Московский Государственный Университет
имени М.В. Ломоносова
Российская Академия Наук
Международная Академия Технологических Наук
Российская Академия Естественных Наук

# Интеллектуальные Системы.

## Теория и приложения

ТОМ **24** ВЫПУСК 4 * **2020**

МОСКВА

ISSN 2411–4448

УДК 519.95; 007:159.955
ББК 32.81

Издается с 1996 г.*



В журнале «Интеллектуальные системы. Теория и приложения» публикуются научные достижения в области теории и приложений интеллектуальных систем, новых информационных технологий и компьютерных наук.

Издание журнала осуществляется под эгидой МГУ имени М. В. Ломоносова, Научного Совета по комплексной проблеме «Кибернетика» РАН, Отделения «Математическое моделирование технологических процессов» МАТН, Секции «Информатики и кибернетики» РАЕН.

Учредитель журнала: ООО «Интеллектуальные системы».

Журнал входит в список изданий, включенных ВАК РФ в реестр публикаций материалов по кандидатским и докторским диссертациям по математике и механике.

Спонсором издания является:

**ООО «Два Облака»**
Разработка корпоративных информационных систем
http://www.dvaoblaka.ru

Индекс подписки на журнал: 64559 в каталоге НТИ «Роспечать».

Адрес редакции: 119991, Москва, ГСП-1, Ленинские Горы, д. 1, механико-математический факультет, комн. 12-01.
Адрес издателя: 115230, Россия, Москва, Хлебозаводский проезд, д. 7, стр. 9, офис 9. Тел. +7 (495) 939-46-37, e-mail: mail@intsysjournal.org

*) Прежнее название журнала: «Интеллектуальные системы».

ОГЛАВЛЕНИЕ

# Часть 1.
# Общие проблемы теории интеллектуальных систем

# Анализ атрибутивной политики безопасности с использованием методов автоматического планирования

Афонин С.А.[1], Бонюшкина А.Ю.[2]

В работе рассматривается задача проверки возможности получения пользователем информационной системы доступа к выделенному объекту при заданной атрибутивной политике безопасности. Показывается, что при некоторых ограничениях на модель информационной системы и правила политики эта задача сводится к задаче интеллектуального планирования. Определяется древовидная структура, построение которой соответствует планированию в пространстве планов и позволяет учитывать специфику решаемой задачи при построении эвристических алгоритмов проверки доступа. Доказывается, что существование такой структуры является необходимым и достаточным условием возможности получения доступа к целевому объекту.[1 2]

Работа выполнена при поддержке гранта РФИИ 18-07-01055.

**Ключевые слова:** ABAC, проверка доступа, интеллектуальное планирование.

## 1. Введение

Управление правами доступа является важной составляющей информационных систем. В многопользовательских системах необходимо ограничивать возможность выполнения *субъектами* доступа (пользователями или программами) тех или иных операций с *объектами* доступа. В общем случае доступ зависит от свойств объекта и субъекта доступа, вида операции и *контекста*. Примером контекста является время выполнения

[1] *Афонин Сергей Александрович* — доцент каф. дискретной математики мех.-мат. ф-та МГУ, e-mail: serg@msu.ru.

Afonin Sergey Aleksandrovich — associate professor, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Numerical Analysis.

[2] *Бонюшкина Антонина Юрьевна* — аспирант каф. дискретной математики мех.-мат. ф-та МГУ, e-mail: abonush@yandex.ru.

Bonyushkina Antonina Yuryevna — graduate student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Numerical Analysis.

операции, текущее физическое положение пользователя или другие подобные данные. Доступ субъектов к объектам определяется правилами о допустимых операциях субъекта над объектом. Например, в правиле «Сотрудник отдела кадров имеет право изменять карточку работника с 10 до 17 часов» объектом доступа является карточка работника (запись в базе данных), субъектом — сотрудник отдела кадров, операция — редактирование записи. При этом накладывается дополнительное ограничение на контекст — время выполнения операции. Набор правил доступа такого вида называется *политикой безопасности*.

На практике политика безопасности описывается документом на естественном языке. Для реализации политики безопасности в коде системы правила доступа представляются в терминах некоторой *модели доступа*. Использование формальной модели позволяет проводить анализ политики на предмет выполнения некоторых свойств. Например, рассматриваются такие свойства как непротиворечивость правил политики друг другу, доступность объектов доступа определенным пользователям, эквивалентность политик и другие [4, 3].

Алгоритмическая разрешимость и сложность задач анализа политики доступа зависит от выразительной силы модели доступа. В литературе описывается множество моделей доступа. К числу наиболее известных и распространенных моделей относятся дискреционная (DAC), мандатная (MAC) и ролевая (RBAC) модель доступа. В последнее время значительную популярность приобретают модели типа ABAC [8], которые позволяют использовать свойства (атрибуты) объектов и субъектов доступа при определении правил доступа, что позволяется расширить спектр возможных правил по сравнению с ролевой моделью. При этом атрибутивные модели включают некоторые классические модели доступа [1].

В данной работе мы рассматриваем задачу проверки возможности получения пользователем доступа к выделенному объекту в результате выполнения последовательности операций, разрешенных ему политикой CBAC. Мы считаем, что в информационной системе, состоящей из набора взаимосвязанных объектов, работает один пользователь (злоумышленник), который пытается получить доступ к выделенному объекту, выполняя только разрешенные ему операции.

Сформулируем задачу более формально. Пусть задано множество *переменных* (или *объектов* информационной системы) $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. Переменные принимают значения из множества $X$. *Состоянием системы* является вектор $x \in X^N$ значений переменных. С каждой переменной $\mathbf{x}_i$ связано *множество доступности* $A_i \subseteq X^N$, которое определяет условия на значения переменных, при которых значение переменной $\mathbf{x}_i$ может быть изменено. Если текущее состояние $x$ принадлежит $A_i$, то пользо-

ватель может перевести систему в состояние $x'$, которое отличается от $x$ только в $i$-ой компоненте. Такое действие назовем *допустимым* и будем обозначать как $x \mapsto_i x'$. За одну операцию можно изменять только одну переменную. Задача состоит в следующем. Для заданной системы множеств $\mathcal{P} = \langle A_1, A_2, \ldots, A_N \rangle$, индекса $t \in \{1, \ldots, N\}$ (определяющего переменную, к которой мы хотим получить доступ) и начального состояния $x^0 \in X^N$ требуется проверить, что существует конечная последовательность допустимых действий, которая переводит состояние $x^0$ в некоторое состояние $x' \in A_t$. Другими словами можно сказать, что в переходной системе, заданной системой множеств $\mathcal{P}$, требуется проверить достижимость множества $A_t$ из начального состояния $x^0$.

Очевидно, что если множества $A_i$ произвольны, то задача проверки доступа не имеет алгоритмического решения. Например, пусть заданы переменные $\{u, v, w\}$ и два рекурсивно перечислимых множества $U = \{u_1, u_2, \ldots\}$ и $V = \{v_1, v_2, \ldots\}$. Определим множества доступности переменных следующим образом:

$$\begin{aligned} A_u &= \{(u_i, v_i, 0) \mid i > 0\}, \\ A_v &= \{(u_{i+1}, v_i, 0) \mid i > 0\}, \\ A_w &= \{u^*, v^*, 0\}. \end{aligned}$$

При начальном состоянии $x^0 = (u_1, v_1, 0)$ единственно возможная бесконечная последовательность состояний системы имеет вид $(u_1, v_1, 0) \mapsto_u (u_2, v_1, 0) \mapsto_v (u_2, v_2, 0) \mapsto_u (u_3, v_2, 0) \mapsto_v \ldots$. Доступ к переменной $w$ возможен тогда и только тогда, когда либо $u^* \in U$, либо $v^* \in V$. В силу нерекурсивности множеств $U$ и $V$ выполнимость этих условий не может быть проверена.

Одно из возможных ограничений на структуру множеств доступности рассматривается в работе [2], где эти множества представляются в виде конечного объединения декартовых произведений некоторых множеств. В этом случае множество состояний системы разделяется на конечное число классов эквивалентности и задача проверки доступа сводится к поиску пути в конечном графе. Этот результат показывает разрешимость задачи, но число вершин этого графа ограничено $2^{2^N}$, что не позволяет использовать данный метод на практике.

Целью данной работы является решение задачи проверки доступа в условиях [2] методами интеллектуального планирования [7]. Для систем с конечным числом состояний универсальные алгоритмы классического планирования, такие как STRIPS, также сводятся к перебору возможных последовательностей действий и имеют экспоненциальную сложность [6]. Возможность разработки прикладных решений связана либо с рассмотрением специальных подклассов исходной задачи, либо с учетом различных эвристик и планированием в пространстве частичных

целей [5]. В данной работе предлагается алгоритм построения плана, который использует свойства задачи проверки доступа.

В следующем разделе приводится основные определения и формальная постановка задачи проверки корректности политики. В разделе 3 описывается интерпретация задачи в виде графа зависимостей и доказываются некоторые его свойства. Метод проверки корректности политики путем выделения промежуточных целей и его связь с задачей интеллектуального планирования описывается в разделе 4. В разделе 5 вводится понятие дерева порядков и доказывается основная теорема — критерий возможности получения доступа.

## 2. Определения и постановка задачи

Пусть задана система из $N$ *переменных* (будем также называть их *объектами*)

$$\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}.$$

Переменные могут принимать значения из множества $X$. *Состоянием системы* является вектор $x = (x_1, \ldots, x_N) \in X^N$ значений переменных.

С каждой переменной $\mathbf{x}_i$ связано непустое *множество доступности* $A_i \subseteq X^N$, которое определяет условия на все переменные, при которых значение переменной $\mathbf{x}_i$ может быть изменено: если текущее состояние $x$ принадлежит $A_i$, то систему можно перевести в состояние $x'$, изменив значение переменной $\mathbf{x}_i$:

$$x = (x_1, \ldots, x_i, \ldots, x_N), x' = (x_1, \ldots, x'_i, \ldots, x_N),$$

причем на её новое значение $x'_i$ не накладывается никаких ограничений. В один момент времени можно изменять одну переменную. Новое состояние $x'$ назовем *непосредственно достижимым* из $x$.

Условие «$x \in A_i$» называется *условием доступа* к переменной $\mathbf{x}_i$, и если оно выполняется, то изменение значения переменной $\mathbf{x}_i$ *допустимо* (к переменной есть доступ), и такое изменение называется *доступной операцией*. Условие доступа будем записывать через индикаторную функцию множества $A_i$:

$$I_{A_i}(\mathbf{x}) = \begin{cases} 1, & \text{если } \mathbf{x} \in A_i \\ 0, & \text{иначе.} \end{cases}$$

Таким образом, система непустых множеств $\mathcal{P} = \langle A_i \big| i = 1, \ldots, N \rangle$ определяет политику доступа к переменным для операции изменения.

Доступ к $\mathbf{x}_i$ *существенно зависит* от переменной $\mathbf{x}_j$, если существует пара состояний $x = (x_1, \ldots, x_j, \ldots, x_N)$ и $x' = (x_1, \ldots, x'_j, \ldots, x_N)$, отличающихся только в $j$-й координате, такие что $x \in A_i, x' \notin A_i$. Если таких

состояний не существует, то зависимость *фиктивная*. Из непустоты $A_i$ следует, что множество $A_i$ содержит все значения фиктивной $\mathbf{x}_j$. Существенные для доступа к $\mathbf{x}_i$ переменные (объекты) назовём *подобъектами* $\mathbf{x}_i$.

Доступную операцию изменения состояния обозначим $x \mapsto_i x'$: состояние $x$ переходит в состояние $x'$ изменением в $i$-той компоненте. Если из контекста ясно, какая переменная меняет значение, или это не важно, переход в другое состояние будем обозначать как $x \mapsto x'$.

Состояние $x'$ назовем *достижимым* из состояния $x^0$, если существует конечная последовательность состояний $x^1, \ldots, x^k = x'$, таких что $x^i \mapsto x^{i+1}\ \forall i \in \{0, 1, 2, \ldots, k-1\}$. Здесь каждое состояние отличается от предыдущего в одной переменной. Подчеркнем, что все изменения в цепочке должны быть доступны (то есть каждое состояние принадлежит множеству доступности переменной, которая меняется для перехода к следующему состоянию). Переход в достижимое состояние обозначим $x \mapsto^* x'$.

Множество $S$ назовем *достижимым* из состояния $x$, если существует состояние $x' \in S$, достижимое из $x$: $x' \in S : x \mapsto^* x'$, обозначение $x \mapsto^* S$.

*Замечание.* Если $X^N$ — это аффинное пространство, каждая переменная — координата, а состояние — это точка, то непосредственно достижимое из $x$ состояние $x'$, отличающееся в $i$-той координате получено «сдвигом по прямой вдоль $i$-той координаты». Множество $A_i$ является цилиндрической поверхностью вдоль фиктивной для $\mathbf{x}_i$ переменной.

**Задача проверки доступа** Будем считать, что у каждой переменной $\mathbf{x}_j$ есть $k_j$ существенных переменных $\mathbf{x}_{j_1}, \ldots, \mathbf{x}_{j_{k_j}}$, а множество доступности $A_j$ задается индикаторной функцией вида

$$I_{A_j}(\mathbf{x}) = \prod_{i=1}^{k_j} I_{A_{j,j_i}}(\mathbf{x}_{j_i}), \tag{1}$$

где $A_{j,j_i} \subseteq X$ есть множество значений переменной $\mathbf{x}_{j_i}$, при которых возможен доступ к $\mathbf{x}_j$. Это означает, что условия доступа $\mathbf{x}_j$ проверяются «независимо» для каждой из её существенных переменных и доступ к $\mathbf{x}_j$ есть, если выполняется каждое из условий справа в (1). Такие условия и функции назовем *разделимыми*.

Множества $A_{i,j}$ могут быть произвольными. Для реализации алгоритмов потребуем, чтобы они были «эффективно определены», то есть известны алгоритмы:

1) проверки пустоты пересечения $A_{i,j}$ со всеми $A_{k,j}$;

2) выбора представителя этого пересечения, если оно не пусто.

То есть мы должны знать, какие условия на $\mathbf{x}_j$ могут быть выполнены одновременно и уметь присваивать $\mathbf{x}_j$ такие значения. Для простоты будем считать, что эта информация известна и нам не надо тратить время на проверку и поиск представителя.

Задача состоит в следующем. Для заданной системы «эффективно определенных» множеств $\mathcal{P}$, удовлетворяющих условию (1), индекса $t \in \{1, \ldots, N\}$ и начального состояния $x^0 \in X^N$ требуется проверить, что $x \mapsto^* A_t$. Переменную $\mathbf{x}_t$ и множество $A_t$ будем называть соответственно *целевой* переменной и *целевым* множеством.

Для удобства обозначим $I_{i,j}(\mathbf{x}) = I_{A_{i,j}}(\mathbf{x}_j)$, подразумевая, что условие принадлежности множеству $A_{i,j}$ накладывается только на $\mathbf{x}_j$. Без ограничения общности можно считать, что целевой переменной является переменная $\mathbf{x}_0$.

Таким образом, если пользователю нужно получить доступ к $\mathbf{x}_0$, необходимо, чтобы выполнялись индикаторные условия на его подобъекты. Если какие-то из подобъектов $\mathbf{x}_0$ не удовлетворяют условиям, пользователь должен изменить их значения на требующиеся. Для этого у него должен быть доступ к этим объектам, который также определяется индикаторными условиями на их подобъекты, и так далее.

## 3. Графическое представление

Совокупность переменных с условиями доступа к ним можно представить ориентированным графом, который мы назовем *графом зависимостей.*

**Определение 1.** Полным графом зависимостей для состояния $x$ *называется ориентированный граф* $G(x) = \langle \mathbf{x}, E, \mathbb{I}(\mathcal{P}), \mu \rangle$ *с помеченными ребрами, который удовлетворяет следующим условиям:*

- *вершинами* $G$ *являются переменные* $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$*;*
- *пара вершин* $(\mathbf{x}_i, \mathbf{x}_j)$ *образует ориентированное ребро* $e_{i,j} \in E$*, если* $\mathbf{x}_j$ *является существенной переменной для* $\mathbf{x}_i$*;*
- *задан набор множеств доступности* $\mathcal{P}$*, пересечения которых мы умеем проверять на непустоту и брать представителя. Задан соответствующий ему набор разделимых функций доступа* $\mathbb{I} = \{I_{i,j}(\mathbf{x})\}$*;*
- разметкой *полного графа зависимостей по текущему состоянию* $x \in X^N$ *назовем функцию* $\mu : E \times X^N \to \{0, 1\}$*, такую, что* $\mu(e_{i,j}, x) = I_{i,j}(x)$*. Для определенности будем считать, что если ребра* $e_{i,j}$ *не существует, то* $\mu((\mathbf{x}_i, \mathbf{x}_j), x) \equiv 1$ $\forall x$ *(то есть*

*условие доступа всегда выполнено, что и следует из фиктивной зависимости между этими вершинами).*

В случае, когда понятно, какое состояние текущее, будем обозначать метку ребра в нем $\mu(e_{i,j}, x) = \mu(e_{i,j})$.

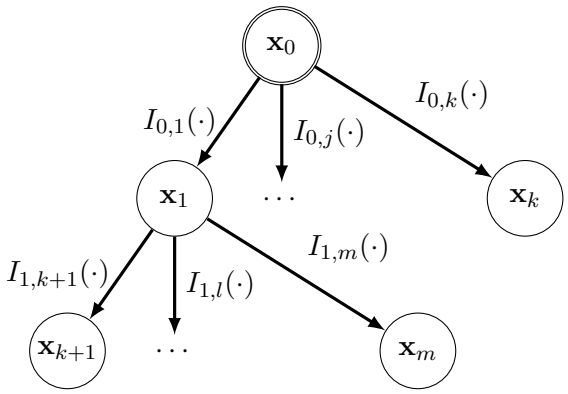



Рис. 1. Построение графа зависимостей методом поиска в ширину.

Для удобства обозначим через $out(\mathbf{x}_i)$ множество всех существенных переменных объекта $\mathbf{x}_i$, а $in(\mathbf{x}_i)$ — множество переменных, для которых $\mathbf{x}_i$ является существенным. Таким образом доступ к $\mathbf{x}_i$ зависит только от объектов из множества $out(\mathbf{x}_i)$, а сам $\mathbf{x}_i$ влияет только на доступ к переменным множества $in(\mathbf{x}_i)$.

В фиксированном состоянии $x$ индикаторные функции на ребрах принимают значения 0 или 1, что соответствует булевой разметке графа. Изменение значений объектов не меняют структуру графа, изменяя только разметку.

Для проверки доступа важно не то, какие именно значения принимают переменные, — а только, выполняются ли зависящие от них индикаторные функции. Поэтому для удобства будем рассматривать не значения объектов, а классы эквивалентности их значений через разметку: $\forall \mathbf{x}_i$ занумеруем элементы множества $in(\mathbf{x}_i) = \{1, \ldots, k\}$ и значение $\mathbf{x}_i$ запишем $k$-разрядным числом в двоичной системе счисления, где в $j$-том разряде стоит значение $I_{j,i}(\mathbf{x})$. Таким образом в двоичной записи представлено, какие условия на объект выполняются, а какие нет. Это число обозначим $val.in(\mathbf{x}_i)$. Обозначим $val.in(\mathbf{x}_i)|_x$ это значение в состоянии $x$. Если по контексту понятно, что рассматривается текущее состояние, то «$|_x$» будем опускать.

Аналогично поступим с исходящими ребрами объекта: перенумеровав их, запишем натуральное число в двоичной записи, каждому разряду которого соответствует значение, приписанное некоторому исходящему

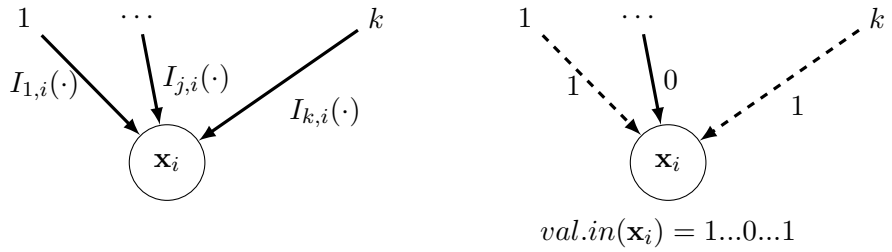



Рис. 2. Кодирование разметрки рёбер значением вершины.

ребру. Это число обозначим $val.out(\cdot)$. Обозначим $val.out(\cdot)|_x$ это значение в состоянии $x$. Если по контексту понятно, что рассматривается текущее состояние, то «$|_x$» будем опускать.

Доступ к объекту есть, только если его $val.out()$ состоит из одних единиц (обозначим такое значение $\overline{1}$) , и тогда мы можем менять его $val.in()$. При этом вполне возможно, что далеко не все значения $val.in()$ можно получить: то есть, не всегда можно подобрать такое значение, которое удовлетворяет всем условиям $I_{j,i}()$ (что соответствует $val.in() = \overline{1}$), поэтому задача и сложна). Набор значений, которые мы можем получить, напрямую связан с непустотой пересечения множеств $A_{i,j}$. Для решения нам важно знать, какие значения могут быть у всех $val.in()$. Для этого нужно знать, не пусто ли пересечение любого конечного числа $A_{i,j}, \mathbf{x}_i \in in(\mathbf{x}_j)$, и уметь брать представителя этого пересечения, чтобы присваивать его значение переменной $\mathbf{x}_j$.

Для удобства будем называть ребра, помеченные единицей, *зелеными*, а нулем — *черными*. Ориентированный путь, который является последовательностью черных ребер, назовём *черным путем*. Ориентированный цикл из черных ребер назовем *черным циклом*.

Заметим, что для наших целей — проверить, можно ли получить доступ к $\mathbf{x}_0$ — весь полный граф зависимостей с разметкой не нужен. Мы будем рассматривать такой подграф полного графа зависимостей, в который входит целевая вершина $\mathbf{x}_0$; все вершины, в которые ведет черный путь из $\mathbf{x}_0$; все ребра, которые соединяют эти вершины (и черные, и зеленые). Считаем, что число вершин в этом графе равно $n$. Получившийся граф $D$ будем называть просто *графом зависимостей* для доступа к $\mathbf{x}_0$.

### 3.1. Свойства графа зависимостей

Первое, что мы понимаем о графе зависимостей, — это то, что он не содержит лишних рёбер. Если доступ к целевой вершине можно получить, то каждое ребро графа зависимостей на некотором шаге процесса получения доступа должно принять метку 1. Аналогично и каждая вершина

на некотором шаге должна стать доступной. В противном случае получится, что к вершинам, которые требовалось изменить, не было получено доступа, а значит изменить их не удалось, и доступ к целевой вершине не мог быть получен. Это наблюдение формально записано ниже.

