. Во-вторых, для всякого i в программе машины встречается не более одного вхождения команды $t_i := t_i + 1$.

В начале работы машины SRM в рабочем регистре и регистрах стека находятся нули. Нулевой регистр, как отмечалось, постоянно содержит 0. Если на машине вычисляется n -местная функция, то число входных регистров машины равно n , и в начале вычисления в регистры $x_1, \dots, x_n$ помещаются значения соответствующих переменных. В дальнейшем содержимое входных регистров не меняется. Работа машины начинается с команды, помеченной меткой L_1 , и заканчивается после выполнения команды, помеченной меткой L_s . При этом результатом работы считается содержимое рабочего регистра после выполнения команды C_s .

Пусть $\mathcal{M}$ — машина SRM, имеющая n входных регистров. Результат работы машины над входными значениями $x_1, \dots, x_n$ будем обозначать через $\mathcal{M}(x_1, \dots, x_n)$. Зоной вычисления $\text{SPACE}_{\mathcal{M}}(x_1, \dots, x_n)$ машины $\mathcal{M}$ на входе $x_1, \dots, x_n$ считаем наибольшее из значений всех регистров, достигнутое в процессе вычисления значения $\mathcal{M}(x_1, \dots, x_n)$. Пусть $f(x), g(x_1, \dots, x_n)$ — всюду определенные функции. Говорим, что функция g вычислима на машине $\mathcal{M}$ в пределах зоны f , если

$$\begin{aligned} g(x_1, \dots, x_n) &= \mathcal{M}(x_1, \dots, x_n), \\ \text{SPACE}_{\mathcal{M}}(x_1, \dots, x_n) &\leq f(\max(x_1, \dots, x_n)). \end{aligned}$$

Множество всех всюду определенных функций, вычисляемых на машинах SRM в пределах зоны f , обозначим через $\text{SRM}(\text{SPACE}, f)$.

Следующее утверждение представляет собой частный случай общей теоремы, доказанной в работе [1].

Предложение 4. *Имеет место равенство*

$$\mathcal{E}^2 = \bigcup_i (\text{SRM}(\text{SPACE}), (x^2 + 1)^{(i)}),$$

где $(x^2 + 1)^{(i)}$ — i -кратная суперпозиция функции $x^2 + 1$.

Нетрудно понять, что время вычисления произвольной функции из класса $\mathcal{E}^2$ на машине SRM ограничено подходящим полиномом (от значений аргументов функции).

Обратимся теперь к вычислительным устройствам, в которых аргументы вычисляемых функций и значения функций представляются в двоичном коде. Рассмотрим сначала самый простой вариант машины Тьюринга — одноленточную машину Тьюринга. Считаем, что ленточный алфавит машины Тьюринга состоит из символов $\Lambda, 0, 1$ (Λ — пустой символ). При вычислении функций на машине Тьюринга аргументы кодируются в двоичной системе счисления. Если аргументов несколько, то они записываются на ленте подряд с использованием разделительного символа Λ .

Пусть всюду определенная функция $f(x_1, \dots, x_n)$ вычислима на машине Тьюринга $\mathcal{T}$. При фиксированных $x_1, \dots, x_n$ зоной вычисления значения $f(x_1, \dots, x_n)$ называем наименьшую связную часть ленты, которая содержит исходную двоичную запись набора $(x_1, \dots, x_n)$, а также все клетки, которые машина $\mathcal{T}$ посетила в процессе вычисления значения $f(x_1, \dots, x_n)$. Пусть $S(l)$ — функция натурального аргумента l с натуральными значениями. Говорим, что функция $f(x_1, \dots, x_n)$ вычислима в пределах зоны (или с зоной) $S(l)$, если для любого набора $(x_1, \dots, x_n)$, имеющего двоичную запись длины l , зона вычисления значения $f(x_1, \dots, x_n)$ не превосходит $S(l)$. Через LSPACE обозначим множество всех всюду определенных функций, которые вычислимы на машинах Тьюринга с линейными зонами (то есть с зонами $S(l)$ вида $al + b$, где a, b — натуральные числа).

