Пусть $K, L \in G$. Пусть $a \in E_2, \alpha \in E_2^n$. Пусть $H(K) = A, H(L) = B$. Тогда $K \circ L(a\alpha) = aB(A(\alpha)) = a(A \circ B)(\alpha)$, откуда следует $H(K \circ L) = H(K) \circ H(L)$.

**Список литературы**

1. Кудрявцев В. Б., Алешин С.В., Подколзин А. С. Введение в теорию автоматов. — М.: Наука, 1985.

2.. Каргаполов М. И., Мерзляков Ю. И. Основы теории групп. — М.: Наука, 1982.

3. Makarov V. V. The automaton permutation group $AS_n$ is generated by elements of infinite order // Discrete Mathematics and Applications. — 1997. — Vol. 7, Iss. 5. — P. 455–463.

4. Макаров В. В. О группах автоматных перестановок // Фунд. и прикл. мат. — 1996. —Т. 2, Вып. 1. — С. 171–186.

DOI: 10.20948/dms-2022-35

## ТРУДНО ОПИСУЕМЫЕ ПОЧТИ ПОЛИНОМИАЛЬНЫЕ КЛАССЫ ВОЗВРАТНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

**С. Д. Макеев (Москва)**

Возвратная последовательность — последовательность чисел (в нашем случае — целых), в которой каждое следующий элемент вычисляется как некоторая функция от нескольких предыдущих: $x_n = f(x_{n-m}, ..., x_{n-1})$. Функция $f$ называется *порождающей*, а число $m$ — *порядком* последовательности. Каждая последовательность полностью задаётся её порождающей функцией и первыми $m$ элементами.

Класс последовательностей задаётся классом функций, из которого можно выбирать их порождающие функции. Некоторые классы последовательностей являются «просто описуемыми»: например, для класса, в котором порождающими функциями могут быть только линейные, известны явные формулы $n$-го члена последовательности. Но для класса, порождающими функциями которого являются все полиномы с целыми коэффициентами, неизвестно, является ли он просто описуемым. Мы будем доказывать, что некоторые чуть большего размера классы являются «трудно описуемыми», в том смысле, что предсказать поведение последовательности этого класса — настолько же сложная задача, как предсказать поведение машины Тьюринга, т.е. алгоритмически неразрешимая. Если мы докажем, что последовательности этого класса могут «кодировать» собой вычисления некоторой машины, это будет считаться достаточным обоснованием того, что этот класс трудно описуем.

Приведём исторический пример использования подобного рода доказательств: Джон Конвей исследовал последовательности наподобие Коллатца, в которых порождающие функции содержат проверку на делимость, и получил то, что впоследствие назвал языком программирования FRACTRAN [4]. Каждая FRACTRAN-программа задаёт возвратную последовательность, порождающая функция которой состоит из линейных функций и проверок на делимость. Конвей продемонстрировал, что FRACTRAN - полный по Тьюрингу язык, и поэтому класс всех «Коллатц-подобных» последовательностей трудно описуем.

Мы рассматриваем классы последовательностей, которые можно назвать «почти полиномиальными». Их порождающие функции — такие, что их можно построить композициями из некоторой функции $f$ и всех полиномов с целыми коэффициентами. Функцию $f$, для которой вышеупомянутый класс последовательностей является трудно описуемым, мы будем называть «граничной». Из ранее опубликованных исследований этой задачи известно, что граничной является, например, функция $sg(x)$ [2], а в некоторых неопубликованных работах также рассматривались функции $|x|$ и $\lfloor\sqrt{x}\rfloor$.

**Теорема.** *Для любого целого $N$, кроме $0$, функция $Nj_0(x)$, равная $N$ при $x = 0$ и нулю иначе, является граничной.*

Для доказательства этой теоремы мы используем машины Минского [1] — вариант машин Тьюринга, про который известно, что они являются настолько же универсальными, как и обычные машины Тьюринга. Мы берём произвольную машину Минского и записываем процесс её работы в виде числовой последовательности - так называемой «хронологии», а потом про эту последовательность доказываем, что она является возвратной последовательностью с порождающей функцией, которая построена из $Nj_0$ и полиномов. Следовательно, предсказать поведение последовательностей такого класса — алгоритмически неразрешимая задача, а поэтому функция $Nj_0$ граничнa.

Подобная техника доказательства ранее применялась в работе [2] для доказательства граничности функции $sg(x)$.

Все остальные полученные нами результаты - это следствия из этой теоремы. Сначала мы доказываем несколько основных следствий, которые потом будут «фундаментом» для дальнейших построений.

**Теорема.** *Любая функция, которая равна нулю во всех точках,* *кроме конечного количества (но не является тождественно нулевой) — граничная*

**Теорема.** *Любая функция, которая на какой-либо бесконечной полупрямой равна нулю (но не является тождественно нулевой) — граничная.*

Из последнего очень просто следует граничность ранее упомянутых $sg(x)$ и $|x|$.

Из этих основных следствий выводятся более сложные, применимые к очень большим семействам функций.

**Теорема.** *Пусть $\varphi(x)$ — функция, определённая на действительных числах, такая что:*
*— На некоторой полупрямой её производная существует.*
*— На этой полупрямой её производная не меняет знака, но стремится к нулю на соответствующей бесконечности.*
*— Функция $f(x) = \lfloor\varphi(x)\rfloor$ в целочисленных точках этой полупрямой принимает как минимум три разных значения.*
*Тогда $f(x)$ (рассматриваемая как целочисленная функция) является граничной.*

Эта теорема, неформально говоря, делает граничными все «медленно растущие» функции. Ранее упомянутая функция $\lfloor\sqrt{x}\rfloor$ является лишь частным случаем.

**Теорема.** *Функции вида $\lfloor x^a \rfloor$ для всех действительных, нецелых a являются граничными.*

Последнее особенно примечательно тем, что описываемые функции, из-за ошибок округления, ведут себя «псевдослучайно» при больших $x$. Тем не менее, они являются граничными, что означает, что из псевдослучайной функции можно сделать детерминированное вычислительное устройство.

**Список литературы**

1. Марченков С. С., Савицкий И. В. Машины в теории вычислимых функций. — М.: МАКС Пресс, 2018.

2. Марченков С. С. О сложности полиномиальных возвратных последовательностей // Проблемы передачи информации — 2018. — Т. 54, вып. 3. — С. 258–262.

3. Холл М. Комбинаторика. — М.: Мир, 1970.

4. Conway J. FRACTRAN — a simple universal programming language for arithmetic // Open problems in communication and computation — 1986 — P. 4–26

DOI: 10.20948/dms-2022-36