**Утверждение 1.** *Пусть состояние $x^m \in A_0$ достижимо по цепочке состояний $x^0$, $x^1$, ..., $x^m$. Тогда:*
*(1) для каждого ребра $e_{ij}$ графа $D$ существует $k \in \{0, \dots, m\}$, такое что $\mu(e_{ij}, x^k) = 1$;*
*(2) для любой вершины $\mathbf{x}_j$ графа $D$ существует $k$, такое что $val.out(\mathbf{x}_j)|_{x_k} = \overline{1}$.*

*Доказательство.* Следует из построения графа зависимостей. □

**Утверждение 2.** *Если в графе зависимостей в состоянии $x$ есть цикл из черных ребер, то доступ к целевой вершине получить невозможно.*

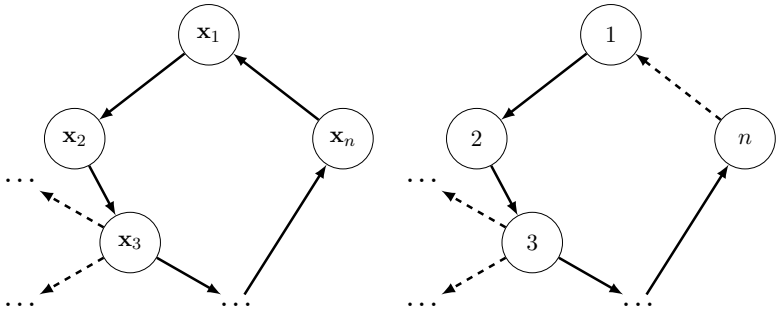
*Доказательство.* Черный цикл — это последовательность вершин $\mathbf{x}_1$, ..., $\mathbf{x}_k$, что $\mathbf{x}_{k+1} = \mathbf{x}_1$ и $\mu((\mathbf{x}_i, \mathbf{x}_{i+1}), x) = 0$, $i = 1, \dots, k$ (рис.3 (a)). Для получения доступа к любой вершине цикла необходимо изменить значения ее подобъектов, для получения доступа к подобъекту нужно изменить его подобъекты, и так далее. Приходим к тому, что для получения доступа к вершине ее саму необходимо изменить, что невозможно, так как доступа нет. Далее заметим, что из утверждения 1 следует, что если целевое множество достижимо, то каждая вершина должна по крайней мере один раз изменить значение (поскольку в графе в каждую вершину ведёт хотя бы одно черное ребро, которое должно на некотором шаге выполниться, а значит эту вершину придётся менять). □

Получается, в случае с черным циклом мы знаем, что у задачи нет решения (доступ получить нельзя). Ниже доказано, что если в графе зависимостей для начального состояния нет черных циклов, то они не могут появиться при выполнении допустимых операций.

**Утверждение 3.** *Пусть состояние $x^m$ достижимо из состояния $x^0$. Если в графе зависимостей $D(x^0)$ нет черных циклов, то и граф $D(x^m)$ не содержит черных циклов.*

*Доказательство.* Так как в один момент времени можно изменить значение только в одной вершине, достаточно показать, что нельзя выполнить один последний шаг, чтобы замкнуть цикл (рис.3 (b)).

Пусть есть цепочка пронумерованных вершин $1, \dots, n$, соединенных ребрами $(i, i + 1 \bmod n)$, $i = 1, ..n$, причем все указанные ребра, кроме $(n, 1)$ черные, а $(n, 1)$ — зеленое. Тогда его нельзя сделать черным, так как для этого необходимо изменить значение в вершине 1 на такое чтобы $\mu(n, 1) = 0$. А для этого нужен доступ к вершине 1, а для того чтобы его




(a) Цикл из чёрных ребер. (b) Дополнение до цикла.

Рис. 3. Циклы черных ребер в графе зависимостей.

получить, нужно чтобы $val.out(1) = \bar{1}$, в том числе $\mu(1,2) = 1$. Для этого нужен доступ к вершине 2, и т.д., нужен доступ к вершине $n$, допустим, что он есть, тогда изменяем ее так что $\mu(n-1,n) = 1\ldots$, $\mu(1,2) = 1$, теперь можно положить $\mu(n,1) = 0$ (теперь вершина $n$ недоступна), но даже если последовательно сделать черными ребра $\mu(1,2) = 0,\ldots,$ $\mu(n-2,n-1) = 0$, тогда $\mu(n-1,n)$ изменить нельзя, так как $n$ недоступна, оно останется зеленым, и ситуация та же, что была изначально: остался цикл без одного ребра. □

### 3.2. Порядок вершин графа зависимостей

Теперь мы знаем, что когда в графе зависимостей есть черный цикл, доступ получить нельзя, и что если в начальном состоянии черного цикла нет, то появиться он не может. Поэтому предполагаем, что мы умеем проверять наличие черного цикла в начальном состоянии графа и далее мы будем рассматривать только графы без черных циклов. Это позволяет упорядочить вершины естественным образом. Расположим вершины $D$ по уровням, введя частичный порядок $\preceq$, порожденный ориентированным черным путем при начальной разметке: если есть черный путь из $\mathbf{x}_i$ в $\mathbf{x}_j$, то $\mathbf{x}_j \preceq \mathbf{x}_i$. При этом любая вершина $\preceq \mathbf{x}_0$, которая находится на уровне с максимальным номером, а на самом нижнем уровне находятся вершины, доступные в начальном состоянии.

Далее мы без ограничения общности будем считать, что вершины графа зависимостей занумерованы в соответствии с отношением $\preceq$, то есть из $\mathbf{x}_i \preceq \mathbf{x}_j$ следует строгое неравенство $i < j$. Несравнимые вершины, находящиеся на одном уровне, пронумеруем между собой случайным образом, но в соответствии с частичным порядком относительно вершин других уровней.

Ниже показана корректность такой нумерации в смысле отсутствия черных рёбер, ведущих на уровень с бо́льшим номером.

**Утверждение 4.** *Пусть вершины графа зависимостей $D$ занумерованы в соответствии с отношением $\preceq$, и пара вершин $(\mathbf{x}_i, \mathbf{x}_j)$ образует ребро. Тогда $\mu(e_{ij}, x^0) = 0 \implies i < j$.*

*Доказательство.* Противоречие с тем, что путь до вершины, в которую ведет обратное ребро, самый длинный. □

**Следствие 1.** *При таком порядке вершин $\preceq$ для получения доступа к каждой вершине из начального состояния может потребоваться менять только вершины с меньшим номером (непосредственно следует из отсутствия обратных черных ребер).*

Приведенные выше свойства графа зависимостей обосновывают корректность его построения из полного графа $G$. По построению $D$ мы не взяли в него те вершины, в которые не ведут черные ребра из вершин, уже входящих в $D$. Это вершины, в которые входят только зеленые ребра из $D$ или не входят никакие (могут быть входящие ребра из $G$, но не $D$). Если в вершину $\mathbf{x}_j$ не входят ребра из $D$, то другие вершины из $D$ от нее не зависят, и изменение $\mathbf{x}_j$ не повлияет на доступ к ним, а значит нет причин получать к ней доступ и знать, от чего она зависит. Если в вершину $\mathbf{x}_j$ входят только зеленые ребра из $D$, значит все условия, зависящие от нее, уже выполнены, и ее также незачем менять. Таким образом, все вершины, которые не были включены в граф зависимостей целевой вершины, отделены от него ориентированными зелеными ребрами, которые никогда не понадобится менять.

Теперь мы имеем граф зависимостей, в котором последовательно будем менять значения вершин, попутно изменяя цвет ребер и доступность вершин. А значит порядок, связанный с длиной черного пути до вершины, будет меняться. Тем не менее, покажем, что при некоторых условиях можно использовать начальный порядок (и свойство, что для доступа к вершине нужны только вершины с меньшим номером в этом порядке) до момента получения доступа.

Прежде всего заметим, что любая цепочка допустимых действий обратима:

**Утверждение 5.** *Если состояние $x'$ было достигнуто из начального состояния $x^0$ цепочкой допустимых действий и соответствующей ей цепочкой состояний, то систему можно вернуть в состояние $x^0$ цепочкой обратных действий.*

*Доказательство.* Поскольку доступ к вершине не зависит от нее самой, а каждое действие — изменение значения одной вершины, не меняющее

ее доступность, то каждое действие в отдельности может быть обращено, то есть состояние переведено в предыдущее, в котором может быть обращен уже предыдущий шаг, и так далее до $x^0$. □

Свойство обратимости верно для любой цепочки допустимых действий. Справедливо и более сильное утверждение: в определенных случаях систему можно перевести в состояние, которое отличается от начального значением одной вершины.

**Утверждение 6.** *Пусть состояние $x' = (x'_1, \ldots, x'_n, x_{n+1}, \ldots, x_N)$ графа зависимостей $D$ было достигнуто из начального состояния $x^0 = (x^0_1, \ldots, x^0_n, x_{n+1}, \ldots, x_N)$ цепочкой допустимых действий и соответствующей ей цепочкой состояний, в которой менялись без ограничения общности только первые $n$ вершин $\mathbf{x}_1, \ldots, \mathbf{x}_n$. Пусть состояние $x'$ таково, что в нем существует вершина $\mathbf{x}_i, i \in \{1, \ldots, n\}$, принимающая значение $x'_i$, такое что $\mu((\mathbf{x}_j, \mathbf{x}_i), x'_i) = 1$, $j \in \{1, \ldots, n\}, j \neq i$. Тогда систему можно перевести в состояние $x^{0'} = (x^0_1, \ldots, x'_i, \ldots, x^0_n, x_{n+1}, \ldots x_N)$, отличающееся от начального в одной вершине, цепочкой обратных действий.*

Условие $\mu((\mathbf{x}_j, \mathbf{x}_i), x'_i) = 1$, $j \in \{1, \ldots, n\}, j \neq i$ значит, что в этом состоянии выполняются условия доступа для вершин, зависящих от $\mathbf{x}_i$ и изменяющихся в данной цепочке действий.

*Доказательство.* По условию мы меняли только вершины подграфа $D_n$ из первых $n$ вершин графа зависимостей $D$. Новое значение $x'_i$ удовлетворяет условиям доступа ко всем вершинам $D_n$. Значит присваивание вершине этого значения не повлияло на доступность других вершин $D_n$. Значит, за исключением нового значения $\mathbf{x}_i$, мы можем обратить все изменения для этих вершин к начальному состоянию, выполнив в обратном порядке все совершенные до сих пор изменения вершин. На каждом шаге единственное, что отличается от состояния, через которое проходил прямой процесс — это значение $\mathbf{x}_i$, но его изменение не повлияло на доступ к вершинам, которые изменяются, поэтому обратный шаг можно сделать. □

**Следствие 2.** *Очевидно, что порядок $\preceq$ на всех вершинах за исключением $\mathbf{x}_i$, в состоянии $x^{0'}$ «не испортится» в сравнении с $x^0$, в том смысле что для каждой вершины множество вершин $\{\mathbf{x}_j | \mathbf{x}_j \preceq \mathbf{x}_i\}$ в новом состоянии, не увеличилось.*

*Доказательство.* Поскольку значения вершин вернулись к прежним, а значит метки всех ребер, кроме тех, что входят в $\mathbf{x}_i$, остались прежними.

Могли добавиться только черные ребра, входящие в $\mathbf{x}_i$, но эта вершина не рассматривается. □

**Следствие 3.** *Аналогично можно из состояния $x^{0'}$ снова вернуться в состояние $x^0$, изменив в $x'$ вершину $\mathbf{x}_i$ на прежнее значение.*

## 4. Проверка доступа как задача планирования

Коротко опишем задачу классического планирования [7]. Пусть задана переходная система $\Sigma = \{S, A, \gamma\}$, где $S$ — конечное множество состояний, $A$ — конечное множество действий, $\gamma : S \times A \to S$ — функция переходов состояний.

Пусть также фиксировано начальное состояние $s_0 \in S$ и множество целевых состояний $S_g \subseteq S$. Задача классического планирования заключается в том, чтобы найти последовательность действий (план) $(a_1, a_2, \ldots, a_k)$ и соответствующую ему последовательность состояний $(s_0, s_1, \ldots, s_k)$, такую что $s_1 \in \gamma(s_0, a_1)$, $s_2 \in \gamma(s_1, a_2)$, $\ldots s_k \in \gamma(s_{k-1}, a_k)$, и $s_k \in S_g$. Действиям могут дополнительно ставиться в соответствие стоимость их реализации. В этом случае требуется найти план минимального (или максимального) веса.

Несложно заметить, что задача проверки доступа в точности является задачей классического планирования: состояния системы — это вектор значений вершин графа зависимостей для $\mathbf{x}_0$ в смысле класса эквивалентностей $\{val.in(\mathbf{x}_0), \ldots, val.in(\mathbf{x}_n)\}$. Количество классов эквивалентности конечно, поскольку это вектор из $n+1$ вершины, у каждой из которых не более $2^n$ возможных значений. Действием будет изменение значения доступной вершины, переход в другой класс, то есть изменение $val.in()$ этой вершины на другое возможное значение. Функция переходов состояний по состоянию и действию определяет новое состояние — с новым значением измененной вершины.

Для поиска оптимального плана используются методы *планирования в пространстве состояний* и *в пространстве планов*. В первом случае на каждом шаге алгоритм выбирает конкретное действие. Во втором — сначала строится множество промежуточных целей, последовательное достижение которых приводит к достижению состояния из целевого множества $S_g$, а потом производится поиск действий, переводящих систему от одной промежуточной цели к другой.

Планирование в пространстве планов допускает наличие достаточно эффективных эвристических алгоритмов поиска решения. Для использование этого метода требуется определь промежуточные цели решения

задачи. Для этого разберемся, какие условия на значения вершин должны выполняться в процессе получения доступа.

Предположим, что существует последовательность $S$ переменных $\mathbf{x}_i$, такая что изменение значений вершин в этой последовательности приводит к получению доступа к целевой вершине. Она конечна, поскольку завершается с получением доступа. Поэтому в этой последовательности каждая вершина на каком-то шаге встретится в последний раз. Это значит, что на этом шаге она в последний раз поменяет значение и оно «фиксируется». Порядок вершин, в котором они принимают окончательное значение, назовем *порядком фиксации* (это подпоследовательность $S$). Номер вершины в этом порядке назовем *номером фиксации*.

Для каждой вершины $\mathbf{x}_i$ в последовательности $S$ можно выделить фрагмент с начала последовательности до момента, когда эта вершина встречается впервые. Такой фрагмент $S$ соответствует «получению доступа к нецелевой вершине $\mathbf{x}_i$» и определяет некоторый «локальный» порядок фиксации, который может отличаться от порядка фиксации для всей последовательности $S$: вершины, которые фиксируются в этом порядке, могут быть использованы позже.

Для получения доступа к некоторой вершине из начального состояния с его порядком на вершинах потребуется менять только вершины, которые меньше данной в введенном частичном порядке $\preceq$. Поэтому порядок фиксации для доступа к нецелевой вершине $\mathbf{x}_i$ начального размеченного графа зависимостей $D$ — это порядок фиксации на подграфе, для которого $\mathbf{x}_i$ является корневой.

Следующие утверждения определяют условия, которым должны удовлетворять последние значения вершин.

**Утверждение 7.** *Если $\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_n}$ — порядок фиксации вершин графа зависимости для доступа к целевой вершине $\mathbf{x}_0$, то после фиксации в состоянии $x^j$ вершины $\mathbf{x}_{i_j}, j < n$ нефиксированные вершины $\mathbf{x}_{i_k}, k > j$ достижимы по черному пути от целевой.*

То есть нефиксированные вершины не могли оказаться ненужными для дальнейшего процесса получения доступа.

*Доказательство.* Пусть в нефиксированную вершину $k$ в состоянии $x^j$ нет черных путей из $\mathbf{x}_0$. Значит, она уже не понадобится для получения доступа к целевой вершине. Значит, нам не потребуется менять ее значение. Значит, ее значение стало фиксированным при последнем изменении, противоречие с тем, что она нефиксированная. □

**Лемма 1** (Критерий про черные пути)**.** *Пусть $\mathbf{x}_{i_1}$, $\mathbf{x}_{i_2}, \ldots$, $\mathbf{x}_{i_n}$ — порядок фиксации вершин, приводящий к доступу к целевой вершине, а*

*$x^1, \ldots, x^n$ — состояния в момент фиксации значений этих вершин. Тогда для любого $k \leq n$ и $j > k$ в графе $D(x^k)$ нет черных путей из вершины $\mathbf{x}_{i_j}$ в вершину $\mathbf{x}_{i_k}$.*

*Доказательство.* Утверждение означает, что последнее принимаемое вершиной значение должно выбираться таким образом, чтобы не возникало чёрных путей из вершин с бо́льшим номером фиксации.

Пусть у вершины $f$ фиксировано значение и есть черный путь в неё из нефиксированной вершины $g$. Так как значение $g$ ещё не фиксировано, её ещё надо будет менять, а для этого к ней нужно будет получить доступ. Значит её $val.out(g)$ должен будет состоять только из единиц, в том числе первое ребро черного пути из неё в $f$ должно будет стать зелёным. А для этого нужно изменить соответствующий подобъект, а для этого к нему надо получить доступ. И так далее, все ребра черного пути из $g$ в $f$ придется выполнить, в том числе входящее в $f$, а это значит, что $f$ надо будет изменить, что противоречит тому, что её значение фиксировано. □

**Лемма 2** (О выборе фиксированного значения)**.** *В условиях предыдущей леммы отсутствие черных путей из нефиксированных вершин в вершину $\mathbf{x}_{i_k}$ в состоянии $x^k$ — это в точности условие, что для любого $j > k$ $\mu((\mathbf{x}_{i_j}, \mathbf{x}_{i_k}), x^j) = 1$ .*

То есть все ребра, непосредственно ведущие из вершин, находящимся справа от данной в порядке фиксации, должны быть зелеными, и вершине $\mathbf{x}_{i_k}$ должно быть присвоено значение, удовлетворяющее условиям, приписанным этим ребрам.

*Доказательство.* Проверим по критерию. Разделим множество всех вершин, из которых есть ребра в $\mathbf{x}_{i_k}$ непосредственно, на фиксированные (множество $fix(\mathbf{x}_{i_k})$) и нефиксированные ($notfix(\mathbf{x}_{i_k})$).

Существующие ребра непосредственно из нефиксированных вершин в вершину $\mathbf{x}_{i_k}$ зеленые, так как не должно быть черных путей в $\mathbf{x}_{i_k}$ из нефиксированных вершин. Есть ли черные пути из нефиксированных вершин в $\mathbf{x}_{i_k}$? Если такой путь проходит через $notfix(\mathbf{x}_{i_k})$, то ближайшее к $\mathbf{x}_{i_k}$ ребро (между нефиксированной вершиной и $\mathbf{x}_{i_k}$) зеленое, противоречие.

Если такой путь проходит через $fix(\mathbf{x}_{i_k})$, то на этом пути (он конечен) будет ребро из нефиксированной вершины в фиксированную, оно зеленое по предположению для этой фиксированной вершины, значит путь также не черный. □

Получается, первая фиксированная вершина должна принять такое значение, что все возможные условия, зависящие от нее, выполняются.

Поэтому после фиксации доступ к ней не понадобится. Можно рассматривать подграф зависимостей без неё и уже в нём искать вершину, на которую все условия (на меньшем графе) могут быть выполнены. По сути, именно этим мы и будем заниматься для получения доступа к целевой вершине: последовательно «выкидывать» из графа зависимостей вершины, у которых есть «хорошее» значение, пока не останется только целевая вершина.

После фиксации вершины $v$ и возвращения остальных вершин к начальным значениям на подграфе $D \setminus \{v\}$ возникает индуцированный порядок $\preceq_{fix}$, являющийся подпорядком начального порядка $\preceq$ в том смысле, что если $p \preceq_{fix} k$, то и $p \preceq k$ (по тому же черному пути). Здесь $fix$ — множество фиксированных вершин, состоящее пока из одной вершины $v$, $fix = \{v\}$.

По начальному состоянию мы можем вычислить какой порядок $\preceq_{sub}$ будет на любом подграфе $D$ и по следствию из Утверждения 6 он будет совпадать с порядком $\preceq_{fix}$ на подграфе $D \setminus fix$ после фиксации соответствующих вершин и возвращения остальных вершин к начальным значениям. Поэтому для доступа к любой вершине подграфа нужны в точности те вершины, которые меньше нее в индуцированном порядке (это те вершины, в которые ведут черные пути из данной).

**Следствие 4.** *После фиксации первой вершины $v$ граф зависимостей можно вернуть в состояние, в котором для доступа к нефиксированной вершине $k$ нужны только вершины $p \preceq_{fix} k$.*

*Доказательство.* Следует из Утверждения 6 и Леммы 2. □

То есть после фиксации вершины, поскольку все оставшиеся «хорошо» зависят от нее, мы исключаем ее из рассмотрения и переходим к подграфу.

По Утверждению 7 мы знаем, что в момент фиксации некоторой вершины все нефиксированные вершины еще будут нужны для доступа к целевой. Однако после возвращения нефиксированных вершин к начальным значениям порядок $\preceq_{fix}$ на них может быть таким, что подграф зависимости для целевой вершины по черным ребрам не содержит те вершины, которые раньше были достижимы по черному пути, проходящему через фиксированную вершину. Это значит, что эти вершины на самом деле были фиксированы в ходе выполнения обратной последовательности действий (возвращения нефиксированных вершин к начальным значениям). Рассмотрим вершину $p$, которая при возвращении к начальным значениям окажется недостижимой по черному пути от целевой вершины $\mathbf{x}_0$. В ходе выполнения обратной цепочки действий будет состояние, в котором она поменяет значение в последний раз, и оно останется таким

же в состоянии с начальными значениями нефиксированных вершин, в котором мы видим, что менять ее больше не потребуется. Значит, по Лемме 2 это значение удовлетворяет всем условиям нефиксированных вершин и $p$ необходимо добавить в множество фиксированных вершин $fix$. То есть на обратном ходе нам нужно собрать все вершины, которые принимают значения, удовлетворяющие всем условия от нефиксированных вершин и они войдут в порядок фиксации.

Теперь множество $fix$ действительно содержит все фиксированные вершины, а в подграфе из нефиксированных вершин все сравнимы в целевой в порядке $\preceq_{fix}$.

## 5. Дерево порядков

До этого момента мы предполагали, что известна последовательность изменений значений переменных, которая является решением задачи проверки доступа. Были выяснены необходимые условия получения доступа, которые мы используем для нахождения этой последовательности. Будем использовать следующую структуру.