Предложение 5. *Имеет место равенство*

$$\mathcal{E}^2 = \text{LSPACE}.$$

Доказательство. Для доказательства включения $\mathcal{E}^2 \subseteq \text{LSPACE}$ мы воспользуемся предложением 2. С помощью машины Тьюринга $\mathcal{T}$ будем моделировать вычисления на k -ленточной машине Минского $\mathcal{M}$. При этом набору чисел $y_1, \dots, y_k$, отмечаемых на лентах головками машины $\mathcal{M}$, сопоставим на ленте машины $\mathcal{T}$ тот же самый набор, но записанный в указанном выше двоичном коде. Нетрудно понять, как именно машина $\mathcal{T}$ будет преобразовывать набор $(y_1, \dots, y_k)$ в непосредственно следующий (для машины $\mathcal{M}$) набор $(y'_1, \dots, y'_k)$.

Несколько сложнее устанавливается включение $\text{LSPACE} \subseteq \mathcal{E}^2$. Здесь можно либо провести арифметизацию вычислений на машине Тьюринга непосредственно в классе $\mathcal{E}^2$, либо промоделировать эти вычисления на машине Минского. Наметим основные моменты подобного моделирования.

Одним из основных моментов является кодирование конфигураций на ленте машины Тьюринга с помощью пар натуральных чисел. Прежде всего, будем считать, что символам $\Lambda, 0, 1$ ленточного алфавита машины Тьюринга сопоставлены числа $0, 1, 2$, а словам в алфавите $\{\Lambda, 0, 1\}$ — соответствующие числа при использовании троичной системы счисления.

Будем предполагать, что клетки ленты машины Тьюринга перенумерованы целыми числами. Для краткости вместо клетка с номером s будем говорить клетка s . Пусть в некоторый момент времени t головка машины Тьюринга находится в клетке s , и пусть L_s — наименьшее слово, которое записано на ленте слева от клетки s и которое содержит все символы 0 и 1 (если слева от клетки s символы $0, 1$ отсутствуют, то L_s по определению есть Λ). Аналогично, R_s есть слово, которое записано на ленте справа от клетки s , включая саму эту клетку, и которое содержит все символы 0 и 1 . Слова L_s и R_s определяют в троичной системе счисления соответствующие числа l_s и r_s (по техническим причинам младшие разряды слова R_s считаем расположенными слева). Именно эта пара чисел будет кодировать конфигурацию на ленте машины Тьюринга в момент времени t .

Если на двух лентах моделирующей машины Минского представлены числа l_s и r_s , то нетрудно преобразовать эту пару чисел в пару чисел (l'_p, r'_p) , отвечающую моменту времени $t + 1$. В самом деле, если, например, в момент t машина Тьюринга сдвигает головку в клетку $s + 1$, записывая в клетке s символ b , то на лентах машины Минского число l_s будет преобразовано в число

$$l'_{s+1} = 3l_s + b,$$

а число r_s — в число

$$r'_{s+1} = \lfloor r_s/3 \rfloor.$$

При движении головки машины Тьюринга влево (по-прежнему в клетку s записывается символ b), соответствующие преобразования будут

$$\begin{aligned} l'_{s-1} &= \lfloor l_s/3 \rfloor, \\ r'_{s-1} &= \text{rm}(l_s, 3) + 3(r_s - \text{rm}(r_s, 3) + b). \end{aligned}$$

Участвующие здесь арифметические преобразования (умножение на 3, деление на 3 с остатком, сложение и вычитание) легко выполняются на машине Минского.

Случай, когда в момент t машина Тьюринга не сдвигает головку, весьма прост, и мы его не рассматриваем.