**Определение 2.** Деревом порядков по графу $D$ *назовем упорядоченное дерево, удовлетворяющее следующим условиям:*

- *узлами дерева являются упорядоченные подмножества вершин графа зависимостей $D$;*
- *если узлом дерева является набор длины $n$, то этот узел имеет $n-1$ дочерний узел, каждый из которых связан со своим элементом из первых $n-1$ элементов данного узла;*
- *последним элементом набора $k$-ого дочернего узла является $k$-й элемент набора родительского узла;*
- *последним элементом набора корневой вершины дерева является целевая вершина $t$.*

Схематично узел дерева порядков представлен на рис. 4.

Чтобы избежать терминологического замешательства, будем вершины дерева порядков называть узлами или наборами, а слово «вершина» использовать только для элементов $\mathbf{x}$ вершин графа зависимостей. Листья дерева порядков — наборы, состоящие из одной вершины. Последнюю вершину набора будем называть *целевой* вершиной этого набора. Упорядоченный набор $N = \langle \mathbf{x}_{N1}, \ldots, \mathbf{x}_{Nk} \rangle$ вершин $D$ *реализуем*, если для любой его вершины $\mathbf{x}_{Ni}$ существует допустимое значение, такое что $\mu(\mathbf{x}_{Nj}, \mathbf{x}_{Ni}) = 1$ для всех $j > i$. *Предками* узла $N$ дерева порядков будем

называть все элементы последовательности родительских узлов вплоть до корня (у каждого узла только один родитель).

*Фиксированными вершинами узла $N$ для его дочернего узла $Ni$* назовем все вершины $\mathbf{x}_{Nj}$, $j < i$ узла $N$. *Фиксированными вершинами дерева порядков для узла $N$* назовем все фиксированные вершины всех предков $N$. Это множество обозначим $F(N)$. Если по контексту понятно, какой узел текущий, обозначим просто $F$.

Узел дерева порядков *согласован*, если не содержит фиксированные вершины дерева порядков. Дерево порядков назовем *согласованным*, если все его узлы согласованы. Узлу $N$ дерева порядков соответствует текущий порядок вершин $\preceq_{F(N)}$, индуцированный порядком $\preceq$ на подграфе $D \setminus F(N)$.

Набор $N$ дерева порядков *возрастающий*, если для любого $i \leq k$ верно $\mathbf{x}_i \preceq_{F(N)} \mathbf{x}_k$ (все вершины в индуцированном порядке меньше последней) и он содержит все вершины $\mathbf{x}_i \preceq_{F(N)} \mathbf{x}_k$.

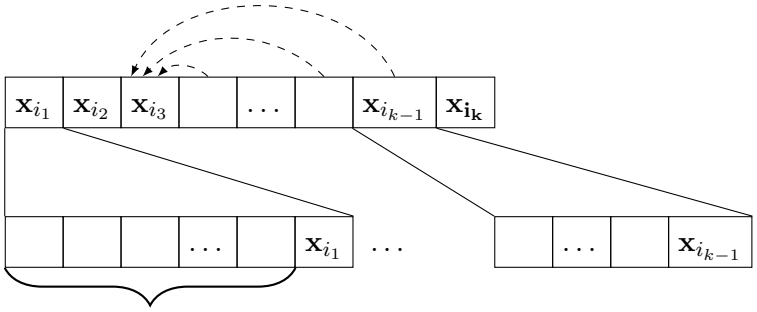



Рис. 4. Схематичное представление узла дерева порядков.

*Пояснение.* Каждый узел дерева порядков — это порядок фиксации для доступа к последней вершине этого узла. Условие реализуемости набора — это в точности условие выбора фиксированного значения по Лемме 2 в терминах дерева порядка: значение фиксируемой вершины должно быть выбрано таким образом, чтобы условия доступа к нефиксированным вершинам выполнялись (все нефиксированные вершины в узле находятся справа от текущей). Согласованность нужна, чтобы не пытаться изменить те вершины, которые уже фиксированы ранее. Набор возрастающий, поскольку граф зависимостей упорядочен и для получения доступа к вершине $k$ могут понадобиться только вершины меньшего порядка в текущем графе зависимостей. Листовые вершины соответствуют тем вершинам графа зависимостей, которые доступны в начальном состоянии.

**Определение 3.** *Дерево порядков назовем* корректным*, если (1) для каждого его узла набор вершин реализуем и (2) для любого узла набор его вершин возрастающий.*

## 5.1. Критерий корректности дерева порядков

**Теорема 1.** *Корректное и согласованное дерево порядков, построенное по графу $D(x^0)$, существует тогда и только тогда, когда доступ к целевой переменной $\mathbf{x}_0$ может быть получен из начального состояния $x^0$.*

*Доказательство.* Пусть задано корректное согласованное дерево порядков. Покажем, что в этом случае целевая переменная достижима. Сначала опишем цепочку изменений вершин, которую мы строим по дереву порядков, а ниже покажем, что все действия в ней допустимы.

Через $v$ будем обозначать узел дерева порядков. Обозначим вектор дочерних узлов узла $v$ Через $d(v) = \langle d(v)_0, \ldots, d(v)_{N(v)} \rangle$. В узле $v$ им соответствуют вершины $(v_0, \ldots, v_i, \ldots v_{N(v)})$. Через $v.t$ обозначим последнюю (целевую) вершину набора $v$, а через $val(v.t)$ — её значение. Для листового узла $v$ $d(v) = \varnothing$, а набор $v$ состоит только из $v.t$. Через $F$ обозначается текущее множество фиксированных вершин.

Весь процесс описывается рекурсивным алгоритмом GAESBR (Алгоритм 5.1). Название является аббревиатурой Get Access, Edit the Subject, get Back the Rest.

**Algorithm 5.1** Получение доступа к последней вершине узла $v$ дерева порядков $T$.

**function** GAESBR($T$, $v$, *newvalue*)
*Изменение значения вершины $v.t$ на newvalue.*
$history \leftarrow \varnothing$
**if** $d(v) \neq \varnothing$ **then**
 **for** $i \leftarrow 0$ **to** $N(v)$ **do**
  выбрать новое значение $newval_i$ для $i$-ой вершины согласно Лемме 2
  добавить в стек *history* тройку $\langle v, i, val(v_i) \rangle$
  GAESBR($T$, $d(v)_i$, $newval_i$)
$val(v.t) \leftarrow newvalue$
*Отменить все остальные шаги*
**while** $history \neq \varnothing$ **do**
 извлечь из стека последнее действие $\langle v, i, oldval \rangle$
 $val(v_i) \leftarrow oldval$
**return**

Будем присваивать вершинам значения в порядке обхода дерева в глубину: доходим до доступной вершины и присваиваем ей такое значение, которое нужно в текущем узле дерева порядков. После этого все действия, совершенные для получения доступа к этой вершине, отменяются: значения в обратном порядке возвращаются на предыдущие.

Если $v$ — листовой узел, то $d(v)$ пусто и мы сразу присваиваем вершине $v.t$ значение, нужное в родительском узле — такое, благодаря которому набор родительского узла реализуем: что если ребро из каждой вершины, которая в родительском наборе находится справа от $v.t$, существует, то соответствующая ему метка должна выполняться (иметь значение 1). То есть все условия на эту вершину должны выполняться, и ее значение локально фиксируется: до конца этого родительского узла.

Если $v$ не листовой узел, то мы последовательно присваиваем нецелевым вершинам набора новые значения в соответствии с реализуемостью набора $v$. Когда мы доходим до целевой вершины узла, у нас есть к ней доступ. Почему: рассмотрим все вершины, от которых она зависит.

Для получения доступа к вершине $v.t$ по определению индуцируемого порядка нужны только вершины $\preceq_{F(v)} v.t$. Все эти вершины входят в узел $v$ и в ходе выполнения алгоритма для узла $v$ принимали значения согласно реализуемости набора $v$, а значит от них $v.t$ зависит с меткой 1. Значит, $v.t$ доступна. Присваиваем ей значение в соответствии с реализуемостью родительского узла.

А теперь нам нужно вернуть все остальные вершины узла $v$ к тем значениям, которые у них были в начальном состоянии. В родительском узле $v.t$ — первая фиксированная вершина на подграфе из $v.t$ и нефиксированных вершин этого узла с индуцированным на них порядком. По следствию 4 систему можно привести в такое состояние, что порядок на нефиксированных вершинах $\preceq_F$ таков, что для получения доступа к любой нефиксированной вершине понадобятся только вершины меньше нее в порядке $\preceq_F$ на подграфе без $v.t$.

Значит, после каждого возвращения значений нефиксированных вершин к начальным доступ к каждой из них снова зависит только от вершин с меньшим порядком на текущем подграфе, что является предположением для начала действия функции.

Таким образом, с помощью описанной функции по корректному согласованному дереву порядков мы получим доступ к целевой вершине.

Пусть теперь известно, что целевая переменная достижима. Тогда существует порядок фиксаций, приводящий к доступу к целевой вершине. Этот порядок и будет набором корневой вершины дерева. Значения, которые вершины принимают в момент фиксации, соответствуют реализуемому набору по лемме 2. Аналогично, и для каждой вершины графа

существует порядок фиксации, приводящий к доступу к ней. Вопрос в том, какие вершины используются в этом порядке.

По утверждению 6 после получения доступа к некоторой вершине корневого набора остальные вершины можно откатывать в начальное состояние, так что частичный порядок на нефиксированных вершинах является подпорядком начального порядка. Поэтому после получения доступа к каждой из вершин набора корня можно произвести обратную последовательность действий для возвращения нефиксированных вершин к начальным значениям. И после этого для доступа к каждой вершине набора корневого узла понадобятся только вершины индуцированным порядком меньше нее. Это условие означает, что дочерние узлы корня возрастающие.

Оно же значит, что узлы согласованы: в них используются только нефиксированные вершины.

Все это верно, если к каждому узлу дерева относиться как к корневому на подграфе, содержащем вершины этого узла. Таким образом, дерево порядков достраивается до листов и является корректным и согласованным. □

## 5.2. Построение дерева порядков

Задано множество вершин $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ и для каждой вершины $\mathbf{x}_i \in \mathbf{x}$ задан набор $S_{i1}, S_{i2}, \ldots, S_{ik_i}$ подмножеств $\mathbf{x}$. Порядком на $\mathbf{x}$ назовем взаимно-однозначного отображение $f : \mathbf{x} \to \{1, \ldots, n\}$. Через $succ_f(\mathbf{x}_i)$ обозначим множество вершин, которые находятся правее $\mathbf{x}_i$ в порядке $f$, то есть $succ_f(\mathbf{x}_i) = \{\mathbf{x}_j \in \mathbf{x} \mid f(\mathbf{x}_j) > f(\mathbf{x}_i)\}$. Требуется найти порядок $f$ при котором выполняется следующее условие:

$$\forall i \leq n \; \exists m \leq k_i \;\; succ_f(\mathbf{x}_i) \subseteq S_m. \tag{2}$$

То есть для каждой вершины $\mathbf{x}_i$ найдется набор $S_{im}$, содержащий все вершины, которые в последовательности $f$ находятся после $\mathbf{x}_i$. Порядок $f$, удовлетворяющий условию (2) будем называть *корректным*.

Здесь множества $S_{ij}$ соответствуют возможным значениям $val.in(\mathbf{x}_i)$, где единицы стоят в разрядах, отвечающих вершинам, входящим в $S_{ij}$.

**Утверждение 8** (Сдвиг влево)**.** *Пусть задано корректное отображение $f$. Если существует вершина $\mathbf{x}_i$, для которой:*

1) *$f(\mathbf{x}_i) > 1$ ($\mathbf{x}_i$ имеет левого соседа в порядке $f$) и*

2) *найдется набор $S_{im}$, такой что $\{f^{-1}(f(\mathbf{x}_i) - 1)\} \cup succ_f(\mathbf{x}_i) \subseteq S_{im}$ (набор $S_{im}$ содержит левого соседа $\mathbf{x}_i$ и все вершины, которые стоят справа от $\mathbf{x}_i$), то*

*порядок $f'$, отличающийся перестановкой $\mathbf{x}_i$ с его левым соседом, является корректным.*

*Доказательство.* Множества $succ_{f'}(\mathbf{x}_j)$ для всех вершин с номерами $j \in (1, \ldots, f(\mathbf{x}_i) - 2, f(\mathbf{x}_i) + 1, \ldots n)$ остались прежними.

Множество $succ_{f'}(\mathbf{x}_i) \subseteq S_{im}$ по условию, значит для $\mathbf{x}_i$ выполняется условие корректности $f'$.

Для $succ_{f'}(f^{-1}(f(\mathbf{x}_i) - 1))$ можно взять то же множество $S'$, которое использовалось в $f$: оно покрывает все вершины, находящиеся правее в новом порядке. □

**Алгоритм нахождения корректного порядка** Если для каждого $i$ задано единственное множество $S_{i1}$, то есть $k(i) = 1$, то задача в точности является задачей топологической сортировки ориентированного графа и решается за линейное время [9].

Для произвольного набора множеств применяем аналогичный алгоритм. Находим все вершины $\mathbf{x}_i$, для которых существует набор $S_{im}$, такой что $(\mathbf{x} \setminus \{\mathbf{x}_i\}) \subseteq S_{im}$. Они будут первыми в искомом порядке $f$. «Удаляем их из рассмотрения» и решаем аналогичную задачу на меньшем множестве вершин. Таким образом строится корневой порядок фиксации вершин. После этого надо перейти на подграф для доступа к каждой вершине этого порядка и найти порядок фиксации на нем, проверив аналогичные условия. Проделать это с каждым порядком.

Если входными данными задачи считать множество вершин $\mathbf{x}$ и наборы множеств $S_{ij}$, и предположить, что множество $S_{ij}$ проверяется на полноту за $O(1)$, то сложность алгоритма для нахождения порядка одного узла $O\left((\sum_{i=1}^{n} k_i)^2\right)$ — квадратичная от количества подмножеств $S_{ij}$. Возьмем одну вершину, проверим все ее множества на полноту, перейдем к другой. В худшем случае нам придется проверить все $\sum_{i=1}^{n} k_i$ множеств. Если одна вершина подойдет, то ее берем первой (без ограничения общности она имеет номер 1) и на второе место нам надо проверить уже $\sum_{i=2}^{n} k_i$, и так далее. Всего проверок будет не больше квадрата от числа подмножеств.

Трудность в том, что подмножеств $S_{ij}$ может быть много. Если от вершины $\mathbf{x}_i$ зависят все остальные вершины, то множеств $S_{ij}$ может быть $2^N$, где $N+1$ — число вершин (если есть все возможные пересечения множеств, удовлетворяющих условиям вершин). Однако вообще говоря нам не требуются все множества: если частично упорядочить все множества

для одной вершины по вложению, достаточно из каждой цепи взять максимальный элемент (такое множество, очевидно, можно использовать в наборе вместо вложенного в него). Тогда их не больше, чем длина максимальной антицепи $C_N^{[N/2]}$, что все еще экспоненциально много.

Напомним, что мы считаем, что сами множества $S_{ij}$ уже построены, но вообще говоря их построение может внести дополнительную сложность. Вопрос в том, как строятся множества $S_{ij}$, нас до сих пор не интересовал. Вспомним, что они связаны с пересечением множеств доступности $A_{ij}$, которые как-то задаются, и сложность задания этих множеств влияет на количество $S_{ij}$ и на сложность выполнения алгоритма: построение пересечений множеств доступности и поиск значения, принадлежащего пересечению нескольких множеств доступности, может быть трудоемкой процедурой при некоторых способах задания множеств (например, если множества заданы как решение уравнений или как удовлетворяющие некоторому предикату).

## 5.3. Сложность построения дерева порядков в частных случаях графа зависимостей

Если множества $A_{ij}$ — отрезки, то даже в предположении, что $\mathbf{x}_i$ зависит от всех остальных вершин, экспоненты в количестве $S_{ij}$ не возникает, поскольку классов пересечения $n$ отрезков на прямой не более $2n$, значит количество множеств $S_{ij}$ не больше $2N$, где $N$ — число вершин. Таким образом, экспоненциальная сложность алгоритма построения дерева порядков может не являться препятствием для практического применения.

Если множества $A_{ij}$ в исходных данных задаются, например предикатами, то на «сложность» множеств влияет сложность формул. Отрезок задается простой формулой $(\mathbf{x}_i > a)\&(\mathbf{x}_i < b)$. Но если предикаты могут быть какими угодно, понятно, что формула может быть сложной и алгоритм с ней будет работать ожидаемо долго.

Входные данные могут оказаться простыми и в другом смысле. Если граф зависимостей $D$ является ациклическим ориентированным графом (то есть упорядочивается так, что обратных ребер вообще нет), то доступ к корневой всегда можно получить. Предположим, что граф — полное дерево, то есть содержит все возможные ребра (если в исходном графе отсутствовало некоторое ребро, то добавим фиктивное ребро, метка которого тождественно равна 1). Тогда вершины нумеруются $1 \preceq 2 \ldots \preceq N$. Можно выбрать порядок фиксации для каждой вершины $k$ следующим образом: $k-1, k-2, \ldots, 2, 1, k$. В этом порядке единственное существующее ребро, ведущее из вершины справа — ребро из вершины $k$, а значит всегда существует значение, удовлетворяющее условию этого ребра. Несложно проверить, что количество действий в последовательности из-

менений значений вершин не превосходит $O(2^N)$, где $N$ — число вершин, и эта оценка является точной.

## 6. Заключение

В работе рассмотрена задача проверки невозможности получения доступа к выделенному объекту информационной системы с атрибутивной политикой безопасности. Показано, что при некоторых ограничениях на структуру правил доступа, задача сводится к задаче интеллектуального планирования, что дает возможность применять для ее решения известные эффективные методы планирования с использованием эвристических характеристик задачи [7].

В данной работе доказан критерий возможности получения доступа в зависимости от входных данных задачи. Вводится понятие дерева порядков — структуры, с помощью которой проверяется выполнение критерия. Предложен алгоритм получения доступа с помощью дерева порядков.

Предложен алгоритм, отвечающий на вопрос «возможно ли получение доступа к заданному объекту». Этот алгоритм имеет экспоненциальную оценку сложности в зависимости от числа объектов информационной системы. При этом показано, что при выполнении некоторых дополнительных условий на структуру графа зависимостей или множеств доступности объектов, которые соответствуют правилам политики безопасности информационной системы, сложность этого алгоритма может быть уменьшена до полиномиальной.

Возможными направлениями дальнейших исследований могут быть: поиск кратчайшей последовательности действий при заданном корректном дереве порядков, поиск оптимального порядка вершин в узле дерева порядков в процессе построения корректного дерева, оценка алгоритмической сложности проверки корректности дерева порядков.

## Список литературы


[1] А.В. Галатенко, Плетнёва В.А., “Выразимость моделей безопасности take–grant и невлияния в модели СВАС”, *Программная инженерия*, № 1, 40–46.

[2] Sergey Afonin, Antonina Bonushkina., “Validation of safety-like properties for entity-based access control policies.”, *Advances in Soft and Hard Computing*, Advances in Intelligent Systems and Computing, **889**, Springer International Publishing, 259–271.

[3] Sandra Alves, Maribel Fernández, “A graph-based framework for the analysis of access control policies”, *Theoretical Computer Science,*, **685**, 3–22.

[4] Clara Bertolissi, Maribel Fernández., “A metamodel of access control for distributed environments: Applications and properties”, *Information and Computation*, **238**, 187–207.

[5] Avrim L Blum, Merrick L Furst, "Fast planning through planning graph analysis.", *Artificial intelligence*, **90**:1-2, 281–300 ..

[6] Tom Bylander, "The computational complexity of propositional STRIPS planning.", *Artificial Intelligence*, **69**:1-2, 165–204.

[7] Malik Ghallab, Dana Nau, Paolo Traverso, *Automated Planning: theory and practice*, Elsevier.

[8] Vincent C Hu, D Richard Kuhn, David F Ferraiolo, Jeffrey Voas, "Attribute-based access control", *Computer*, **48**:2, 85–88.

[9] A. B. Kahn, "Topological sorting of large networks", *Communications of the ACM*, **5**:11, 558–562.

## Attribute-based access control policy analysis using automated planning technique

**Afonin S.A., Bonushkina A. Yu.**

The paper considers the problem of testing the possibility of a user of an information system gaining access to the selected object with a given attribute-based security policy. It is shown that under some restrictions on information system model and policy rules, this task is reduced to the task of automated planning. There is a tree structure determined, the construction of which corresponds to planning in the space of plans and allows to take into account the specifics of the problem when constructing heuristic algorithms for checking access. It is proved that the existence of such a structure is a necessary and sufficient condition for the possibility of getting an access to the target.

This work was supported by RFBR Grant 18-07-01055.

*Keywords:* ABAC, access control, automated planning.

# Искусственный интеллект: проблемы и перспективы

Кудрявцев В.Б.[1], Козлов В.Н.[2], Рыжов А.П.[3],
Мазуренко И.Л.[4], Боков Г.В.[5], Петюшко А.А.[6]

В работе излагаются результаты дискуссии на тему проблем и перспектив искусственного интеллекта, состоявшейся на кафедре математической теории интеллектуальных систем 14 октября 2020 года. Тематика дискуссии восходит к классическим работам А. Тьюринга «Может ли машина мыслит» и Дж. фон Неймана

[1] *Кудрявцев Валерий Борисович* — заведующий каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, профессор, e-mail: ilaky@bk.ru.

Kudryavtsev Valeriy Borisovjch — head of the chair, professor, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

[2] *Козлов Вадим Никитович* — профессор каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: vnkozlov@mail.ru.

Kozlov Vadim Nikitovich — professor, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

[3] *Рыжов Александр Павлович* — профессор каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: ryjov@mail.ru.

Ryjov Alexander Pavlovich — professor, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

[4] *Мазуренко Иван Леонидович* — с.н.с. каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: ivan@mazurenko.ru.

Mazurenko Ivan Leonidovich — senior researcher, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

[5] *Боков Григорий Владимирович* — доцент каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ и зав. лабораторией математических проблем искусственного интеллекта, e-mail: bokovgrigoriy@gmail.com.

Bokov Grigoriy Vladimirovich — associate professor, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems, head of Laboratory of Mathematical Problems of Artificial Intelligence.

[6] *Петюшко Александр Александрович* — к.ф.-м.н., научный эксперт, руководитель команды видео-интеллекта Московского исследовательского центра Хуавэй, e-mail: petyushko@yandex.ru.

Petyushko Alexander Alexandrovich — Candidate of Physical and Mathematical Sciences, scientific expert, head of the video intelligence team of Huawei Moscow Research Center.

«Вычислительная машина и мозг», которые возникли на заре становления кибернетики как науки. С тех пор дискуссии на тему «Может ли машина мыслить» то возникали, то затухали и в понимании этого вопроса особой ясности они не вносили. В последние годы, в связи с мощным развитием технологической базы вычислительных систем, тематика стала вновь актуальной. Вместе с научной базой, наработанной в теории искусственного интеллекта и прикладных программ в этой области, возникло множество работ спекулятивного типа, готовых объявить любое устройство со встроенными в его систему управления тривиальными алгоритмическими добавками «системой искусственного интеллекта». В работе делается попытка отделить «зерна от плевел», изложив несколько точек зрения по этому вопросу.

**Ключевые слова:** искусственный интеллект, машинное обучение, нейронные сети.

## Профессор, д.ф.-м.н. Козлов В. Н.

В дискуссии я занимал позицию скептика в отношении нейронных сетей. При этом я осознаю важность и значимость этой модели для многочисленных приложений. И, тем не менее, полагаю, что это тупиковый путь как для моделирования нервной системы, так и для распознающих систем. В обоснование своей точки зрения я приведу некоторые соображения по этим двум пунктам.

1) Нейронные сети как модель для реальной нервной системы. Это направление основывается на примерно такой логике: нервная система – конечное множество нейронов. Если мы по возможности точнее воспроизведем в некоторой формальной модели свойства реального нейрона, а затем исследуем все возможные соединения этих формальных нейронов в сети, то в их свойствах чудесным образом проявятся интересные и загадочные свойства реальной нервной системы. Здесь страдает сама логика подхода: выделить «кирпичик» , основной элемент, т.е. нейрон, из которых состоит объект (нервная система), и затем, отправляясь от «кирпичика», соединяя их в разных комбинациях, пытаться прийти к проявлению свойств мозга. Давайте повторим эту логику в несколько ином варианте: возьмем в качестве «кирпичика» не нейрон, а молекулу. Ведь все нейроны, а значит и нервная система, состоят из молекул, и их конечное множество. Воспроизведем в формальной модели свойства молекул как можно точнее, и будем соединять эти модели молекул в своеобразные «молекулярные сети» вместо нейронных сетей. Получим , ясно, в лучшем случае молекулярную физику, но не свойства мозга. То есть «молекулярный уровень» , скорее всего, не тот уровень, который требуется для изучения , например, алгоритмов мозга. Но можно пред-

положить такое же и про «нейронный уровень». Еще одна аналогия: если выделить «кирпичик» как элемент, из которых строят здания, то, ясно, архитектурный облик здания (аналог функциональных механизмов мозга) определяется идеями искусства, истории, философии и пр., и лишь в малой степени свойствами кирпича.

2) Нейронные сети как распознающие системы. На мой взгляд, работа нейронных сетей по большому счету имеет характер перебора, со всеми издержками таких алгоритмов. Я поясню свою мысль на следующем примере. Пусть мы хотим использовать некоторую функцию $f(x)$, определяя ее значения для некоторых значений аргументов. К сожалению, у нас нет формулы для задания функции, но есть достаточно большое множество примеров ее значений при некоторых значениях аргументов (это своеобразное обучающее множество). Можно представить эти значения как множество точек в некоторой системе координат. Мы дополняем это обучающее множество системой отрезков прямых между соседними точками, получая , тем самым , ее предположительный график. Теперь для любого значения аргумента можно получить «значение» функции, и, нетрудно видеть, по сути, перебором и интерполяцией. Разумеется, в нейронных сетях все и сложнее, и более громоздко: не плоскость, а многомерное пространство, интерполяции тоже гораздо замысловатее, и пр. Но, все же, очевидно, было бы лучше иметь (и искать!) формулу для функции.

Кроме того, когда говорят о поразительных успехах нейронных сетей, например, в распознавании изображений, большей частью за кадром остается вопрос о том, при каких условиях эти успехи достигнуты. Поясню сказанное на примере перцептрона «Марк-1» Ф.Розенблатта. Я читал описание экспериментов с перцептроном , там говорилось, что после предъявлений в обучении с алгоритмом «с учителем» 20-40 начертаний каждой буквы , правильность распознавания букв достигала почти 100%. Получается, что еще в конце 50-х - начале 60-х годов проблема распознавания, по крайней мере, фигур была решена ?! Но другое выясняется, если начинаешь вчитываться в те условия, в которых проходили эксперименты по распознаванию. На квадратный ячеистый экран («сетчатка») проецировалась фигура так, чтобы полностью его заполнить, по крайней мере , по высоте (то есть, получается , фигуры приводились к «каноническому» размеру). Фигура на экране располагалась так, чтобы ее верх был вверху, низ – внизу, правое – справа, левое – слева. Но с какой стати предполагается, что это известно? Мы еще не распознали фигуру, но уже знаем, где у нее верх, где низ, и т.д. Подсказка «доброго дяди»? Порочный круг? Если же считать, что «подсказок» нет, то таких замечательных результатов уже далеко не будет. Этим же в немалой мере грешат и нынешние нейронные сети. В какой то мере к

недостаткам нейронных сетей можно отнести и малочисленность математических результатов для них, т.е . это эвристика. На мой взгляд, главных результатов - два: теорема Колмогорова, которую сейчас трактуют как «накрывающую» сверху всю эту модель, и теорема Новикова про перцептрон. Но сам А.Н.Колмогоров, доказывая теорему (середина 50-годов), вряд ли вообще знал о существовании перцептронов, они тогда только начинались.

Представленный текст выступления местами, может быть, категоричен и полемичен. Не считаю это бедой, в дискуссии, полагаю это приемлемо.

## Профессор, д.т.н., к.ф.-м.н. Рыжов А. П.

Нас почему-то не удивляет, что калькулятор считает быстрее и точнее человека, компьютер – быстрее и точнее решает, например, дифференциальные уравнения и многое другое, к чему мы привыкли и просто не замечаем. Вот теперь спецпроцессор в виде искусственной нейросети распознаёт лица быстрее и точнее человека. Ну и что? Какое это отношение имеет к интеллекту? Наверно, для почти любой задачи можно придумать алгоритм, работающий быстрее и точнее человека. Даже научившись решать сотни таких задач, мы из полученных элементов не соберем пазл под названием ИИ. Поэтому все эти «достижения» (безусловно, очень важные и интересные для кого-то) к теме нашей дискуссии не имеют почти никакого отношения.

К сожалению, вокруг ИИ сейчас возникло много хайпа и шальных денег (по разным оценкам, инвестиции в ИИ уже составят сотни миллиардов долларов[1]). Это притягивает множество проходимцев, не очень умных, но шумных, создающих ложные ожидания. Все больше появляется мнений, что ИИ – это обман. Например, «AI heading back to the trough. The expectations over artificial intelligence (AI) are becoming too inflated. AI will indeed change everything, but not any time soon»[2] или «Inflated Expectations: Artificial Intelligence Still Depends on Humans»[3]. Компании – лидеры рынка меняют свою политику в отношении ИИ: Facebook сократил подразделение M, занимающиеся ИИ[4], IBM Watson health сократил 70% сотрудников[5], Cambridge Analytica объявила о банкротстве[6]; много

[1] https://hightech.plus/2020/01/15/investicii-v-ii-startapi-ssha-dostigli-rekordnih-visot

[2] https://www.networkworld.com/article/3206313/internet-of-things/ai-heading-back-to-the-trough.html

[3] https://techonomy.com/2016/07/27222/

[4] https://www.theverge.com/2018/1/8/16856654/facebook-m-shutdown-bots-ai

[5] https://www.theregister.co.uk/AMP/2018/05/25/ibms_watson_layoffs/

аналогичных фактов без труда можно найти в интернет. Означает ли это, что наступает новая зима ИИ? Не хотелось бы.

В такие моменты правильно обратиться не к мнению не разговорчивых блогеров, юристов и бухгалтеров, а к видению отцов-основателей ИИ. А они никогда не писали об ИИ как о самодостаточном самодумающем чёрном ящике с собственным сознанием. Так, Эшби рассуждал об усилении интеллектуальной силы человека [1], Ликлайдер писал о симбиозе человеческого и компьютерного интеллектов [2]. Компании-лидеры рынка начинают говорить в подобных терминах: IBM вводит понятие дополненного интеллекта (Augmented Intelligence [3]) близкое к пониманию Эшби; McKinsey ввело понятие Automation of knowledge work [4], близкое к пониманию Ликлайдера.

Такие человеко-компьютерные системы гибридного интеллекта, скорее всего, и есть выход из складывающегося кризиса ИИ. Для разработчиков это означает прежде всего переосмысление задач. С такими постановками задач и сценариями использования систем гибридного интеллекта можно ознакомиться в недавно вышедшей книге [5].

Хочется отметить, что это направление включается в дорожные карты ведущих экспертных и грантообразующих организаций. Так, Национальный научный фонд США выделил 10 прорывных направлений[7], первое из которых – про гибридный интеллект[8]; Управление перспективных исследований Министерства обороны США определило третью волну ИИ (AI Next Campaign[9] как гибридный интеллект (Towards this end, DARPA research and development in human-machine symbiosis sets a goal to partner with machines) и уже инвестирует в это направление 2 миллиарда долларов[10]; в утвержденном недавно стратегическом плане развития ИИ США[11] стратегия №2 – про гибридный интеллект (Strategy 2: Develop effective methods for human-AI collaboration. Increase understanding of how to create AI systems that effectively complement and augment human capabilities).

[6] https://ca-commercial.com/news/cambridge-analytica-and-scl-elections-commence-insolvency-proceedings-and-release-results-3

[7] https://www.nsf.gov/news/special_reports/big_ideas/index.jsp

[8] https://www.nsf.gov/news/special_reports/big_ideas/human_tech.jsp

[9] https://www.darpa.mil/work-with-us/ai-next-campaign

[10] https://www.darpa.mil/news-events/2018-09-07

[11] https://www.whitehouse.gov/wp-content/uploads/2019/06/National-AI-Research-and-Development-Strategic-Plan-2019-Update-June-2019.pdf

## Старший научный сотрудник, к.ф.-м.н. Мазуренко И. Л.

Еще на рубеже 50-60 годов 20 века выдающимися советскими учеными А. Н. Колмогоровым [6] и В. И. Арнольдом [7] была доказана универсальная теорема представимости для непрерывных функций, а именно, что любая многоместная непрерывная функция на компактном носителе может быть представлена в виде конечной композиции непрерывных функций одной переменной и бинарной операции сложения.

Эта теорема тесно связана с 13-ой проблемой Гильберта [8] и может рассматриваться как фундаментальный теоретический базис для всей теории нейронных сетей.

В 1989 Д. Цыбенко доказал универсальную теорему аппроксимации [9] для многослойных нейросетей прямой связи (без циклов), а именно, что любая непрерывная функция многих переменных на компактном носителе может быть приближена с любой заранее заданной точностью 2-х слойным перцептроном (полносвязной сетью с одним скрытым слоем) с монотонно возрастающей ограниченной функцией активации. Та же теорема была независимо доказана в [10].

Нелинейности в современных нейросетях, как правило, представлены кусочно-линейной функцией активации ReLU=max(0,x) [11]. Тем самым, современные нейросети - это всего лишь эффективный способ представления непрерывных функций путем их приближения кусочно-линейными функциями общего вида, а их широкое применение связано исключительно с наличием большого объема данных для обучения этих нейросетевых интерполяторов и большими вычислительными ресурсами, представляемыми современными графическими ускорителями (GPU). Интерполяционные свойства нейросетей обладают рядом фундаментальных проблем, связанных с их устойчивостью к аномальным данным [12], обобщающей способностью и проблемами переобучения/отсутствия сходимости [13].

Никакой прямой связи данных "нейросетевых" аппроксиматоров с биологией мозга и тем более с т.н. "искусственным интеллектом" (возможность существования которого никак не связана с теорией нейронных сетей), автором не усматривается.

## Доцент, к.ф.-м.н. Боков Г. В.

Успехи в развитии технологий искусственного интеллекта сместили сегодня вектор развития искусственного интеллекта в сторону алгоритмов решения отдельных и частных задач. Сложность решаемых задач и ограниченность вычислительных ресурсов заставляют детализировать

и подстраивать алгоритмы, затачивая их не только под конкретную задачу, но и под конкретный тип входных данных. Как это, например, происходит сегодня с задачами распознавания изображений. Такая узкая заточенность и гонка за производительностью в итоге упускают из вида универсальность — одно из главных отличительных свойств человеческого интеллекта. Вопрос о том, следует ли относить универсальность к базовым принципам искусственного интеллекта, является спорным. В то же время, некоторые свойства универсальности определенно должны закладываться в системы искусственного интеллекта. В этой связи представляется уместным рассмотреть понятие искусственного интеллекта в более широком смысле.

Искусственный интеллект является предметом междисциплинарных исследований, как фундаментальных, в которых участвуют математиками, биологи, психологи, лингвисты, философы, так и прикладных, где технологии искусственного интеллекта разрабатывают специалисты в области информатики, вычислительной техники, программирования, робототехники и так далее. Со времен Джона фон Неймана в основы представления об искусственном интеллекте закладывались принципы работы естественных когнитивных систем и человеческого мозга. Современные достижения в области сканирования мозга позволяют сегодня учёным в области нейронаук исследовать работу мозга в режиме реального времени. Это даёт основание надеяться, что понимание принципов работы мозга человека послужит толчком к созданию искусственного интеллекта следующего поколения.

Все, что связано с изучением человеческого мозга, является одним из безусловных приоритетов современной науки. Все ведущие научные страны создают свои собственные программы исследований мозга. Европейцы работают над десятилетним мегапроектом «Human Brain Project»[12]. Европейский союз выделил более миллиарда долларов на разработку компьютерной модели человеческого мозга. В 2013 году президент США выдвинул проект «Brain Initiative»[13], который он назвал величайшим вызовом XXI века. Первоначальное финансирование этого проекта составляет более 100 миллионов долларов. Китай, между тем, вкладывает в свою программу в несколько раз больше, чем США. И такие программы существуют во многих странах мира. Сегодня инвестиции в науку о мозге так же важны, как и инвестиции в ядерную энергетику, космические исследования, альтернативные источники энергии и расшифровку генома. Нобелевский лауреат Джеймс Уотсон сказал: «мозг — это последний и самый грандиозный рубеж, самая сложная вещь, которую мы когда-либо открывали в нашей вселенной».

[12] `https://www.humanbrainproject.eu`

[13] `https://braininitiative.nih.gov`

В мае 2013 года Глобальный институт McKinsey опубликовал доклад «Прорывные технологии: достижения, которые изменят жизнь, бизнес и мировую экономику». В докладе предложено 12 технологий, которые могут привести к масштабным экономическим преобразованиям в ближайшие годы. Потенциальный экономический эффект от внедрения этих технологий оценивается в размере от 14 до 33 триллионов долларов в год к 2025 году. Среди этих технологий выделена автоматизация умственного труда как одна из наиболее важных прорывных технологий. Достижения в области методов искусственного интеллекта, машинного обучения и автоматизированного доказательства теорем позволяют автоматизировать многие процессы, связанные с умственным трудом человека, которые долгое время считались невозможными или непрактичными для компьютерной реализации. К 2025 году средства автоматизации умственного труда могут иметь экономический эффект в размере от 5,2 до 6,7 триллионов долларов в год.

Автоматизация умственного труда в широком смысле означает использование компьютеров для выполнения задач, которые опираются на комплексный анализ, тонкие рассуждения и творческий подход к решению задач. Она включает в себя такие области, как:

- *Интерактивные системы решения интеллектуальных задач, системы автоматического доказательства и обучающиеся системы* дают компьютерам возможность решать интеллектуальные задачи, используя сложный анализ, сложные логические рассуждения и творческий подход к решению.

- *Методы распознавания образов и машинного обучения, включающие нейронные сети и глубокое обучение,* дают компьютерам возможность делать заключения из шаблонов, распознаваемых в массивах входных данных.

- *Пользовательские интерфейсы и системы обработки естественного языка* дают компьютерам возможность напрямую реагировать на команды и запросы человека.

- *Интеллектуальный анализ данных и многоагентные системы* дают компьютерам возможность обнаруживать закономерности в больших массивах данных и сложных многоагентных системах.

Эти области сейчас определяют тенденции в развитии технологий искусственного интеллекта и вычислительной техники.

Первые попытки создать так называемые «думающие» системы и первые фундаментальные математические теории, предназначенные для их описания, появились в конце 30-х годов XX века благодаря Алану

Тьюрингу, Клоду Шеннону, Норберту Винеру, Джону фон Нейману и другим. Эти системы были необходимы для решения таких задач, как расшифровка сообщений, слежение за движущимися целями, быстрые вычисления и так далее. Даже термин «искусственный интеллект» был придуман только 20 лет спустя Джоном Маккарти. В последующие годы развитие вычислительной техники привело к созданию программ для решения интеллектуальных задач в различных областях. Они имитировали действия эксперта в соответствующей области и, соответственно, назывались экспертными системами. Таких систем очень много. Некоторые из них могут успешно конкурировать с человеком, например, компьютерные шахматы. Такие системы достигают впечатляющих результатов не за счет проникновения в логику человеческих действий, а, так сказать, методом «грубой силы», за счет огромной производительности компьютеров. В настоящее время задачи стали настолько огромными, что компьютеры иногда не могут их решить. Конечно, если появятся более мощные компьютеры, будут решены и эти проблемы. Но разве человек меняет свой образ мышления, подстраиваясь под конкретную задачу, например, для игры в шахматы? Наблюдаемые различия в способе решения задач человеком и компьютером приводят к выводу о том, что говорить об искусственном интеллекте как о технологиях, способных заменить интеллект человека, пока преждевременно.

## Выпускник кафедры, к.ф.-м.н. Петюшко А. А.

Несмотря на отсутствие связи между классическим понятием "сильного искусственного интеллекта"[14] и искусственными нейронными сетями, последние получили большое распространение в современных исследованиях. Например, на данный момент прогресс в области применения нейронных сетей к практическим задачам, в особенности в разрезе компьютерного зрения, просто поражает. Основные классические задачи компьютерного зрения, а именно: 1) Классификация (по входному изображению необходимо предъявить метку класса объекта, находящегося на изображении), 2) Обнаружение (по входному изображению необходимо не просто предъявить метку класса объекта, находящегося на изображении, но и обвести местонахождение этого объекта на изображении прямоугольником), 3) Сегментация (каждый пиксель входного изображения должен быть отнесен к некоторому классу - таким образом, может рассматриваться как уточнение задачи обнаружения на пиксельном уровне) - на данный момент решаются в современных системах исключительно с помощью сверточных нейросетей, первое успешное применение которых

[14] https://en.wikipedia.org/wiki/Artificial_general_intelligence

для распознавания рукописных цифр было осуществлено еще в 1989 году [14]. Сейчас, например, существуют нейросетевые решения-комбайны, которые совмещают в себе все три вышеперечисленные аспекта (классификация, обнаружение и сегментация), например, Mask R-CNN [15]. Возьмем, к примеру, задачу классификации для компьютерного зрения. Обычно классификаторы проверяют на огромной (порядка 1.5 млн изображений) базе данных изображений, названной ImageNet[15] [16], в которой ровно 1000 классов объектов (различные виды животных, предметы интерьера, техника и т.п.). Было проведено исследование[16], в котором уровень ошибки человеческого распознавания (при должной тренировке) был оценен в 5.1%. При этом на данный момент ведущие системы классификации, основанные на сверточных нейросетях, дают ошибку гораздо меньше - 2% и ниже [17]. Предваряя возможные возражения о том, что человеку, даже подготовленному, сложно ориентироваться среди десятков видов собак или кошек, подобное исследование было проведено в области сравнения распознавания такого, казалось бы, близкого любому человеку объекта, как лицо, которое люди начинают распознавать еще до года, и уже с рождения младенец способен распознавать по крайней мере 3 различных выражения лица [18]. Для тестирования использовалась известная база данных лиц Labeled Faces in the Wild (http://vis-www.cs.umass.edu/lfw/) [19], содержащая порядка 13 тысяч лиц от примерно 6 тысяч разных людей. Так вот, было установлено, что ошибка распознавания лиц человеком составляет 2.47% [20], в то время как современные системы распознавания, основанные на сверточных нейросетях, на этой базе допускают ошибку не больше 0.17% [21]. Также можно отметить, что и в языковых задачах (таких как, например, перевод с одного естественного языка на другой) существуют замечательные современные решения [22], которые способны решать на должном уровне большое количество проблем, связанных с семантической обработкой естественного языка.

## Список литературы


[1] *Ashby W. R.* An Introduction to Cybernetics[17] // London, UK: Chapman and Hall, 1956, p. 271.

[2] *Licklider J. C. R.* Man-Computer Symbiosis[18] // IRE Transactions on Human Factors in Electronics, vol. HFE-1, 4-11, 1960, p. 4.


[15] `http://www.image-net.org`

[16] `http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet`

[17] `http://pespmc1.vub.ac.be/books/IntroCyb.pdf`

[18] `http://groups.csail.mit.edu/medg/people/psz/Licklider.html`

[3] *IBM* Cognitive computing. Preparing for the Future of Artificial Intelligence[19], 2018.

[4] *McKinsey Global Institute* Disruptive technologies: Advances that will transform life, business, and the global economy[20], 2013, p. 41.

[5] *Рыжов А.П.* Гибридный интеллект. Сценарии использования в бизнесе[21] // Новосибирск, Академиздат, 2019, 116 с.

[6] *Колмогоров А. Н.* О представлении непрерывных функций нескольких переменных суперпозициями непрерывных функций меньшего числа переменных // Известия АН СССР, 108 (1956), с. 179–182.

[7] *Арнольд В. И.* О функции трех переменных // Известия АН СССР, 114 (1957), с. 679–681.

[8] *Hilbert D.* Mathematical problems // Bulletin of the American Mathematical Society : journal. — 1902. — Vol. 8. — P. 461–462.

[9] *Cybenko G.* Approxtmation by superpositions of a sigmoidal function // Tech. Rep. No.856. Urbana, IL: University of Illinois Urbana-Champaign Department of Electrical and Computer Engineering, 1988.

[10] *Hornik K., Stinchcombe M., White H.* Multilayer feedforward networks are universal approximators // Discussion Paper 88-45. San Diego, CA: Department of Economics, university of California, San Diego, 1988.

[11] *Hodgkin A. L., Huxley A. F.* A quantitative description of membrane current and its application to conduction and excitation in nerve // The Journal of Physiology. — 1952. — vol. 117, no. 4. — pp. 500–544.

[12] *Szegedy C. et al.* Intriguing properties of neural networks // arXiv preprint 1312.6199, 2013.