Остается рассмотреть преобразование начальной конфигурации машины Минского в пару чисел, кодирующую начальную конфигурацию машины Тьюринга. Для упрощения рассуждений будем предполагать, что в начальный момент времени головка машины Тьюринга находится в клетке s , соседней с крайним правым разрядом двоичной записи числа x_n . Тогда, очевидно, $r_s = 0$. Для вычисления l_s необходимо найти двоичные представления чисел $x_1, \dots, x_n$, преобразовать их в троичные представления (напомним, что при этом 0 заменяется на 1, а 1 на 2) и затем получить число, имеющее троичное представление $d(x_1)0d(x_2)0 \dots 0d(x_n)$, где $d(x_i)$ — указанное выше троичное представление числа x_i . Эту процедуру несложно осуществить средствами машины Минского. Сначала вычисляем двоичные разряды числа x_n и создаем из них число с троичным представлением $d(x_n)$. Так, последний двоичный разряд числа x_n равен $\text{gm}(x_n, 2)$, а в число с записью $d(x_n)$ он внесет величину $\text{gm}(x_n, 2) + 1$. Вообще, если i не превосходит длины двоичного представления числа x_n , то i -й разряд этого представления равен $\lceil \text{gm}(x_n, 2^i) / 2^{i-1} \rceil$, а его взнос в число с троичной записью $d(x_n)$ получается из найденного значения прибавлением 1 с последующим умножением на 3^{i-1} .

По завершении обработки двоичного представления числа x_n переходим к двоичному представлению числа x_{n-1} , при этом взнос младшего разряда $\text{gm}(x_{n-1}, 2)$ числа x_{n-1} будет равен $3^{l(x_n)+1}(\text{gm}(x_{n-1}, 2) + 1)$, где $l(x_n)$ — длина двоичного представления числа x_n , и так далее.

В заключение доказательства отметим, что числа вида 2^i , 3^i , появляющиеся в описанной выше процедуре, как нетрудно видеть, не превосходят некоторого полинома от значений переменных $x_1, \dots, x_n$.

Предложение 5 можно обобщить на случай многоленточных машин Тьюринга, поскольку сведение нескольких лент к одной ленте можно осуществить максимум с линейным увеличением зоны.

Рассмотрим еще один вариант одноленточных машин Тьюринга, которые в процессе вычисления не используют дополнительные клетки ленты (по сравнению с теми, которые в начальный момент вычисления заняты двоичной записью аргументов). Назовем эти машины машинами Тьюринга без расширения зоны (сокращенно машины БРЗ, эти машины называются также линейно ограниченными автоматами [12, 13]). Чтобы иметь возможность вычислять на машинах БРЗ произвольные функции из класса $\mathcal{E}^2$, расширим прежний ленточный алфавит $\{\Lambda, 0, 1\}$ дополнительными символами $2, 3, \dots, m-1$, где $m \geq 2$. При вычислении функции $f(x_1, \dots, x_n)$ на машине БРЗ $\mathcal{M}$ в начальный момент времени на ленте машины (так же, как в случае рассмотренного выше варианта машин Тьюринга) представлены в двоичной записи значения аргументов $x_1, \dots, x_n$. В процессе вычисления машина $\mathcal{M}$ не выходит за пределы зоны, ограниченной слева первым символом двоичной записи x_1 и справа символом Λ , примыкающим к последнему символу двоичной записи x_n . При этом на обозначенной зоне вычисления машина $\mathcal{M}$ может использовать любые символы алфавита $\{\Lambda, 0, 1, \dots, m-1\}$. В отличие от предыдущего варианта машин Тьюринга требуется только, чтобы при некотором k , $2 \leq k \leq m$, результат вычисления значения $f(x_1, \dots, x_n)$ был представлен в k -ичной системе счисления с цифрами $0, 1, \dots, k-1$.

Понятно, что время вычисления значения $f(x_1, \dots, x_n)$ определяется величиной $(m+1)^{l(x_1, \dots, x_n)}$, где $l(x_1, \dots, x_n)$ — длина двоичной записи набора $(x_1, \dots, x_n)$. Поэтому для всюду определенной функции $f(x_1, \dots, x_n)$ время вычисления на машине БРЗ ограничено некоторым полиномом от $x_1, \dots, x_n$. Это дает основание предполагать, что

справедливо следующее утверждение.

Предложение 6. *Класс $\mathcal{E}^2$ состоит в точности из всех всюду определенных функций, которые можно вычислить на машинах Тьюринга БРЗ.*

Мы не будем проводить доказательство этого утверждения во всех деталях — его можно доказать, например, так же, как предложение 5. Однако, имея доказанное предложение 5, можно провести взаимное моделирование машин Тьюринга из предложения 5 и машин Тьюринга БРЗ.