[13] *Arora S. et al* Stronger generalization bounds for deep nets via a compressionapproach // arXiv preprint 1802.05296v4, 2018.

[14] *LeCun Y., Boser B., Denker J. S., Henderson D., Howard R. E. et al* Backpropagation applied to handwritten zip code recognition // Neural computation, 1989, 1(4), 541–551.

[15] *He K., Gkioxari G., Dollar P., Girshick R.* Mask r-cnn. In Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.

[16] *Deng J., Dong W., Socher R., Li L. J., Li K., Fei-Fei L.* Imagenet: A large-scale hierarchical image database // In 2009 IEEE conference on computer vision and pattern recognition, 2009, pp. 248–255.

[17] *Touvron H., Vedaldi A., Douze M., Jegou H.* Fixing the train-test resolution discrepancy // In Advances in Neural Information Processing Systems, 2019, pp. 8252–8262.

[18] *Веракса Н. Е.* Познавательное развитие в дошкольном детстве // Учебное пособие. — МОЗАИКА-СИНТЕЗ, 2012. — 338 с.

[19] *Huang G. B., Mattar M., Berg T., Learned-Miller E.* Labeled faces in the wild: A database forstudying face recognition in unconstrained environments, 2008.

[20] *Kumar N., Berg A. C., Belhumeur P. N., Nayar S. K.* Attribute and simile classifiers for face verification // In 2009 IEEE 12th international conference on computer vision, 2009, pp. 365–372.

[19] `http://research.ibm.com/cognitive-computing/ostp/rfi-response.shtml`

[20] `http://www.mckinsey.com/insights/business_technology/disruptive_technologies`

[21] `http://itm.ranepa.ru/node/566`

[21] *Deng J., Guo J., Xue N., Zafeiriou S.* Arcface: Additive angular margin loss for deep face recognition // In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 4690–4699.

[22] *Brown T. B. et al.* Language models are few-shot learners / /arXiv preprint 2005.14165, 2020.

## Artificial intelligence: problems and prospects

**Kudryavtsev V.B., Kozlov V.N., Ryjov A.P., Mazurenko I.L., Bokov G.V., Petyushko A.A.**

In this article we present the results of a discussion on the problems and prospects of artificial intelligence, held at the Chair of Mathematical Theory of Intelligent Systems on October 14, 2020. The topic of the discussion goes back to the classic works of Alan Turing "Can machines think?" and John von Neumann "The Computer and the Brain", which emerged at the dawn of Cybernetics as a science. Since then, the discussion on the topic "Can machines think?" appeared and faded, but they did not bring much clarity in the understanding of this issue. In recent years, due to the development of the technological base of computing systems, the topic has become relevant again. Together with the scientific base developed in the theory of artificial intelligence and applications in this area, a lot of speculative works have emerged that are ready to declare any device with trivial algorithmic additions "an artificial intelligence system". This paper attempts to separate the wheat from the chaff by presenting several points of view on this issue.

*Keywords:* artificial intelligence, machine learning, neural networks.

# Часть 2.
# Специальные вопросы теории интеллектуальных систем

# Замечания к определению клеточного автомата с локаторами

Калачев Г.В.[1]

В работе [1] дано определение клеточного автомата с локаторами. В данной работе указаны некоторые неточности и недостатки этого определения и предлагаются уточнения, позволяющие избавиться от этих недостатков. Также приводятся примеры классов клеточных автоматов с локаторами, обладающих в определённом смысле хорошими свойствами.

**Ключевые слова:** клеточные автоматы, однородные структуры.

## 1. Введение

В статье [1] вводится понятие клеточного автомата (КА) с локаторами. Клеточный автомат с локаторами задаётся набором $(\mathbb{Z}^n, Q, V, E, +, L, \varphi, \psi)$. В КА с локаторами по сравнению с обычным КА $(\mathbb{Z}^n, Q, V, \varphi)$ добавляется эфир, в который различные элементарные автоматы могут отправлять сигналы из множества *сигналов вещания* $E$, вычисляемые *функцией вещания* $\psi$, а также принимать сигналы с заданных направлений. При этом отправленные в эфир сигналы суммируются с помощью полугрупповой коммутативной операции $+$, и локаторы принимают сумму сигналов из эфира с направлений, задаваемых телесными углами из множества $L$. КА с локаторами можно рассматривать, как математическую модель устройства, в котором есть как локальные взаимодействия между соседними элементами, так и нелокальные взаимодействия через эфир, который может быть реализован в виде некоторой подложки, суммирующей сигналы от элементов за счёт некоторого физического принципа. Такие устройства могут потенциально решать некоторые задачи более естественным образом, чем обычные клеточные

[1] *Калачев Глеб Вячеславович* — к.ф.-м.н., м.н.с. лаборатории проблем теоретической кибернетики мех.-мат. ф-та МГУ, e-mail: gleb.kalachev@yandex.ru.

Kalachev Gleb Vyacheslavovich — Candidate of Physical and Mathematical Sciences, Junior Researcher, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Problems of Theorecical Cybernetics Lab.

автоматы, где иногда приходится придумывать сложные алгоритмы, в том числе, чтобы передавать управляющие сигналы.

## 2. Определение клеточного автомата с локаторами по Гасанову

Напомним понятие клеточного автомата с локаторами, введённое Э.Э. Гасановым в [1].

Под *телесным углом* в $\mathbb{R}^k$ будем понимать часть пространства $\mathbb{R}^k$, которая является объединением всех лучей, выходящих из данной точки (*вершины угла*) и пересекающих некоторую гиперповерхность в $\mathbb{R}^k$. По определению будем считать, что вершина телесного угла не входит в телесный угол. В частности, в данной работе мы будем рассматривать два вырожденных случая: полный телесный угол, совпадающий с $\mathbb{R}^k$ без вершины угла, который будем обозначать через $\Omega$, и телесные углы, равные одному лучу, такие телесные углы будем обозначать через вектора, являющиеся направляющими лучей.

*Клеточным автоматом с локаторами* называется восьмерка $\sigma = (\mathbb{Z}^k, E_n, V, E_q, +, L, \varphi, \psi)$, где $\mathbb{Z}^k$ — множество $k$-мерных векторов с целыми координатами, $E_n = \{0, 1, \ldots, n-1\}$, $V = (\alpha_1, \ldots, \alpha_{h-1})$ — упорядоченный набор попарно различных ненулевых векторов из $\mathbb{Z}^k$, $E_q = \{0, 1, \ldots, q-1\}$, $+$ — коммутативная полугрупповая операция, заданная на $E_q$, $L = (\nu_1, \ldots, \nu_m)$ — упорядоченный набор попарно различных телесных углов в $\mathbb{R}^k$ с вершиной в начале координат, $\varphi$ — функция, зависящая от переменных $x_0, x_1, \ldots, x_{h-1}, z_1, \ldots, z_m$, $\varphi : E_n^h \times E_q^m \to E_n$, $\varphi(0, \ldots, 0) = 0$, $\psi$ — функция, зависящая от переменных $x_0, x_1, \ldots, x_{h-1}$, $z_1, \ldots, z_m$, $\psi : E_n^h \times E_q^m \to E_q$. Элементы множества $\mathbb{Z}^k$ называются *ячейками* клеточного автомата $\sigma$; элементы множества $E_n$ называются *состояниями ячейки* клеточного автомата $\sigma$; набор $V$ называется *шаблоном соседства* клеточного автомата $\sigma$; элементы множества $E_q$ называются *сигналами вещания*; набор $L$ называется *шаблон локаторов* клеточного автомата $\sigma$; функция $\varphi$ называется *локальной функцией переходов* автомата $\sigma$; функция $\psi$ называется *функцией вещания* автомата $\sigma$; переменные $x_0, x_1, \ldots, x_{h-1}$ принимают значения из $E_n$, переменные $z_1, \ldots, z_m$ принимают значения из $E_q$. Состояние 0 интерпретируется как *состояние покоя*, а условие $\varphi(0, \ldots, 0) = 0$ — как условие сохранения состояния покоя.

Здесь нам нужно было вводить упорядочение шаблона соседства $V$ и шаблон локаторов $L$ для того, чтобы установить взаимно однозначное соответствие между векторами из $V$ и телесными углами из $L$ и переменными локальной функции переходов $\varphi$ и функции веща-

ния $\psi$ соответственно $x_0, x_1, \ldots, x_{h-1}$ и $z_1, \ldots, z_m$. Это соответствие можно сделать более явным, если индексировать переменные функций $\varphi$ и $\psi$ самими векторами и телесными углами, т.е. считать, что локальная функция переходов $\varphi$ и функция вещания $\psi$ зависят от переменных $x_0, x_{\alpha_1}, \ldots, x_{\alpha_{h-1}}, z_{\nu_1}, \ldots, z_{\nu_m}$, здесь индекс первой переменной есть нулевой вектор $0 = (0, \ldots, 0) \in \mathbb{Z}^k$. Если договориться так индексировать переменные локальной функции переходов и функции вещания, то их можно записывать в любом порядке, и тогда можно воспринимать шаблон соседства и шаблон локаторов просто как множество, а не упорядоченный набор.

В дальнейшем мы так и будем поступать: воспринимать шаблон соседства как множество векторов, а шаблон локаторов как множество телесных углов и индексировать переменные локальной функции переходов и функции вещания векторами из шаблона соседства и телесными углами из шаблона локаторов. При этом мы часто будем опускать в индексах внешние круглые скобки у векторов. Например, если $k = 2$, $n = 2$, $q = 2$ и $V = \{(-1, 0), (1, 0)\}$, $L = \{\Omega, (0, 1)\}$, то пример локальной функции переходов может выглядеть так: $\varphi = x_{-1,0} \& z_\Omega \vee x_{1,0} \& z_{0,1}$.

Если $\alpha \in \mathbb{Z}^k$, $\nu$ — телесный угол с вершиной в начале координат, то через $\nu(\alpha)$ обозначим телесный угол, полученный параллельным переносом угла $\nu$ в точку $\alpha$.

Если $\alpha \in \mathbb{Z}^k$ — ячейка клеточного автомата $\sigma$, то множество $V(\alpha) = \{\alpha, \alpha+\alpha_1, \ldots, \alpha+\alpha_{h-1}\}$ называется *окрестностью ячейки* $\alpha$, а множество $L(\alpha) = \{\nu_1(\alpha), \ldots, \nu_m(\alpha_m)\}$ называется *локаторами ячейки* $\alpha$.

*Состоянием клеточного автомата с локаторами* $\sigma$ назовем пару $(e, f)$, где $e$ — произвольная функция, определенная на множестве $\mathbb{Z}^k$, принимающая значения из $E_q$, называемая *состоянием эфира*, $f$ — произвольная функция, определенная на множестве $\mathbb{Z}^k$, принимающая значения из $E_n$ и называемая *распределением состояний клеточного автомата с локаторами* $\sigma$. Такую функцию можно интерпретировать как некую мозаику, возникающую в $k$-мерном пространстве в результате приписывания каждой точке с целочисленными координатами некоторого состояния из множества $E_n$ и некоторого сигнала из множества $E_q$. Множество всевозможных состояний клеточного автомата с локаторами обозначим $\Sigma$.

Если $\alpha \in \mathbb{Z}^k$, $(e, f)$ — состояние клеточного автомата с локаторами $\sigma$, то значение $e(\alpha)$ называем *сигналом ячейки* $\alpha$*, определяемым состоянием* $(e, f)$, а значение $f(\alpha)$ — *состоянием ячейки* $\alpha$*, определяемым состоянием* $(e, f)$. Для каждого $i \in \{1, \ldots, m\}$ значение

$$s_i(\alpha) = \sum_{\beta \in \nu_i(\alpha) \cap \mathbb{Z}^k} e(\beta) \tag{1}$$

называем *значением локатора* $\nu_i$, *определяемым состоянием* $(e, f)$. Здесь суммирование сигналов осуществляется с помощью определяющей операции + полугруппы $E_q$.

На множестве $\Sigma$ определим *глобальную функцию переходов* $\Phi$ клеточного автомата с локаторами $\sigma$, полагая $\Phi(e, f) = (e', f')$, где $(e, f), (e', f') \in \Sigma$ и для любой ячейки $\alpha \in \mathbb{Z}^k$ выполняются тождества

$$f'(\alpha) = \varphi(f(\alpha), f(\alpha + \alpha_1), \ldots, f(\alpha + \alpha_{h-1}), s_1(\alpha), \ldots, s_m(\alpha)), \qquad (2)$$

$$e'(\alpha) = \psi(f(\alpha), f(\alpha + \alpha_1), \ldots, f(\alpha + \alpha_{h-1}), s_1(\alpha), \ldots, s_m(\alpha)). \qquad (3)$$

Содержательная интерпретация отображения $\Phi$ такова, что сигнал каждой ячейки и состояние каждой ячейки "после перехода" определяется по состоянию упорядоченной окрестности ячейки и по значениям локаторов "до перехода" с помощью законов $\psi$ и $\varphi$ одинаково для всех ячеек.

*Поведениями клеточного автомата с локаторами* $\sigma$ называем такие последовательности $(e_0, f_0), (e_1, f_1), (e_2, f_2), \ldots$ его состояний, для которых выполняется $(e_{i+1}, f_{i+1}) = \Phi(e_i, f_i)$ для всех $i = 0, 1, 2, \ldots$, причем $(e_i, f_i)$ называется *состоянием клеточного автомата с локаторами* $\sigma$ *в момент* $i$, а $(e_0, f_0)$ также называется *начальным состоянием клеточного автомата с локаторами* $\sigma$.

Состояние клеточного автомата, у которого лишь конечное число ячеек находится в отличном от 0 состоянии и сигналы всех ячеек равны нулю, назовем *конфигурацией*. Множество конфигураций будем обозначать через $\Sigma'$.

Если задано некоторое состояние клеточного автомата, то ячейки, находящиеся в отличном от 0 состоянии, будем называть *активными*.

# 3. Поправки к определению

## 3.1. Ограничения на телесные углы

Согласно определению в разделе 2, телесный угол — это объединение лучей, пересекающих некоторую гиперповерхность. Однако даже в 2-мерном случае угол задаётся вещественным числом, и это позволяет закодировать в угол бесконечное количество информации. Для 2-мерного случая предлагается ограничить множество телесных углов множеством углов, ограниченным лучами, проходящими через точки с рациональными координатами.

Для многомерного случая здесь ещё больше свобода выбора для телесного угла. Здесь можно также наложить ограничение, что граница телесного угла должна состоять из гиперплоскостей, натянутых на точки

с целочисленными координатами. Заметим, что допускаются вырожденные телесные углы, полностью содержащиеся в подпространстве меньшей размерности. На такие углы также можно наложить ограничения, что их границы в этом подпространстве должны быть частями гиперплоскостей этого подпространства, задаваемых линейными уравнениями с целыми коэффициентами.

### 3.2. Ограничения на полугруппу и функцию вещания

Поскольку в определении самого клеточного автомата требуется наличие нулевого состояния, которое сохраняется функцией перехода, то вполне естественно потребовать то же самое и для алфавита эфира. Формально, в [1] само множество $E$ всегда имеет вид $\{0, .., q-1\}$, и всегда содержит 0, но отсутствует требование, что $0 + x = 0$. Предлагается не требовать, чтобы $E$ всегда имело вид $\{0, ..., q-1\}$, а могло содержать элементы произвольной природы (кроме чисел часто бывает удобно использовать пары или наборы чисел), но потребовать, чтобы полугруппа $(E, +)$ была моноидом, то есть, чтобы был нейтральный элемент $0 \in E$, $0 + x = x$.

В [1] накладывается ограничение на функцию перехода: $\varphi(\mathbf{0}, \mathbf{0}) = 0$. Вполне естественно наложить аналогичное ограничение и на функцию эфира:

$$\psi(\mathbf{0}, \nu) = 0,$$

то есть неактивная ячейка, у которой нет активных соседей, не может посылать сигналы в эфир.

### 3.3. Частичная определённость глобальной функции перехода

В формуле (1) определяется значение локатора $s_i(\alpha)$, которое равно сумме бесконечного количества слагаемых по целочисленным точкам телесного угла, где в качестве сложения используется полугрупповая операция. Бесконечная сумма здесь понимается в обычном смысле (как предел частичных сумм) с уточнением, что на множестве $E$ введена дискретная топология. В данном случае, чтобы рад сходился, нужно, чтобы начиная с некоторого момента частичные суммы были равны константе, которая и является суммой ряда.

Эта сумма может быть не определена, если в сумме участвует бесконечное число ненулевых слагаемых. В общем случае значение локатора будет частично определённой функцией, вследствие чего глобальная функция перехода автомата с локаторами будет также частично определённой. Однако даже здесь требуется обоснование корректности, а именно, что сходимость ряда (1) и значение суммы не зависит от порядка сла-

гаемых (в случае числовых рядов это выполняется только для абсолютно сходящихся рядов).

**Утверждение 1.** *Пусть $(E, +)$ — коммутативная полугруппа с дискретной топологией. Пусть $\{x_j\}_{j=1}^{\infty}$ — последовательность элементов $E$, $\{y_j\}_{j=1}^{\infty}$ — её перестановка ($y_j = x_{i_j}$). Тогда если один из рядов $\sum_{j=1}^{\infty} x_j$ и $\sum_{j=1}^{\infty} y_j$ сходится, то сходится и второй и их суммы совпадают.*

*Доказательство.* Будем доказывать от противного. Без ограничения общности предположим, что $\sum_{j=1}^{\infty} y_j = a$, а ряд $\sum_{j=1}^{\infty} x_j$ либо расходится, либо его сумма не равна $a$. Это означает, что в последовательности частичных сумм $(X_n)_{n=1}^{\infty}$, $X_n = \sum_{j=1}^{n} x_j$ бесконечное число раз встречаются элементы отличные от $a$. Поскольку первый ряд сходится, то существует $N_0$ такое, что для всех $n \geq N_0$ все частичные суммы $Y_n = \sum_{j=1}^{n} y_j$ равны $a$. Это означает, что $a + y_j = a$ для всех $j > N_0$.

Обозначим $K_n = \{j \mid i_j \leq n\}$. Возьмём такое $N \geq \max_{j \leq N_0} i_j$, что $X_N = b \neq a$. По построению $1, 2, ..., N_0 \in K_N$. Значит

$$b = X_N = \sum_{j=1}^{N} x_j = \sum_{k \in K_N} y_k = \sum_{k=1}^{N_0} y_j + \sum_{j > N_0, j \in K_N} y_j = a + \sum_{j > N_0, j \in K_n} y_j = a.$$

Но по предположению $b \neq a$ — противоречие. Значит $\sum_{j=1}^{\infty} x_n = a$, что и требовалось. □

Отметим, что с учётом предыдущих поправок для конфигураций (состояний, где лишь конечное количество активных ячеек), глобальная функция определена, поскольку лишь клетки в отличном от 0 состоянии могут посылать в эфир сигналы. Однако рассмотрим такой КА с локаторами:

$$\sigma = (\mathbb{Z}, \{0, 1\}, \varnothing, \{0, 1\}, \{\Omega\}, \max, \max),$$

где $\Omega$ соответствует локатору, принимающему сигналы со всех направлений.

Пусть вначале ровно одна ячейка находится в состоянии 1, и таким образом состояние является конфигурацией. Тогда в эфире будет сигнал $\max(0, 1) = 1$, и все ячейки в следующий такт получат сигнал 1 из эфира, и перейдут в состояние 1, в результате чего полученное состояние уже не будет конфигурацией. Если же в качестве полугрупповой операции взять $\oplus$ вместо max, то функционирование в первый такт будет таким же, а во второй такт уже функция перехода будет не определена.

# 4. Классы КА с локаторами, представляющие интерес

## 4.1. Классы, решающие проблему частичной определённости функции перехода

Учитывая пример из раздела 3.3, важно выделить классы КА с локаторами $(\mathbb{Z}, Q, V, E, +, L, \varphi, \psi)$, когда гарантируется определённость глобальной функции перехода в любой момент времени для некоторого класса начальных условий.

### 4.1.1. Идемпотентный моноид

Рассмотрим случай, когда моноид $(E, +)$ идемпотентный (является полурешёткой), то есть для любого $x \in E$ выполнено $x + x = x$. В этом случае сумма бесконечного числа элементов зависит лишь от множества слагаемых, присутствующих в сумме, и таким образом сводится к конечной сумме. Поэтому выполнено следующее утверждение.

**Утверждение 2.** *Если моноид $(E, +)$ идемпотентный, то глобальная функция перехода КА с локаторами $\sigma = (\mathbb{Z}, Q, V, E, +, L, \varphi, \psi)$ всюду определена.*

Например, если $E$ — линейно упорядоченное множество, то $(E, \max)$ — идемпотентный моноид с нейтральным элементом $\min E$.

### 4.1.2. Финитные КА с локаторами

Если любая конфигурации $S$ КА с локаторами $\sigma$ переводится глобальной функцией переходов $\Phi$ в конфигурацию, будем автомат $\sigma$ называть *финитным.*

**Утверждение 3** (Достаточное условие финитности КА с локаторами)**.** *Пусть для КА с локаторами $\sigma = (\mathbb{Z}^n, Q, V, E, +, \{\nu_1, ..., \nu_m\}, \varphi, \psi)$ выполнено:*

$$\text{если } \varphi(\vec{\mathbf{0}}, (e_1, ..., e_m)) \neq 0, \text{ то } \bigcap_{i: e_i \neq 0} \nu_i = \varnothing.$$

*Тогда $\sigma$ — финитный.*

В формулировке этого утверждения важно, что мы исключаем из телесного угла его вершину, иначе пересечение телесных углов $\nu_i$ всегда содержало бы начало координат.

*Доказательство.* Рассмотрим произвольную конфигурацию $s$, $A$ — множество активных ячеек, и ячеек, у которых есть активные соседи, $r$ — максимальное евклидово расстояние между элементами $A$.

Допустим, что $\Phi(s)$ не является конфигурацией. В этом случае есть бесконечное множество $M$ ячеек, у которых в конфигурации $s$ не было активных соседей, и которые стали активными в конфигурации $\Phi(s)$. Для каждой ячейки $x$ из $M$ рассмотрим множество её активных локаторов $a(x)$ и выберем такое множество локаторов $L' \subset L$, которое встречается бесконечное число раз среди $a(x)$ при $x \in M$. Пусть $M' = \{x \in M : a(x) = L'\}$.

Без ограничения общности будем считать, что $L' = \nu_1, ..., \nu_k$. Из условия утверждения следует, что $\bigcup_{j=1}^{k} \nu_j = \varnothing$.

Пусть $S$ — единичная сфера в $\mathbb{R}^n$, $P = \prod_{j=1}^{k} (\nu_{i_j} \cap S)$. Покажем, что

$$\hat{d} := \inf_{p \in P} \max_{1 \le j,j' \le k} \|p_j - p_{j'}\| > 0, \tag{4}$$

где $\|\cdot\|$ — евклидова норма.

Заметим, что каждое множество $\nu_{i_j} \cap S$ компактно, поэтому компактно и их произведение $P$, поэтому на нём непрерывная функция $d(p) := \max_{j \neq j'} \|p_j - p_{j'}\|$ достигает своего минимума. Допустим, что этот минимум равен 0. Тогда существует $p \in P$, $p_j \in \nu_j$ и $p_j = p_{j'}$ для всех $1 \le j, j' \le k$, то есть $p_1 = .... = p_k \in \bigcap_{j=1}^{k} \nu_j = \varnothing$ — противоречие. Значит (4) выполнено.

Поскольку множество $M'$ бесконечно, то найдётся элемент $x \in M'$, находящийся на расстоянии $D > r/d$ от множества $A$. Поскольку у ячейки $x$ локаторы $\nu_1, ..., \nu_k$ активны, то существуют элементы $y_1, ..., y_k \in A$ такие, что $v_j = y_j - x \in \nu_j$. Положим $p_j = \frac{v_j}{\|v_j\|}$. Тогда для любых $1 \le i, j \le k$ выполнено:

$$\begin{aligned}
\|p_i - p_j\|^2 = \|p_i\|^2 + \|p_j\|^2 - 2(p_i, p_j) = 2 - 2\frac{(v_i, v_j)}{\|v_i\|\|v_j\|} \le \\
\le \frac{\|v_i\|}{\|v_j\|} + \frac{\|v_j\|}{\|v_i\|} - 2\frac{(v_i, v_j)}{\|v_i\|\|v_j\|} = \frac{\|v_i - v_j\|^2}{\|v_i\|\|v_j\|} \le \\
\le \frac{\|v_i - v_j\|^2}{D^2} = \frac{\|y_i - y_j\|^2}{D^2} \le \frac{r^2}{D^2} < d^2.
\end{aligned}$$

Таким образом, $\max_{1 \le i,j \le k} \|p_i - p_j\| < d$. С другой стороны, $p_j \in \nu_j \cap S$, то есть $p = (p_1, ..., p_k) \in P$ — противоречие с (4). Значит предположение неверно, и $\Phi(s)$ является конфигурацией. Утверждение доказано. □

### 4.2. Класс с простой физической реализацией

Наиболее естественно представлять реализацию КА с локаторами в виде чипа. В качестве эфира должна быть некоторое устройство, «суммиру-

ющее» неограниченное количество электрических сигналов. В качестве такого устройства может выступать проводник, подключённый ко всем выходам элементов, которые нужно суммировать, и подключённый к усилителю, выход которого подключён ко входам-локаторам всех элементов. Таким образом, если один из элементов послал в эфир сигнал, этот сигнал усилится и на локаторы всех элементов придёт сигнал 1. Если же все элементы выдали 0, то и из эфира придёт 0. Таким образом можно реализовать операцию max от неограниченного числа аргументов, принимающих значения из множества $\{0, 1\}$.

Однако для КА с локаторами требуется уметь вычислять $\max_{j \neq i} a_j$ для всех $i = 1, ..., m$. Можно заметить, что

$$\max_{j \neq i} a_j = \min\Big(\sum_{j \neq i} a_j, 1\Big) = \min\Big(\min\Big(\sum_{j=1}^{m} a_j, 2\Big) - a_i, 1\Big).$$

Операцию $M_2(a_1, ..., a_n) = \min\big(\sum_{j=1}^{m} a_j, 2\big)$ также возможно реализовать, но сложнее, чем операцию max. Например, это можно сделать следующим образом. Каждый вход, представляющий аргумент операции, равный 1, выдаёт ограниченный ток на провод, соединяющий все аргументы, и подключённый к нулевому проводу через резистор. В зависимости от числа входов, равных 1, на соединяющем проводнике будет различное напряжение. Сам проводник можно подключить к двум компараторам, из которых один срабатывает при напряжении, когда хотя бы один вход активен, а второй срабатывает при напряжении, когда хотя бы 2 входа активны. Используя результаты сравнения с компараторов, легко получить значение функции $M_2$. Затем по общему проводу можно подвести результат $s = M_2(a_1, ..., a_m)$ обратно ко всем ячейкам, и $i$-й ячейке вычислить $\min(s - a_i, 1)$, получим таким образом в $i$-й ячейке требуемый результат $\max_{j \neq i} a_j$.

Используя $n$ таких схем можно реализовать операцию Max на множестве $\{0, 1\}^n$, которая представляет собой покомпонентную операцию max:

$$\mathrm{Max}((a_1^1, ..., a_n^1), \cdots, (a_1^m, ..., a_n^m)) = (\max(a_1^1, ..., a_1^m), \cdots, \max(a_n^1, ..., a_n^m)).$$

Используя такую операцию Max и обычные функциональные элементы реализуем произвольный идемпотентный коммутативный моноид $(E, +)$, где $|E| = n < \infty$. Для этого закодируем ненулевые элементы $E$ наборами $(1, 0, ..., 0)$, $(0, 1, 0, ..., 0)$, ..., $(0, ..., 0, 1) \in \{0, 1\}^{n-1}$, а $0 \in E$ закодируем набором из всех нулей. Пусть $v$ — описанная функция кодировки. Для множества $E' = \{e_1, ..., e_m\} \subseteq E$ определим $\hat{v}(E') = \mathrm{Max}_{e \in E'} v(e)$. В наборе $\hat{v}(E')$ единицы стоят на позициях, соответствующих ненулевым элементам множества $E'$. Реализуем булев оператор $F : \hat{v}(E') \mapsto v(\sum_{e \in E'} e)$ обычной СФЭ. Используя идемпотентность

моноида, для произвольного числа аргументов получим

$$v\Big(\sum_{i\in I} e_i\Big) = v\Big(\sum_{e\in\{e_i|i\in I\}} e\Big) = F(\hat{v}(\{e_i \mid i \in I\})) = F(\operatorname*{Max}_{i\in I} v(e_i)).$$

Итак, мы получили, что для любого конечного идемпотентного моноида можно реализовать его полугрупповую операцию от неограниченного числа элементов, используя фиксированную СФЭ и несколько проводников, подключённых ко всем элементам, выходы которых суммируются.

Это как раз класс моноидов из пункта 4.1.1, для которых глобальная функция переходов всюду определена. С реализацией локаторов всё обстоит хуже. Проводник проводит одинаково во все стороны. Если же использовать диоды, пропускающие ток только в одну сторону, глубина схемы тут же станет линейной по числу аргументов, и здесь уже нельзя говорить, что сигнал по эфиру распространяется мгновенно, таким образом теряется смысл использования данной модели. Поэтому описанным способом можно реализовать лишь телесные углы, совпадающие с подпространствами. Например, $\Omega$ реализуется, если соединить пластиной выходы всех ячеек. Можно сделать слой с множеством проводов, идущих в одном направлении, и таким образом будет реализовываться локатор $\{v, -v\}$, где $v$ — направление проводов в данном слое.

Для реализации других локаторов требуется использовать какие-то другие физические принципы, выходящие за рамки обычной схемотехники.

## Список литературы

[1] Гасанов Э.Э., “Клеточные автоматы с локаторами”, *Интеллектуальные системы. Теория и приложения*, **22**:2 (2020), 119–132.

### Remarks on the definition of cellular automaton with locators

**Kalachev G.V.**

In [1], a cellular automaton with locators is defined. In this paper we indicate some inaccuracies and issues of this definition and clarify it to get rid of these issues. We also give examples of cellular automata classes with locators that have good properties in a certain sense.

**Keywords:** cellular automata, homogeneous structures.

# Классы кусочно-параллельных функций, содержащие все одноместные

А. Отрощенко[1]

Для класса кусочно-параллельных функций, реализуемых схемами из линейных элементов и функций Хэвисайда, получен алгоритм проверки полноты конечных подмножеств, дополненных одноместными функциями. Таким образом, для рассматриваемого класса решена задача Слупецкого.

**Ключевые слова:** Кусочно-линейная функция, кусочно-параллельная функция, проблема полноты, критерий Слупецкого.

## 1. Определение кусочно параллельной функции

В соответствии с [2], мы рассматриваем класс $PP$ кусочно-параллельных функций, которые строятся из линейных функций $a_1x_1 + ... + a_nx_n + a_0 : R^n \to R, a_i \in R, i = 0, 1, ..., n, n \in 0, 1, ...$ и функции Хэвисайда $\theta(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases}$ с использованием операций суперпозиции. Как показано в [2], функция $f$ из $PP$ может быть представлена в следующем виде:$f = f_L + f_{PC}$, где $f_L$ -линейная функция, а $f_{PC}$ - кусочно-постоянная функция. Будкм обозначать $\vec{a} = (a_1, a_2, \ldots, a_n)$, $\vec{b} = (b_1, b_2, \ldots, b_n)$, $\langle \vec{a}, \vec{b} \rangle = \sum_{i=1}^{n} a_i b_i$.
В соответствии с [1], кусочно-параллельная функция имеет вид

$$f(\vec{x}) = \langle \vec{a}_0, \vec{x} \rangle + \sum_{i=1}^{s} d_i \theta(\sum_{j=1}^{k} \chi(sgn(\langle \vec{a}_j, \vec{x} \rangle + c_j) = \sigma_{ij}) - k). \qquad (1)$$

[1] *Отрощенко Александр Дмитриевич* — аспирант каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: iskander.aka@mail.ru.

Otroschenko Alexander Dmitrievich — graduate student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

где $\sigma_{ij} \in \{-1, 0, 1\}$, $\chi(A) = \begin{cases} 1, \text{условие } A \text{ выполнено} \\ 0, \text{условие } A \text{ не выполнено} \end{cases}$.
В дальнейшем, вместо $\sum_{i=1}^{s} d_i \theta(\sum_{j=1}^{k} \chi(sgn(\langle \vec{a}_j, \vec{x} \rangle + c_j) = \sigma_{ij}) - k)$ мы будем писать $NL_f(\vec{x})$.
Будем называть множество значений аргумента соответствующую определенному $d_i$ в дальнейшем носителем сигнатуры i, а сам $d_i$ - сдвигом. Носитель сигнатуры, неограниченный хотя бы с одной стороны по каждой из координат, будем называть нограниченным. Плоскости разделяющие носители сигнатуры будем называть разрезами. Мы будем рассматривать дальше кусочно-параллельные функции с конечным числом сдвигов. Обозначим $U$ множество кусочно-параллельных функций, замыкание которого содержит все одноместные.

# 2. Предполнота и замкнутость класса функций с линейной частью, зависящей от не более чем одной переменной

Обозначим множество функций с линейной частью, зависящей от не более чем одной переменной $NLL_1$.

**Теорема 1.** *$NLL_1$ - предполный класс в $PP$.*

*Доказательство.* Замкнутость его очевидна. Пусть есть $f \notin NLL_1$.
$f = \langle \vec{f}_l, \vec{x} \rangle + NL_f(\vec{x})$ и в $\vec{f}$ больше одной ненулевой компоненты. Возьмем $g(y, \vec{z}) = y - NL_f(\vec{z})$, $g \in NLL_1$. Тогда

$$g(f(\vec{x}), \vec{x}) = \langle \vec{f}_l, \vec{x} \rangle + NL_f(\vec{x}) - NL_f(\vec{x}) = \langle \vec{f}_l, \vec{x} \rangle$$

где у $\vec{f}_l$, больше одной ненулевой компоненты. Очевидно, что можно с помощью одноместных функций далее получить сумматор, а затем и все кусочно-параллельные функции. □

# 3. Критерий Слупецкого для пространства кусочно-параллельных функций

## 3.1. Формулировка критерия и общий план доказательства

**Теорема 2.** *Замыкание $U$ совпадает с классом кусочно-параллельных функций тогда, и только тогда, когда $U \not\subseteq NLL_1$*

Наша цель - получение сумматора, т.к. при добавлении его к одноместным, мы получим базис пространства кусочно-параллельных функций. Для доказательства мы сначала получим функцию $x + \theta(y)$. Далее, мы дадим определение угловой функции, и получим эту функцию. Затем, с помощью угловой функции и функции $x + \theta(y)$ мы покажем получение функций $F(x, y) = \theta(ax + by + c)$, после чего избавимся от нелинейной части у двухместной функции.

### 3.2. Проводник с независимой ступенью

Назовем функцию $x + \theta(y)$ проводником с независимой ступенью.

**Теорема 3.** *Пусть $U \not\subseteq NLL_1$. Тогда $g \in [U]$, где $g(x, y) = x + \theta(y)$*

*Доказательство.* Пусть $f \in U/NLL_1$.
Значит, в $f(\vec{x}) = \langle \vec{a}_0, \vec{x} \rangle + \sum_{i=1}^{s} d_i \theta(\sum_{j=1}^{k} \chi(sgn(\langle \vec{a}_j, \vec{x} \rangle + c_j) = \sigma_{ij}) - k)$ линейная часть зависит от не менее чем двух переменных, поэтому у $\vec{a}_0$ есть две ненулевые компоненты. Пусть эти компоненты по первой и второй переменной. Для простоты, подставим во все остальные переменные ноль, в $x_1 = x/a_{01}$, $x_2 = y/a_{02}$ и получим $f_{01}(x, y)$. Итак

$$f_{01}(x, y) = x + y + \sum_{i=1}^{s} d_i \theta(\sum_{j=1}^{k} \chi(sgn(\frac{a_{j1}}{a_{01}} x + \frac{a_{j2}}{a_{02}} y + c_j) = \sigma_{ij}) - k) =$$

$$= x + y + \sum_{i=1}^{s} d_i \theta(\sum_{j=1}^{k} \chi(sgn(A_j x + B_j y + c_j) = \sigma_{ij}) - k).$$

Далее определим константы $C_1 > 0$, $C_2 > 0$ и $0 < \epsilon \leq 1$. Рассмотрим $f_{02}(x, y) = f_{01}(x + 2C_1\theta(x) - C_1, \epsilon\theta(y) + C_2)$.

$$f_{02}(x, y) = x + 2C_1\theta(x) - C_1 + \epsilon\theta(y) + C_2 +$$

$$+ \sum_{i=1}^{s} d_i \theta(\sum_{j=1}^{k} \chi(sgn(A_j(x + 2C_1\theta(x) - C_1) + B_j(\epsilon\theta(y) + C_2) + c_j) = \sigma_{ij}) - k) =$$

$$= x + \epsilon\theta(y) + 2C_1\theta(x) +$$

$$+ \sum_{i=1}^{s} d_i \theta(\sum_{j=1}^{k} \chi(sgn(A_j(x + 2C_1\theta(x) - C_1) + \epsilon B_j\theta(y) + B_j C_2 + c_j) = \sigma_{ij}) - k) +$$

$$+C_2 - C_1.$$

Теперь подберем $C_1$ и $C_2$, так чтобы $\forall j, \epsilon$, выражение

$$W(j,x,y) = sgn(A_j(x + 2C_1\theta(x) - C_1) + \epsilon B_j\theta(y) + B_jC_2 + c_j)$$

не зависело от $y$. Сделаем это так. Если $\forall j$, $B_j = 0$, то $W(j,x,y)$ не зависит от $y$. Тогда положим $C_2 = 0$. В противном случае, положим $C_2 = \epsilon + \max\limits_{j:B_j\neq 0} \frac{|c_j|}{|B_j|} + 1$.

Пусть $V = \max\limits_j(|\epsilon B_j| + |B_jC_2| + |c_j|)$. Положим $C_1 = \frac{V+1}{\min\limits_{j:A_j\neq 0}|A_j|}$.

Теперь заметим, что если для какого-то $j$, $A_j = 0$, то или

$$\begin{cases} B_j(C_2 + \epsilon) + c_j > 0 \\ B_jC_2 + c_j > 0, \end{cases}$$

или

$$\begin{cases} B_j(C_2 + \epsilon) + c_j < 0 \\ B_jC_2 + c_j < 0, \end{cases}$$

т.е. в этом случае $W(j,x,y)$ не зависит от $y$. Пусть в этом случае $W(j,x,y) = G_j$.

Если $A_j \neq 0$, то

$$W(j,x,y) = sgn(A_j(x + 2C_1\theta(x) - C_1) + \epsilon B_j\theta(y) + B_jC_2 + c_j) =$$

$$= sgn(A_j(x + \frac{2(V+1)}{\min\limits_{i:A_i\neq 0}|A_i|}\theta(x) - \frac{V+1}{\min\limits_{i:A_i\neq 0}|A_i|} + \frac{\epsilon B_j\theta(y) + B_jC_2 + c_j}{A_j})) =$$

$$= sgnA_j sgn(x + \frac{2(V+1)}{\min\limits_{i:A_i\neq 0}|A_i|}\theta(x) - \frac{V+1}{\min\limits_{i:A_i\neq 0}|A_i|} + \frac{\epsilon B_j\theta(y) + B_jC_2 + c_j}{A_j})$$

Теперь заметим, что $\frac{V+1}{\min\limits_{i:A_i\neq 0}|A_i|} > |\frac{\epsilon B_j\theta(y) + B_jC_2 + c_j}{A_j}|$, а значит, при $x \geq 0$,

$$x + \frac{2(V+1)}{\min\limits_{i:A_i\neq 0}|A_i|}\theta(x) - \frac{V+1}{\min\limits_{i:A_i\neq 0}|A_i|} + \frac{\epsilon B_j\theta(y) + B_jC_2 + c_j}{A_j} \geq$$

$$\geq \frac{V+1}{\min\limits_{i:A_i\neq 0}|A_i|} - |\frac{\epsilon B_j\theta(y) + B_jC_2 + c_j}{A_j}| > 0,$$

а при $x < 0$,

$$x + \frac{2(V+1)}{\min\limits_{i:A_i \neq 0} |A_i|}\theta(x) - \frac{V+1}{\min\limits_{i:A_i \neq 0} |A_i|} + \frac{\epsilon B_j \theta(y) + B_j C_2 + c_j}{A_j} <$$

$$< -\frac{V+1}{\min\limits_{i:A_i \neq 0} |A_i|} + |\frac{\epsilon B_j \theta(y) + B_j C_2 + c_j}{A_j}| < 0,$$

а значит,

$$W(j,x,y) =$$

$$= sgnA_j sgn(x + \frac{2(V+1)}{\min\limits_{i:A_i \neq 0} |A_i|}\theta(x) - \frac{V+1}{\min\limits_{i:A_i \neq 0} |A_i|} + \frac{\epsilon B_j \theta(y) + B_j C_2 + c_j}{A_j}) =$$

$$= (2\theta(x) - 1)sgnA_j,$$

и при нашем выборе $C_1, C_2$, не зависит от $y$. Осуществим эту подстановку, пока не определяя $\epsilon$.

$$f_{02}(x,y) = x + \epsilon\theta(y) + 2C_1\theta(x) +$$

$$+\sum_{i=1}^{k} d_i \theta(\sum_{j=1}^{s} \chi(sgn(A_j(x+2C_1\theta(x)-C_1)+\epsilon B_j\theta(y)+B_jC_2+c_j) = \sigma_{ij}) - k) +$$

$$+C_2 - C_1 = x + \epsilon\theta(y) + 2C_1\theta(x) +$$

$$+\sum_{i=1}^{k} d_i \theta(\sum_{j=1,A_j\neq 0}^{s} \chi(sgnA_j(2\theta(x)-1) = \sigma_{ij}) + \sum_{i=1}^{k} d_i \theta(\sum_{j=1,A_j=0}^{s} \chi(G_j = \sigma_{ij}) - k) +$$

$$+C_2 - C_1$$

При $x \geq 0$ имеем,

$$\sum_{i=1}^{k} d_i \theta(\sum_{j=1,A_j\neq 0}^{s} \chi(sgnA_j(2\theta(x)-1) = \sigma_{ij}) + \sum_{i=1}^{k} d_i \theta(\sum_{j=1,A_j=0}^{s} \chi(G_j = \sigma_{ij}) - k) =$$

$$= \sum_{i=1}^{k} d_i \theta( \sum_{j=1, A_j \neq 0}^{s} \chi(sgn A_j = \sigma_{ij}) + \sum_{i=1}^{k} d_i \theta( \sum_{j=1, A_j = 0}^{s} \chi(G_j = \sigma_{ij}) - k) = T_{pos},$$

а при $x < 0$ имеем,

$$\sum_{i=1}^{k} d_i \theta( \sum_{j=1, A_j \neq 0}^{s} \chi(sgn A_j (2\theta(x) - 1) = \sigma_{ij}) + \sum_{i=1}^{k} d_i \theta( \sum_{j=1, A_j = 0}^{s} \chi(G_j = \sigma_{ij}) - k) =$$

$$= \sum_{i=1}^{k} d_i \theta( \sum_{j=1, A_j \neq 0}^{s} \chi(-sgn A_j = \sigma_{ij}) + \sum_{i=1}^{k} d_i \theta( \sum_{j=1, A_j = 0}^{s} \chi(G_j = \sigma_{ij}) - k) = T_{neg}.$$

Тогда

$$f_{02}(x, y) = x + \epsilon\theta(y) + 2C_1\theta(x) + (T_{pos} + T_{neg})\theta(x) - T_{neg} + C_2 - C_1,$$

то есть

$$f_{02}(x, y) = x + \epsilon\theta(y) + A\theta(x) - B$$

Если $A \leq 0$, то рассмотрим

$$f_{03}(x, y) = f_{02}(x - A\theta(x), y) + B = x - A\theta(x) + \epsilon\theta(y) + A\theta(x - A\theta(x)) - B + B =$$

$$= x - A\theta(x) + \epsilon\theta(y) + A\theta(x) = x + \epsilon\theta(y),$$

а затем положим $\epsilon = 1$ и получим искомую функцию $f_{03}(x, y) = x + \theta(y)$.
Если $A > 0$, то подставим $f_{02}(x, y) + B - A/2$ в $q(z) = z - A\theta(z) + A/2$:

$$f_{13}(x, y) = q(f_{02}(x, y) + B) =$$

$$= x + A\theta(x) + \epsilon\theta(y) - A\theta(x + A\theta(x) + \epsilon\theta(y) - A/2) - B + B - A/2 + A/2 =$$

$$= x + A\theta(x) + \epsilon\theta(y) - A\theta(x + A\theta(x) + \epsilon\theta(y) - A/2).$$