Пусть функция f из класса $\mathcal{E}^2$ вычислима на одноленточной машине Тьюринга $\mathcal{M}$ с линейной зоной. Проведем стандартное преобразование машины $\mathcal{M}$, которое упростит дальнейшее моделирование. Именно, преобразуем машину $\mathcal{M}$ в такую машину $\mathcal{M}_1$, которая также вычисляет функцию f с линейной зоной и у которой головка в процессе вычисления не посещает клетки ленты, расположенные левее первой клетки двоичной записи аргументов функции f .

Пусть зона вычисления функции f на машине $\mathcal{M}_1$ ограничена линейной функцией $al + b$, где a, b — натуральные числа. Машина Тьюринга БРЗ $\mathcal{M}'$, моделирующая работу машины $\mathcal{M}_1$, будет действовать следующим образом. Всю зону вычисления машины $\mathcal{M}_1$ она условно разбивает на $a + 1$ отрезков: a отрезков длины l (здесь l — длина исходной двоичной записи аргументов функции f) и один (последний) отрезок длины b . На каждом из отрезков длины l машина $\mathcal{M}'$ осуществляет прямое моделирование работы машины $\mathcal{M}_1$, помня номер соответствующего отрезка. За счет выбора ленточного алфавита машины $\mathcal{M}'$ все эти a отрезков длины l можно поместить вместе на одном и том же участке длины l ленты машины $\mathcal{M}'$. Моделирование работы машины $\mathcal{M}_1$ на отрезке ленты длины b можно, например, выполнить за счет внутренней памяти машины $\mathcal{M}'$ (напомним, что константа b зависит только от машины $\mathcal{M}_1$). После того как на ленте машины $\mathcal{M}_1$ будет получено значение функции f , машина $\mathcal{M}'$ переводит его двоичную запись (которая, вообще говоря, может занимать зону длины $al + b$) в k -ичную запись.

Обратно, пусть машина Тьюринга БРЗ $\mathcal{M}'$ вычисляет всюду определенную функцию f , и пусть $\{\Lambda, 0, 1, \dots, m - 1\}$ — ленточный алфавит машины $\mathcal{M}'$. Положим

$$r = \lceil \log_2 m \rceil$$

и закодируем символы $0, 1, \dots, m - 1$ словами длины r в алфавите $\{0, 1\}$. Одноленточная машина Тьюринга $\mathcal{M}$ будет моделировать работу машины $\mathcal{M}'$, используя вместо символов $0, 1, \dots, m - 1$ их двоичные коды длины r (символ Λ машина $\mathcal{M}$ использует так же, как и машина $\mathcal{M}'$). Понятно, что в процессе моделирования машине $\mathcal{M}$ достаточно увеличить зону максимум в r раз. По окончании моделирования машина $\mathcal{M}$ переводит блочный код k -ичного результата (напомним, что на ленте машины $\mathcal{M}'$ он записан в k -ичной системе счисления) в обычную двоичную запись.

Нашу серию вычислительных устройств мы завершим рассмотрением линейных однородных структур [2, 4] (сокращённо ЛОС). Линейная однородная структура определяется конечным автоматом A с множеством внутренних состояний $Q = \{q_0, q_1, \dots, q_r\}$, двумя входами, двумя выходами и функцией переходов $g_A: Q^3 \rightarrow Q$. Для любого натурального числа l ЛОС O_l состоит из l копий $A_1, \dots, A_l$ автомата A , расположенных в линейную цепь: при любом i , $1 < i < l$, выходы автомата A_i соединены со входами автоматов A_{i-1} и A_{i+1} , а правый выход автомата A_{i-1} и левый выход автомата A_{i+1} — со входами автомата A_i . В множестве Q выделено состояние q_0 и на левый вход автомата A_1 и правый вход автомата A_l постоянно подается символ q_0 (символ границы ЛОС).

Функционирование ЛОС O_l происходит следующим образом. В начальный момент времени все автоматы ЛОС O_l приводятся в начальное состояние q_1 . В каждый последующий момент времени изменения состояний автоматов $A_1, \dots, A_l$ происходят согласно функции переходов g_A . ЛОС O_l прекращает функционирование, когда все составляющие ее автоматы окажутся в заключительном состоянии q_r .