Теперь выберем $\epsilon = A/4$. Тогда, если $x \geq 0$, то

$$x + A\theta(x) + A\theta(y)/4 - A/2 \geq A - A/2 = A/2 > 0,$$

а если $x < 0$, то

$$x + A\theta(x) + A\theta(y)/4 - A/2 < A/4 - A/2 = -A/2 < 0,$$

то есть $sgn(x + A\theta(x) + \epsilon\theta(y) - A/2) = 2\theta(x) - 1$. Теперь

$$f_{13}(x,y) = x + A\theta(x) + A\theta(y)/4 - A\theta(x + A\theta(x) + A\theta(y)/4 - A/2) =$$

$$= x + A\theta(x) + A\theta(y)/4 - A\theta(sgn(x + A\theta(x) + A\theta(y)/4 - A/2)) =$$

$$= x + A\theta(x) + A\theta(y)/4 - A\theta(2\theta(x) - 1) =$$
$$= x + A\theta(x) + A\theta(y)/4 - A\theta(x) = x + A\theta(y)/4.$$

Значит, $4f_{13}(Ax/4, y)/A = x + \theta(y)$ - искомая. Теорема доказана. □

В условиях теоремы 3 $x + \sum_{i=1}^{n} d_i\theta(y_i) \in [U]$

*Доказательство.* Ясно, что $g_b(x,y) = x + b\theta(y) = b((\frac{1}{b}x) + \theta(y))$. Тогда функция

$$g_{d_n}(g_{d_{n-1}}(g_{d_{n-2}}(\ldots g_{d_2}(g_{d_1}(x,y_1),y_2),\ldots),y_{n-2}),y_{n-1}),y_n) =$$

$$= x + \sum_{i=1}^{n} d_i\theta(y_i)$$

также принадлежит $[U]$. □

### 3.3. Функция 'уголок'

Назовем 'уголком' функцию $\theta(\theta(y-x) + \theta(x) - 2)$. Имеет место:

**Теорема 4.** *Пусть $U \not\subset NLL_1$. Тогда замыкание $U$ содержит $\theta(\theta(y - x) + \theta(x) - 2)$.*

*Доказательство.* В доказательстве теоремы 2.1 было показано, что при условии теоремы, замыкание содержит функцию

$$f(x,y) = x + y + \sum_{i=1}^{s} d_i \theta(\sum_{j=1}^{k} \chi(sgn(A_j x + B_j y + c_j) = \sigma_{ij}) - k).$$

Рассмотрим теперь функцию

$$h(x,y) = f(-x,y) = y - x + \sum_{i=1}^{s} d_i \theta(\sum_{j=1}^{k} \chi(sgn(-A_j x + B_j y + c_j) = \sigma_{ij}) - k).$$

Множество состоящее из всех точек пересечения множества прямых $\{-A_j x + B_j y + c_j = 0\}_{j=1}^{k}$, пусть это будет множество точек $\{x_i, y_i\}_{i=1}^{u}$ Добавим туда и множество точек пересечения этих прямых с осями x и y.
Рассмотрим

$$h_1(x,y) = h(x + C_x^1, y + C_y^1),$$

где $C_x^1 = \max\limits_i |x_i| + 1$, $C_y^1 = \max\limits_i |y_i| + 1$. Вершины областей тоже сдвинутся, и перейдут в $\{x_i - C_x^1, y_i - C_y^1\}_{i=1}^{u}$. Заметим, что каждая координата этих точек меньше нуля. Теперь рассмотрим те, $-A_j x + B_j y + c_j = 0$, у которых $A_j = 1$, $B_j = 1$. Пусть $C_{par} = \max\limits_{A_j=1, B_j=1} |c_j| + 1$, если же $\{A_j = 1, B_j = 1\} = \emptyset$, то положим $C_{par} = 1$.
Рассмотрим

$$H(x,y) = h_1(x, y + C_{par}) - V.$$

Заметим, что точки пересечений разрезов по-прежнему лежат левее оси $y$ и ниже оси $x$. Возьмем $V = \max |d_i| + C_{par}$. Выпишем $H(x,y)$.

$$H(x,y) = y + C_{par} - x + \sum_{i=1}^{s} d_i \theta(\sum_{j=1}^{k} \chi(sgn(-A_j x + B_j y + c_j^{'}) = \sigma_{ij}) - k) - V.$$

Заметим, что в области $x \geq 0$, $y \geq 0$, $H(x,y)$ имеет только неограниченные носители сигнатур, т.к. все точки пересечения прямых, на которых лежат границы носителей находятся в области $x < 0$, $y < 0$. Также, при $x > y$, функция

$$H(x,y) = (y-x) + \sum_{i=1}^{k} d_i \theta(\sum_{j=1}^{s} \chi(sgn(-A_j x + B_j y + c_j^{'}) = \sigma_{ij}) - s) + C_{par} - V =$$

$$= (y - x) + \sum_{i=1}^{k} d_i \theta(\sum_{j=1}^{s} \chi(sgn(-A_j x + B_j y + c_j^{'}) = \sigma_{ij}) - s) - \max |d_i| <$$

$$< y - x < 0.$$

Теперь рассмотрим следующую систему неравенств

$$\begin{cases} H(x,y) \geq 0 \\ x \geq 0 \\ y \geq 0. \end{cases}$$

Рассмотрим разрезы $-A_j x + B_j y + c_j^{'} = 0$. Все прямые с $A_j = 0$, или $B_j = 0$ лежат или ниже, или правее области $x \geq 0$,$y \geq 0$ и положение относительно них не будет влиять на решение системы при $x \geq 0$, $y \geq 0$. Значит мы можем считать, что все прямые имеют вид $y - k_j x - r_j = 0$. Отсортируем их по $k_j$, по убыванию, при равных $k_j$ отсорируем по убыванию $r_j$. Теперь заметим, что если для $j > i$, $k_j < k_i$, то и $r_j < r_i$, т.к. у линий с номерами $j$ и $i$ у точки пересечения координата $x = \frac{r_i - r_j}{k_j - k_i}$, а раз $x < 0$ и $k_j - k_i < 0$, то $r_i - r_j > 0$.
Значит разрезы при нашей нумерации располагаются по порядку против часовой стрелки при обходе относительно точки $(0,0)$ в области $x > 0$ и т.к. они не имеют пересечений в $x \geq 0$,$y \geq 0$, то при $x \geq 0$,$y \geq 0$,

$$H(x,y) = \begin{cases} y - x + W_1 : y - k_1 x - c_1 > 0 \\ y - x + W_1^{'} : y - k_1 x - c_1 = 0 \\ y - x + W_2 : y - k_1 x - c_1 < 0, y - k_2 x - c_2 > 0 \\ y - x + W_2^{'} : y - k_2 x - c_2 = 0 \\ \dots \\ y - x + W_i : y - k_{i-1} x - c_{i-1} < 0, y - k_i x - c_i > 0 \\ y - x + W_i^{'} : y - k_i x - c_i = 0 \\ \dots \\ y - x + W_t : y - k_t x - c_t > 0 \\ y - x + W_t^{'} : y - k_t x - c_t = 0. \end{cases} \tag{2}$$

Тогда система 2 эквивалентна следующей совокупности:

$$
\left[
\begin{array}{l}
\begin{cases} y - x + W_1 \geq 0 \\ y - k_1 x - c_1 > 0 \\ x \geq 0 \\ y \geq 0, \end{cases}
\begin{cases} y - x + W_1 > 0 \\ y - k_1 x - c_1 = 0 \\ x \geq 0 \\ y \geq 0, \end{cases} \\
\begin{cases} y - x + W_2 \geq 0 \\ y - k_1 x - c_1 < 0 \\ y - k_2 x - c_2 > 0 \\ x \geq 0 \\ y \geq 0, \end{cases}
\begin{cases} y - x + W_2 \geq 0 \\ y - k_2 x - c_2 = 0 \\ x \geq 0 \\ y \geq 0, \end{cases} \\
\dots \\
\begin{cases} y - x + W_t \geq 0 \\ y - k_{i-1} x - c_{i-1} < 0 \\ y - k_i x - c_i > 0 \\ x \geq 0 \\ y \geq 0, \end{cases}
\begin{cases} y - x + W_l > 0 \\ y - k_i x - c_i = 0 \\ x \geq 0 \\ y \geq 0, \end{cases} \\
\dots \\
\begin{cases} y - x + W_t > 0 \\ y - k_t x - c_t < 0 \\ x \geq 0 \\ y \geq 0. \end{cases}
\end{array}
\right. \tag{3}
$$

Теперь:
1)$H(x, y) < 0$ при $x < y$. Значит можно не учитывать решения в областях, у которых для обеих ограничивающих прямых $k_i \leq 1$.
2)Условия $y - k_{i-1}x - c_{i-1} < 0, y - k_i x - c_t > 0$ определяют константу для $H(x, y)$

Распишем систему 3 далее:

$$\left[\begin{array}{l} \begin{cases} y \geq x - W_1 \\ y > k_1 x + c_1 \\ x \geq 0 \\ y \geq 0, \end{cases} \begin{cases} y \geq x - W_1^{'} \\ y = k_1 x + c_1 \\ x \geq 0 \\ y \geq 0, \end{cases} \\ \\ \begin{cases} y \geq x - W_2 \\ y < k_1 x + c_1 \\ y > k_2 x + c_2 \\ x \geq 0 \\ y \geq 0, \end{cases} \begin{cases} y \geq x - W_2^{'} \\ y = k_2 x + c_2 \\ x \geq 0 \\ y \geq 0, \end{cases} \\ \dots \\ \begin{cases} y \geq x - W_t \\ y < k_{t-1} x + c_{t-1} \\ x \geq 0 \\ y \geq 0, \end{cases} \end{array}\right. \tag{4}$$

где $k_{t-1} = \min\limits_{k_i > 1} k_i$. Рассмотрим все точки пересечения пар прямых $y = x - W_i, y = x - W_i^{'}$ и $y = k_i x + c_i$ (пары берем учавствующие в одних и тех же системах), а также $y = x - W_i$ и $y = k_{i-1} x + c_{i-1}$ (для пар содержащихся в одних и тех же системах). Пусть это точки $\{X_i^{'}, Y_i^{'}\}$. Пусть $G = \max X_i^{'} + 1$. Рассмотрим, как выглядят решения системы при $x \geq G$. Заметим, что для этих $x$ выполнено следующее: $y \geq x - W_t$, тогда и только тогда, когда $H(x, y) \geq 0$.
Это очевидно из геометрических соображений, но покажем это строго. Пусть $(X, Y)$ - решение следующей системы

$$\begin{cases} y \geq x - W_i \\ y < k_{i-1} x + c_{i-1} \\ y > k_i x + c_i \\ x \geq 0 \\ y \geq 0, \end{cases} \tag{5}$$

и $X = \delta + X_{i0} = \frac{-c_i - W_i}{k_i - 1}$, где $X_{i0}, Y_{i0}$, решение системы

$$\begin{cases} y = x - W_i \\ y = k_i x + c_i, \end{cases}$$

а $\delta > 0$. В этом случае, т.к. $Y > k_i X + c_i = k_i(X_{i0} + \delta) + c_i = k_i X_{i0} + \delta k_i + c_i = \delta(k_i - 1) + \delta + X_{i0} - W_i = X - W_i + \delta(k_i - 1)$, а $k_i > 1$, то $Y > X - W_i$. Это значит, что в рассматриваемом случае система 5 равносильна системе

$$\begin{cases} y < k_{i-1}x + c_{i-1} \\ y > k_i x + c_i \\ x \geq X_{i0} \\ y \geq 0. \end{cases}$$

Повторим те же рассуждения для системы

$$\begin{cases} y \geq x - W_j^{'} \\ y = k_j x + c_j \\ x \geq 0 \\ y \geq 0 \end{cases}$$

и получим, что она равносильна системе

$$\begin{cases} y = k_j x + c_2 \\ x \geq X_{j0}^{'} \\ y \geq 0. \end{cases}$$

Мы рассматриваем, совокупность систем 4 при $x \geq G$, и далее получаем из совокупности

$$\left[\begin{array}{l} \begin{cases} y \geq x - W_1 \\ y > k_1 x + c_1 \\ x \geq max(G, 0) \\ y \geq 0 \end{cases} \quad \begin{cases} y \geq x - W_1^{'} \\ y = k_1 x + c_1 \\ x \geq max(G, 0) \\ y \geq 0 \end{cases} \\ \begin{cases} y \geq x - W_2 \\ y < k_1 x + c_1 \\ y > k_2 x + c_2 \\ x \geq max(G, 0) \\ y \geq 0 \end{cases} \quad \begin{cases} y \geq x - W_2^{'} \\ y = k_2 x + c_2 \\ x \geq max(G, 0) \\ y \geq 0 \end{cases} \\ \ldots \\ \begin{cases} y \geq x - W_t \\ y < k_{t-1} x + c_{t-1} \\ x \geq max(G, 0) \\ y \geq 0, \end{cases} \end{array}\right.$$

совокупность

$$\left[\begin{array}{l}
\begin{cases} y > k_1x + c_1 \\ x \geq max(G,0) \\ y \geq 0 \end{cases} \quad \begin{cases} y = k_1x + c_1 \\ x \geq max(G,0) \\ y \geq 0 \end{cases} \\
\begin{cases} y < k_1x + c_1 \\ y > k_2x + c_2 \\ x \geq max(G,0) \\ y \geq 0 \end{cases} \quad \begin{cases} y = k_2x + c_2 \\ x \geq max(G,0) \\ y \geq 0 \end{cases} \\
\qquad \cdots \\
\qquad \begin{cases} y \geq x - W_t \\ y < k_{t-1}x + c_{t-1} \\ x \geq max(G,0) \\ y \geq 0. \end{cases}
\end{array}\right.$$

Теперь заметим, что эта совокупность переходит в следующую

$$\left[\begin{array}{l}
\begin{cases} y \geq k_{t-1}x + c_{t-1} \\ x \geq max(G,0) \\ y \geq 0, \end{cases} \\
\begin{cases} y \geq x - W_t \\ y < k_{t-1}x + c_{t-1} \\ x \geq max(G,0) \\ y \geq 0, \end{cases}
\end{array}\right. \tag{6}$$

и $(X, Y)$ - одно из ее решений. Теперь, пусть $(X_0, Y_0)$ решение системы:

$$\begin{cases} y = x - W_t \\ y = k_{t-1}x + c_{t-1}. \end{cases}$$

Значит, $X_0 = \frac{-c_{t-1}-W_t}{k_{t-1}-1}$. Имеем, $X = X_0+\delta$, $\delta > 0$. Пусть $Y > k_{t-1}X+c_{t-1}$. Тогда

$$Y > k_{t-1}(X_0 + \delta) + c_{t-1} = (k_{t-1} - 1)X_0 + (k_{t-1} - 1)\delta + X_0 + \delta + c_{t-1} =$$

$$= -c_{t-1} - W_t + (k_{t-1} - 1)\delta + X_0 + \delta + c_{t-1} = X - W_t + (k_{t-1} - 1)\delta > X - W_t.$$

Отсюда следует, что совокупность 6, при $x \geq G$, эквивалентна системе

$$\begin{cases} y \geq x - W_t \\ x \geq max(G,0) \\ y \geq 0. \end{cases}$$

Пусть $max(G,0) = C_{maj}$, а $W_t = D$. Тогда системы $\begin{cases} y - x + W_t \geq 0 \\ x \geq C_{maj} \\ y \geq 0 \end{cases}$ и

$\begin{cases} H(x,y) \geq 0 \\ x \geq C_{maj} \\ y \geq 0 \end{cases}$ эквивалентны.

Отсюда следует, что

$$\theta(\theta(H(x,y))+\theta(x-C_{maj})+\theta(y)-3) = \theta(\theta(y-x+D)+\theta(x-C_{maj})+\theta(y)-3).$$

Функцию $\phi(l,m,n) = \theta(\theta(l) + \theta(m) + \theta(n) - 3)$ несложно получить с помощью функции $g(x,y) = x + \theta(y)$ и одноместных, заметив что $\phi(l,m,n) = g(0, g(g(g(-3,n),m),l)$.

Подставим в $x$, $x = x^{`} + D$, и пусть $R = D - C_{maj}$. Тогда $\theta(\theta(H(x^{`} + D, y)) + \theta(x^{'} + R) + \theta(y) - 3) = \theta(\theta(y - x^{'}) + \theta(x^{'} + R) + \theta(y) - 3)$.

Заметим, что, если $R < 0$, то

$$\theta(\theta(y - x^{'}) + \theta(x^{'} + R) + \theta(y) - 3) = \theta(\theta(y - x^{'}) + \theta(x^{'} + R) - 2),$$

и мы получаем нужную функцию подстановкой $y = y^{''} + R$, $x^{'} = x^{''} - R$.

Если $R \geq 0$, то

$$\theta(\theta(\theta(y - x^{'}) + \theta(x^{'} + R) + \theta(y) - 3) + \theta(x) - 2) = \theta(\theta(y - x) + \theta(x) - 2).$$

Теорема доказана. □

Теперь получим следующий результат.

**Теорема 5.** *Пусть $U \not\subset NLL_1$. Тогда $\forall a, b, c \in R$ замыкание $U$ содержит $F_{a,b,c}(x,y) = \theta(ax + by + c)$.*

*Доказательство.* В предыдущих теоремах мы показали, что

$$\{h(x,y) = \theta(\theta(y - x) + \theta(x) - 2), g(x,y) = x + \theta(y)\} \subset [U]$$

. Теперь покажем, что $\theta(y - x) \in [U]$. Рассмотрим следующую функцию:

$$f_{prot}(x,y) = \theta(h(x,y) + h(-y,-x) + \theta(\theta(-x) + \theta(y) - 2) - 0.5).$$

Ясно, что $h(x,y)$ равна $\theta(y-x)$, при $x \geq 0$, $h(-y,-x) = \theta(\theta(y-x)+\theta(-y)-2)$ равна $\theta(y - x)$, при $x < 0$, $y < 0$, а $\theta(\theta(-x) + \theta(y) - 2)$ равна $\theta(y - x)$ при $x < 0$, $y \geq 0$. При этом, если $\theta(y - x) = 0$, то и $h(x,y)$, $h(-y,-x)$, $\theta(\theta(-x)+\theta(y)-2)$ также равны нулю, а при этом объединение множеств, где эти функции единичны, дают множество на котором $\theta(y - x) = 1$. Теперь несложно увидеть, что $f_{prot}(x,y) = \theta(y - x)$. Осталось заметить, что

$$f_{prot}(-ax, by + c) = \theta(ax + by + c) = F_{a,b,c}(x,y).$$

Теорема доказана. □

### 3.4. Завершение доказательства критерия Слупецкого

Итак, в условиях теоремы 2 замыканию принадлежит $x + y + NL(x, y)$. Построим $u - NL(x, y)$. Пусть

$$NL(x,y) = \sum_{i=1}^{k} d_i \theta(\sum_{j=1}^{s} \chi(sgn(A_j x + B_j y + C_j) = \sigma_{ij}) - k).$$

Следуя [1],

$$NL(x,y) = \sum_{i=1}^{k} d_i \theta(\sum_{j=1}^{v_i} q_j \theta(A'_{ij} x + B'_{ij} y + C'_{ij}) - r_i).$$

С помощью функции $x+\theta(y)$ постром $v_i(p, \vec{o}) = p+\sum_{j=1}^{l_i} q_j \theta(o_j - 0.5) - r_i$. Теперь подставим $o_j = F_{A'_{ij}, B'_{ij}, C'_{ij}}(x, y)$, а $p = r_i$. После этого, получим $B(u, \vec{z}) = u - \sum_{i=1}^{k} d_i \theta(z_i)$, и подставим в $z_i = v_i(r_i, (F_{A'_{ij}, B'_{ij}, C'_{ij}}(x, y))_{j=1}^{l_i})$. Получим функцию

$$\mu(u,x,y) = u - \sum_{i=1}^{k} d_i \theta(\sum_{j=1}^{v_i} q_j \theta(\theta(A'_{ij} x + B'_{ij} y + C'_{ij}) - 0.5) - r_i) =$$

$$= u - \sum_{i=1}^{k} d_i \theta(\sum_{j=1}^{v_i} q_j \theta(A'_{ij} x + B'_{ij} y + C'_{ij}) - r_i).$$

Теперь $\mu(x + y + NL(x, y), x, y) = x + y$.
Мы получили сумматор, а значит базис пространства кусочно-параллельных функций принадлежит $[U]$, а поэтому $[U]$ совпадает со множеством $PP$.

## 4. Заключение

Таким образом в настоящей работе решена задача Слупецкого для класса кусочно-параллельных функций. Автор выражает благодарность своему научному руководителю А.А. Часовских.

## Список литературы

[1] В. С. Половников, “О задаче проверки функциональной полноты в классе кусочно-параллельных функций”, *Вестн. Моск. ун-та. Сер. 1. Матем., мех.*, 2008, № 6, 31–35.

[2] Половников В.С., “О нелинейной сложности нейронных схем Мак-Каллока-Питтса.”, *М., Интеллектуальные системы.*, 2007, № 11, 261-275.

## Classes of piecewise parallel functions containing all single functions

**A. Otroschenko**

For a class of piecewise-parallel functions implemented by schemes of linear elements and Heaviside functions, an algorithm for checking the completeness of finite subsets supplemented by single functions is obtained. Thus, for this class solved the Slupetski problem
**Keyword:** The piecewise-linear function piecewise-parallel function, the completeness problem, the Slupetski criterion.

# Часть 3.
# Математические модели

# Оценка количества разметок графов групповых автоматов

Ищенко Р.А.[1]

Если в диаграмме Мура автомата без выходов убрать информацию о входных буквах, то получится ориентированный граф. Обратная операция, когда эта информация восстанавливается, называется разметкой графа автомата. В этой статье приводятся оценки числа разметок графов, приводящих к групповым автоматам.

**Ключевые слова:** групповой автомат, граф переходов, диаграмма Мура, перманент матрицы, факторизация.

## 1. Введение

Групповые автоматы без выхода $V = (A, Q, \varphi)$ таковы, что для любого $\alpha \in A^*$ отображение $\varphi_\alpha(q) = \varphi(q, \alpha)$ есть перестановка на множестве $Q$. Здесь $A$ — входной алфавит, $Q$ — множество состояний, $\varphi : Q \times A \to Q$ — функция переходов автомата $V$ [1]. Класс групповых автоматов обладает рядом интересных особенностей, что было отмечено в работах [2, 3]. Пусть множество $E = \{(q, \varphi(q, \alpha)) \,|q \in Q, \alpha \in A\}$ образует ребра ориентированного графа $G = (Q, E)$. Автор рассматривает задачу восстановления группового автомата $V = (A, Q, \varphi)$ по заданному графу $G = (Q, E)$. Критерий возможности такого восстановления был рассмотрен автором в работе [4]. В данной статье изучен вопрос оценки количества таких восстановлений.

Для случая автомата с входным алфавитом из двух элементов найдена точная формула числа восстановлений. Кроме того, приводится критерий существования единственного восстановления.