Чтобы иметь возможность вычислять на ЛОС функции, нам придется несколько изменить понятия начальной и заключительной конфигураций. Во-первых, так же, как для одноленточных машин Тьюринга, входные наборы $(x_1, \dots, x_n)$ мы будем подавать на ЛОС в двоичной записи с использованием в качестве разделительного символ Λ . Далее, мы полагаем, что в множестве $\mathcal{Q}$ выделены три начальных состояния $q_1^\Lambda, q_1^0, q_1^1$, которые отвечают символам $\Lambda, 0, 1$, используемым в двоичной записи аргументов. Таким образом, если l — длина двоичной записи набора $(x_1, \dots, x_n)$, то в начальный момент времени в ЛОС O_l начальные состояния $q_1^\Lambda, q_1^0, q_1^1$ автоматов $A_1, \dots, A_l$ образуют двоичную запись набора $(x_1, \dots, x_n)$.

Подобное соглашение мы примем и для заключительных состояний автомата A . Имено, для некоторого $k \geq 2$ в автомате A будет выделено k заключительных состояний $q_r^0, q_r^1, \dots, q_r^{k-1}$, и в заключительные моменты времени значения вычисляемой на ЛОС функции будут считываться в k -ичной системе счисления с заключительных состояний автоматов $A_1, \dots, A_l$.

Предложение 7. *Класс $\mathcal{E}^2$ состоит в точности из всех всюду определенных функций, которые можно вычислить на линейных однородных структурах.*

Предложение 7 можно вывести из предложения 6, поскольку машины Тьюринга БРЗ и линейные однородные структуры чрезвычайно просто моделируют друг друга.

В заключение статьи рассмотрим один вопрос, который, как оказалось, тесно связан с машинными заданиями класса $\mathcal{E}^2$ Гжегорчика. Речь пойдет о построении в классе $\mathcal{E}^2$ конечного базиса по суперпозиции. Вопрос этот сформулирован для класса $\mathcal{E}^2$ (и для других классов иерархии Гжегорчика) А. Гжегорчиком в [11]. Положительное решение вопроса получено в [6], где использовалось задание класса $\mathcal{E}^2$ с помощью некоторых вычислительных устройств, близких к машинам Минского. Впоследствии [7] (см. также [8]) уже с применением машин Минского существование конечных базисов по суперпозиции было доказано для большой серии так называемых $\mathcal{E}^2$ -замкнутых классов (в частности, для всех классов иерархии Гжегорчика за исключением классов $\mathcal{E}^0$ и $\mathcal{E}^1$). Отличительной особенностью доказательств из [6, 7] являлось построение специальных нумераций используемых вычислительных устройств и определение на их основе так называемых квазиуниверсальных функций. Эти построения были довольно громоздки и технически сложны.

Сравнительно недавно С. А. Волков [3] сумел определить монотонной рекурсией очень простую квазиуниверсальную функцию Q , в определение которой не входят явно нумерации каких-либо вычислительных устройств, равно как и другие понятия, связанные с вычислительными устройствами. Тем не менее доказательство квазиуниверсальности предложенной функции Q опирается на машины Минского и предложение 2. Функция Q , определенная С. А. Волковым, замечательна еще и тем, что является едва ли не первым примером достаточно сложно вычислимой функции, определение которой не содержит диагональных и универсально-языковых конструкций.

Чтобы привести примитивно рекурсивную схему определения функции Q , введем еще две вспомогательные функции из класса $\mathcal{E}^2$. Пусть $x \wedge y$ есть поразрядная конъюнкция двоичных представлений чисел x и y : если $x_n x_{n-1} \dots, x_0$ и $y_n y_{n-1} \dots, y_0$ — двоичные представления чисел x и y , то двоичное представление числа $x \wedge y$ есть

$(x_n y_n)(x_{n-1} y_{n-1}) \dots (x_0 y_0)$ (в случае, когда длины двоичных представлений различны, старшие разряды двоичного представления меньшего числа полагаем равными нулю).

Определим $R(x, y)$ как циклический сдвиг двоичного представления числа x на y разрядов вправо. Иными словами, пусть $R(0, y) = 0$, $R(1, y) = 1$, а если $x \geq 2$ и $x_n x_{n-1} \dots, x_0$ — двоичное представление числа x , причем $x_n = 1$, то двоичное представление числа $R(x, y)$ есть $x_{n+y} x_{n+y-1} \dots x_{1+y} x_y$, где все сложения проводятся по модулю $n + 1$.

Функция $Q(x, p_1, p_2, c_1, c_2, t)$ определяется следующей монотонной ограниченной рекурсией:

$$Q(x, p_1, p_2, c_1, c_2, 0) = x,$$

$$Q(x, p_1, p_2, c_1, c_2, t + 1) = \begin{cases} Q(x, p_1, p_2, c_1, c_2, t) & \text{если } Q(x, p_1, p_2, c_1, c_2, t) \\ & \wedge R(p_1, c_1 t) \neq 0, \\ Q(x, p_1, p_2, c_1, c_2, t) & \\ + R(p_2, c_2 t) & \text{в противном случае,} \end{cases}$$

$$Q(x, p_1, p_2, c_1, c_2, t) \leq x + 2p_2 t.$$

В [3] доказана следующая теорема.

Теорема 1. Система функций

$$\begin{aligned} & x + 1, \quad xy, \quad \min(x, 2^y), \quad x \div y, \\ & [x/y], \quad [\log_2 x], \quad Q(x, p_1, p_2, c_1, c_2, t) \end{aligned} \quad (4)$$

образует базис по суперпозиции в классе $\mathcal{E}^2$ Гжегорчика.

Как нетрудно видеть, все функции системы (4), отличные от функции Q , можно вычислить с временной сложностью (относительно длины двоичной записи аргументов), не превосходящей по порядку l^2 (на самом деле даже $l \log^2 l$). Вместе с тем функции из класса $\mathcal{E}^2$ имеют экспоненциальную сложность вычисления (см. предложения 5–7). Из приведенной теоремы следует, что при любом фиксированном k функции Q не может быть вычислена со сложностью, равной по порядку l^k . Действительно, в противном случае всякая функция из $\mathcal{E}^2$, будучи представлена (конечной) суперпозицией функций системы (4), также была бы вычислима со сложностью порядка l^m , где m — подходящая константа.

Итак, функция Q служит примером функции, которую невозможно вычислить с полиномиальной сложностью.

Список литературы

1. Бельтюков А. П., Машинное описание и иерархия начальных классов Гжегорчика. *Записки научн. семинаров Ленинградского отд. Матем. ин-та им. В. А. Стеклова АН СССР* (1979) **88**, 30–46.
2. Варшавский В. И., Мараховский В. Б., Песчанский В. А., Розенблум Л. Я., *Однородные структуры*. Энергия, Москва, 1973.
3. Волков С. А., Пример простой квазиуниверсальной функции в классе $\mathcal{E}^2$ иерархии Гжегорчика. *Дискретная математика* (2006) **18**, №4, 31–44.
4. Кудрявцев В. Б., Подколзин А. С., Болотов А. А., *Основы теории однородных структур*. Наука, Москва, 1990.

5. Мальцев А. И., *Алгоритмы и рекурсивные функции*. Наука, Москва, 1986.
6. Марченко С. С., Устранение схем рекурсий в классе $\mathcal{E}^2$ Гжегорчика. *Матем. заметки* (1969) **5**, №5, 561–568.
7. Марченко С. С., Об ограниченных рекурсиях. *Math. Balk.* (1972) **2**, 124–142.
8. Марченко С. С., Базисы по суперпозиции в классах рекурсивных функций. *Матем. вопросы киберн.* (1991) **3**, 115–139.
9. Марченко С. С., *Элементарные рекурсивные функции*. МЦНМО, Москва, 2003.
10. Минский М., *Вычисления и автоматы*. Мир, Москва, 1971.
11. Гжегорчик А., Некоторые классы рекурсивных функций. *Проблемы математической логики* (1970) **7**, 9–49.
12. Курода С. И., Классы языков и линейно ограниченные автоматы. *Киберн. сб.* (1972) **9**, 36–51.
13. Myhill J., *Linearly bounded automata*, Wright Air Development Division Technical Note 60–165. Wright-Patterson Air Force Base, Ohio, 1960.

Статья поступила 15.09.2009.