Введем необходимые понятия и определения.

**Определение 1.** *Графом автомата* $V = (A, Q, \varphi)$ называется размеченный ориентированный граф $G = (Q, W, f)$, вершины которого соответствуют состояниям автомата, при этом

[1] *Ищенко Роман Андреевич* — аспирант каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: ishchenko.roman1@gmail.com.

Ishchenko Roman Andreevich— graduate student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

$$e = (q_i, q_j) \in W,\ f(e) = a \Leftrightarrow \varphi(q_i, a) = q_j,$$

где $f : W \to A$, $a \in A$.

**Определение 2.** *Групповым графом* будем называть ориентированный граф, ребра которого могут быть размечены таким образом, что образованный граф является графом некоторого группового автомата. Такую разметку будем называть г—*разметкой* или просто *разметкой*. Как было показано в [4]:

**Утверждение 1.** *Граф $G(Q, W)$ — групповой тогда и только тогда, когда существует такое число $m$, что для любой вершины $q \in Q$ входящая и исходящая степени вершины равны $m$ (количеству элементов в алфавите соответствующего автомата).*

Ниже рассматривается *число разметок группового графа $G$*. При этом мы будем различать разметки с точностью до замены букв и/или кратных ребер. К примеру, две нижеприведенные разметки будут одинаковыми (Рис.1).

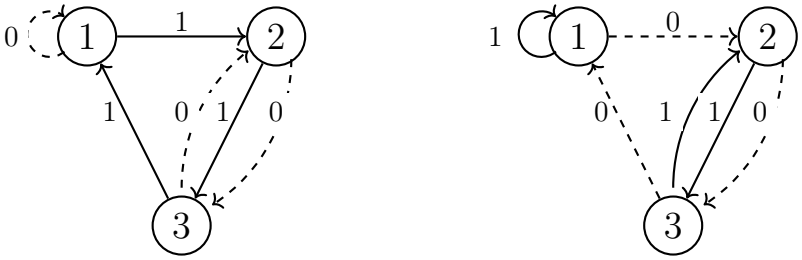



Рис. 1. Две одинаковые разметки группового графа.

Можно показать, что число различных разметок графа равно числу различных разложений соответствующей графу матрицы инцидентности в сумму матриц перестановок.

**Определение 3.** *Матрица инцидентности $A$ ориентированного графа $G$* с числом вершин $n$ (в дальнейшем просто *«матрица графа $G$»*) — это квадратная матрица порядка $n$, где значение элемента $a_{ij}$ равно числу рёбер из $i$-ой вершины в $j$-ую вершину графа $G$.

**Определение 4.** *Матрицей перестановки $P_\sigma$* назовем матрицу размера $n \times n$ вида

$P_\sigma = \begin{pmatrix} e_{\sigma(1)} \\ e_{\sigma(2)} \\ \cdots \\ e_{\sigma(n)} \end{pmatrix}$, где $e_i$ — вектор длины $n$, $i$-й элемент которого равен 1, а остальные равны 0.

Заметим, что утверждение 1 может быть переформулировано в терминах матриц следующим образом: *граф $G$ — групповой тогда и только*

*тогда, когда найдется такое число $m$, что сумма чисел в любой строке и любом столбце его матрицы равна $m$.* Такую матрицу мы будем называть *групповой*, а число $m$ — *степенью матрицы.*

По определению группового графа каждый подграф графа группового автомата из ребер с заданной буквой будет иметь матрицу инцидентности, являющейся матрицей некоторой перестановки. Множество всех матриц перестановок, являющихся подматрицами матрицы $A$ (в дальнейшем иногда *«подматриц перестановок»*) будем обозначать $P(A)$. Натуральное число $k$ будем называть *кратностью подматрицы перестановки* $P$, если $P^k \subseteq A$, где $P^k = \underbrace{P * \cdots * P}_{k}$, а $P^{k+1} \not\subseteq A$.

*Кратностью* $k$ матрицы $A$ будем называть максимальную кратность её подматрицы перестановки.

**Определение 5.** *Перманентом матрицы* $A$ называется число

$$Per(A) = \sum_{\pi \in S_n} \prod_{i=1}^{n} a_{i,\pi_i} = \sum_{\pi \in S_n} a_{1,\pi_1} a_{2,\pi_2} \dots a_{n,\pi_n},$$

где сумма берется по всем перестановкам $\pi$ чисел от 1 до $n$.

Обозначим $Per(n,k)$ максимально возможное значение перманента для бинарных (состоящих только из 0 и 1) групповых матриц порядка $n$ степени $k$. Множество всех разметок матрицы $A$ будем обозначать $C(A)$, количество разметок матрицы $A$ обозначим $g(A)$, $g(A) = |C(A)|$.

Обозначим множество разметок матрицы $A$ степени $m$, содержащих подматрицу $B$ степени $k$, как $C(A|B)$, т.е. $C(A|B) = \left\{ \{P_1, P_2, \dots, P_m\} \in C(A) \mid \exists\, i_1, \dots, i_k \ \bigcup_{j=1}^{k} P_{i_k} = B \right\}$.

Обозначим также $g(A|B) = |C(A|B)|$.

Перед тем, как перейти к основным результатам, докажем несколько свойств групповых матриц.

## 2. Свойства групповых матриц

**Утверждение 2.** *Для групповой матрицы $A$ степени $m$ и групповой подматрицы $B$ степени $k$ справедливо:*

$$g(A|B) = g(A - B) \cdot g(B).$$

*Доказательство.* Утверждение очевидно следует из того, что $C(A|B) = C(A - B) \times C(B)$, где $\times$ — декартово произведение множеств. □

**Утверждение 3.** *Для любой групповой матрицы $A$ и групповой подматрицы $B$ справедливо $g(B) \leqslant g(A)$.*

*Доказательство.* $g(B) \leqslant g(A-B) \cdot g(B) = g(A|B) \leqslant g(A)$. □

**Утверждение 4.** *При перестановке строк или столбцов групповой матрицы количество разметок не меняется.*

*Доказательство.* Без ограничения общности, пусть матрица $A'$ получена из матрицы $A$ степени $m$ путем перестановки строк $i$ и $j$. Тогда множество перманентных подматриц матриц $P_1, P_2, \ldots, P_m$ является разметкой матрицы $A$ тогда и только тогда, когда множество перманентных подматриц $P'_1,\ P'_2, \ldots, P'_m$ — правильная разметка матрицы $A'$, где для любого $i$ $P'_i$ получена из матрицы $P_i$ перестановкой строк $i$ и $j$. □

## 3. Оценки количества разметок

Для оценки количества разметок выведем в явном виде формулу зависимости количества разметок матрицы $A$ от количества разметок её подматриц.

**Теорема 1.** *Пусть $P(A) = P_1, \ldots, P_s$ — множество подматриц перестановок матрицы $A$ степени $m$, при этом для любого $i \in \{1, \ldots, s\}$ кратность подматрицы $P_i$ равна $k_i$, тогда:*

$$g(A) = \frac{\sum_{i=1}^{s} \sum_{j=1}^{k_i} g(A - jP_i)}{m}.$$

*Доказательство.* Поставим матрице $A$ в соответствие гиперграф $G'$ следующим образом:

- Каждой подматрице перестановки $P_i$ поставим в соответствие $k_i$ вершин графа $P_i^1, P_i^2, \ldots, P_i^{k_i}$.
- Для каждого разложения $A$ на матрицы перестановки $A = \sum_{i=1}^{s} j_i P_i$ соответствующей разметке $\{\underbrace{P_1, \ldots, P_1}_{j_1}, \ldots, \underbrace{P_s, \ldots, P_s}_{j_s}\}$ поставим в

  соответствие ребро из $m$ вершин $\{P_1^1,\ P_1^2, \ldots,\ P_1^{j_1}, \ldots, P_s^1,\ P_s^2, \ldots,\ P_s^{j_s}\}$.

Заметим, что $g(A)$ равно количеству ребер гиперграфа $G'$. Так как каждое ребро соединяет ровно $m$ вершин, то $g(A) = \frac{\sum_{P \in V} \deg P}{m} = \frac{\sum_{i=1}^{s} \sum_{j=1}^{k_i} \deg P_i^j}{m} = \frac{\sum_{i=1}^{s} \sum_{j=1}^{k_i} g(A - jP_i)}{m}$, так как $\deg P_i^j$ равно количеству разметок матрицы $A$, содержащих ровно $j$ подматриц $P_i$ (далее применяем Утв 1). Теорема доказана. □

**Следствие 1.** *Для групповой бинарной матрицы $A$ степени $m$ справедливо*

$$g(A) = \frac{\sum_{P \in P(G)} g(A - P)}{m}.$$

**Теорема 2.** *Для групповой матрицы $A$ степени $m$ порядка $n$ выполнено*

$$g\left(A\right) \leqslant \prod_{i=1}^{m} Per(n, i).$$

*Доказательство.* Докажем оценку индукцией по $m$.

**База индукции ($\boldsymbol{m = 1}$).** Групповая матрица степени 1 является матрицей перестановки и очевидным образом имеет единственную разметку. При этом, подставляя в формулу $m = 1$, получаем:

$$\prod_{i=1}^{m} Per\left(n, i\right) = Per\left(n, 1\right) = 1.$$

**Индуктивный переход ($\boldsymbol{m - 1 \ \rightarrow \ m}$).** Предположим, что утверждение справедливо для всех групповых матриц степени $m - 1$. Рассмотрим групповую матрицу $A = (a_{i,j})$ степени $m$. Пусть $P\left(A\right) = \{P_1, \ldots, P_s\}$ — множество подматриц перестановок матрицы $A$ степени $m$, при этом для любого $i \in \{1, \ldots, s\}$ кратность подматрицы $P_i$ равна $k_i$, кратность $A$ равна $k = \max\limits_{i \in \{1, \ldots, s\}} k_i$.

Рассмотрим бинарную матрицу $A' = (a'_{i,j})$ порядка $n$, в которой

$$a'_{i,j} = \begin{cases} 1, & \text{если } a_{i,j} \neq 0, \\ 0, & \text{если } a_{i,j} = 0, \end{cases} \text{где } i, j \ \in \{1, \ldots n\}.$$

Заметим, что $Per\left(A'\right) = s$.

Пользуясь соответственно Теоремой 1, Утверждением 2, предположением индукции, определением кратности матрицы, неравенством $k \leqslant m$ и определением $Per(n, m)$, получаем:

$$g\left(A\right) = \frac{\sum_{i=1}^{s} \sum_{j=1}^{k_i} g\left(A - jP_i\right)}{m} \leqslant \frac{\sum_{i=1}^{s} k_i \cdot g\left(A - P_i\right)}{m} \leqslant$$

$$\leqslant \frac{\sum_{i=1}^{s} \left(k_i \cdot \prod_{j=1}^{m-1} Per\left(n, j\right)\right)}{m} = \frac{\prod_{j=1}^{m-1} Per\left(n, j\right) \cdot \sum_{i=1}^{s} k_i}{m} \leqslant$$

$$\leqslant \frac{sk}{m} \cdot \prod_{j=1}^{m-1} Per\left(n, j\right) \leqslant \frac{Per\left(A'\right) \cdot m}{m} \cdot \prod_{j=1}^{m-1} Per\left(n, j\right) \leqslant \prod_{j=1}^{m} Per\left(n, j\right).$$

Теорема доказана. □

Заметим, что для случая бинарных матриц оценка Теоремы 2 может быть улучшена:

**Теорема 2.1.** *Для групповой бинарной матрицы $A$ степени $m$ порядка $n$ выполнено*

$$g(A) \leqslant \frac{1}{m!} \prod_{i=1}^{m} Per(n,i).$$

Для доказательства теоремы достаточно повторить шаги доказательства Теоремы 2, учитывая, что для любого $i \in \{1,\ldots,s\}$ $k_i = k = 1$.

Приведем верхние оценки перманента матрицы и выведем с их помощью следствия из Теорем 2 и 3.

**Утверждение 5** ([5, 6])**.** *Пусть $A = (a_{i,j})$ — бинарная матрица порядка $n$. Обозначим сумму чисел в $i$-ой строке как $r_i = \sum_{j=1}^{n} a_{i,j}$, $i = 1,2,\ldots,n$. Тогда*

$$Per(A) \leqslant \prod_{i=1}^{n} (r_i!)^{\frac{1}{r_i}} \leqslant \prod_{i=1}^{n} \frac{r_i+1}{2}.$$

**Следствие 2.** *Для групповой матрицы $A$ степени $m$ порядка $n$ выполнено*

$$g(A) \leqslant \prod_{i=1}^{m} (i!)^{\frac{n}{i}} \leqslant \left(\frac{(m+1)!}{2^m}\right)^n.$$

*Доказательство.* Используя Теорему 2 и подставляя для любого $i = 1,2,\ldots,n$ в неравенства Утверждения 3 вместо сумм строк $r_i$ степень соответствующей матрицы, получаем:

$$g(A) \leqslant \prod_{i=1}^{m} Per(n,i) \leqslant \prod_{i=1}^{m}\prod_{j=1}^{n} (i!)^{\frac{1}{i}} \leqslant \prod_{i=1}^{m}\prod_{j=1}^{n} \frac{i+1}{2};$$

$$g(A) \leqslant \prod_{i=1}^{m} (i!)^{\frac{n}{i}} \leqslant \left(\frac{(m+1)!}{2^m}\right)^n.$$

Следствие доказано. □

Аналогично получаем

**Следствие 2.1.** *Для групповой бинарной матрицы $A$ степени $m$ порядка $n$ выполнено*

$$g(A) \leqslant \frac{1}{m!} \prod_{i=1}^{m} (i!)^{\frac{n}{i}} \leqslant \frac{1}{m!} \left(\frac{(m+1)!}{2^m}\right)^n.$$

Докажем нижнюю оценку количества разметок бинарной матрицы при помощи нижней оценки перманента.

**Утверждение 6** ([7]). *Пусть $A$ — бинарная матрица порядка $n$ с $Per(A) > 0$. Пусть вектор $R$ сумм строк матрицы $A$, $R = (r_1, r_2, \ldots, r_n)$ — невозрастающий. Тогда*

$$Per(A) \geqslant \prod_{i=1}^{n} \max\{1, r_i - n + i\}.$$

**Следствие 3.** *Перманент любой матрицы $A$ степени $m$ составляет не менее $m!$*

*Доказательство.*

$$Per\,(A) \geqslant \prod_{i=1}^{n} \max\{1, m - n + i\} = \prod_{i=1}^{n-m} 1 \cdot \prod_{i=n-m+1}^{n} (m - n + i) = m!$$

□

**Теорема 3.** *Для групповой бинарной матрицы $A$ степени $m$ порядка $n$ выполнено*

$$g\,(A) \geqslant \prod_{k=0}^{m-1} k!$$

*Доказательство.* Докажем оценку индукцией по $m$.

**База индукции ($m = 1$).** Любая групповая матрица степени 1 является матрицей перестановки и имеет единственную разметку. При этом, подставляя в формулу $m = 1$, получаем

$$\prod_{k=0}^{m-1} k! = 0! = 1.$$

**Индуктивный переход ($m - 1 \to m$).** Предположим, что утверждение справедливо для всех групповых бинарных матриц степени $m-1$. Рассмотрим групповую матрицу $A = (a_{i,j})$ степени $m$.

Пусть $P\,(A) = \{P_1, \ldots, P_s\}$ — множество подматриц перестановок матрицы $A$, тогда

$$g\,(A) = \frac{1}{m} \sum_{i=1}^{s} g\,(A - P_i) \geqslant \frac{1}{m} Per\,(A) \cdot \min_{i=1,\ldots,s} g\,(A - P_i) \geqslant$$

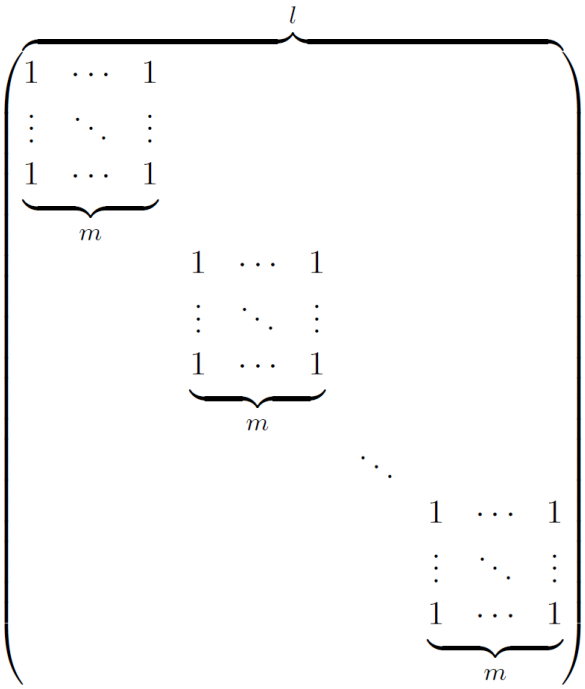
$$\geqslant \frac{1}{m}\, m! \prod_{k=0}^{m-2} k! = \prod_{k=0}^{m-1} k!$$

Теорема доказана. □

В качестве нижней оценки максимального количества разметок для матрицы с заданными порядком и кратностью оценим число разметок для матриц определенного вида в случае $n \vdots m$.

**Теорема 4.** *Для любых натуральных $m$ и $n$ таких, что $n \vdots m$, существует такая групповая матрица степени $m$ порядка $n$, что $g(A) \geqslant \frac{1}{m!}\left(\frac{m\prod_{k=0}^{m} k!}{3}\right)^{\frac{n}{m}}$.*

*Доказательство.* Пусть $n = ml$. Рассмотрим групповую матрицу , на диагонали которой расположены полные подматрицы степени $m$:




Докажем, что

$$g(A) = (m!)^{l-1} \cdot \left(g(K_m)\right)^l. \tag{1}$$

Каждой подматрице перестановки матрицы $A$ можно поставить в соответствие строку из $l$ матриц $(P_1,\ P_2,\ \ldots,\ P_l)$, где $P_i$ — перманентная подматрица $K_m$. Так как каждая из «клеток» $K_m$ может быть разложена на подматрицы перестановки независимо от других «клеток», то задача поиска количества разметок матрицы может быть переформулирована как определение количества различных с точностью до перестановки строк таблиц размера $l \times m$, в ячейках которых расположены подматрицы перестановки $K_m$, а объединение матриц в любом столбце дает $K_m$.

| | | | |
|---|---|---|---|
| $P_1^1$ | $P_2^1$ | $\ldots$ | $P_l^1$ |
| $P_1^2$ | $P_2^2$ | $\ldots$ | $P_l^2$ |
| $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ |
| $P_1^l$ | $P_2^l$ | $\ldots$ | $P_l^l$ |

Таблица 1. Представление разложения матрицы A в виде таблицы из перманентных подматрице.

В таблице 1:

- $\bigcup_{j=1}^{m} P_1^j = K_m$ для любого $i \in \{1, \ldots, l\}$,
- $\bigcup_{i=1}^{l} P_i^j$ — перманентная подматрица $A$ для любого $j \in \{1, \ldots, m\}$,
- $\bigcup_{i=1}^{l} \bigcup_{j=1}^{m} P_i^j = A$.

Количество различных перестановок из $m$ элементов равно $m!$, поэтому количество различных возможных столбцов таблицы равно $m! \cdot g(K_m)$, следовательно, количество различных наборов из $l$ столбцов равно $\left(m! \cdot g(K_m)\right)^l$, а с точностью до перестановки $m$ строк — $\frac{\left(m! \cdot g(K_m)\right)^l}{m!} = (m!)^{l-1} \cdot \left(g\left(K_m\right)\right)^l$.

Докажем, что

$$g\left(K_m\right) \geqslant \frac{\prod_{k=0}^{m} k!}{3\,(m-1)!}. \tag{2}$$

Для $m = 1, 2$ утверждение очевидно, рассмотрим $m \geqslant 3$. Согласно Теореме 1, $g\left(K_m\right) = \frac{\sum_{P \in P(K_m)} g(K_m - P)}{m}$. Заметим, что для любой перманентной матрицы $P \in P\left(K_m\right)$ из матрицы $K_m - P$ может быть получена матрица $K_m - E$ (где $E$ — единичная матрица) путем перестановки соответствующих строк: достаточно сделать $n$ шагов, где на $i$-м шаге строка с нулем в $i$-м столбце при необходимости переставляется с $i$-ой строкой.

$$\begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix} \xrightarrow{r_1 \leftrightarrow r_3} \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix} \xrightarrow{r_2 \leftrightarrow r_3} \tag{3}$$

$$\xrightarrow{r_2 \leftrightarrow r_3} \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix} \xrightarrow{r_3 \leftrightarrow r_4} \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \tag{4}$$

С учетом Утверждения 4 получаем:

$$g\left(K_m\right) = \frac{\sum_{P \in P(K_m)} g(K_m - P)}{m} = \frac{Per\left(K_m\right) \cdot g(K_m - E)}{m} =$$
$$= \frac{Per\left(K_m\right) \cdot \sum_{P \in P(K_m - E)} g(K_m - E - P)}{m(m-1)} \geqslant$$
$$\geqslant \frac{Per\left(K_m\right) \cdot Per\left(K_m - E\right) \cdot \min_{P \in P(K_m - E)} g\left(K_m - E - P\right)}{m(m-1)} \geqslant$$
$$\geqslant \frac{Per\left(K_m\right) \cdot Per\left(K_m - E\right) \cdot \prod_{k=0}^{m-3} k!}{m(m-1)}$$

Заметим, что $Per\left(K_m\right)$ равен числу всевозможных перестановок из $m$ элементов, т.е. $Per\left(K_m\right) = m!$. Задача поиска $Per\left(K_m - E\right)$ эквивалентна задаче определения количества перестановок на множестве из $m$ элементов, которые не оставляют ни одного элемента фиксированным. Можно показать [8], что число таких перестановок равно $m! \sum_{k=0}^{m} \frac{(-1)^k}{k!}$. Легко убедиться, что $\sum_{k=0}^{m} \frac{(-1)^k}{k!} \geqslant \frac{1}{3}$, и после ряда преобразований мы получаем выражение 2.

Подставляя выражение 2 в выражение 1, получаем:

$$g\left(A\right) \geqslant (m!)^{l-1} \cdot \left(\frac{\prod_{k=0}^{m} k!}{3\,(m-1)!}\right)^l = \frac{1}{m!}\left(\frac{m! \prod_{k=0}^{m} k!}{3\,(m-1)!}\right)^l = \frac{1}{m!}\left(\frac{m \prod_{k=0}^{m} k!}{3}\right)^l.$$

Подставляя $\frac{n}{m}$ вместо $l$, получаем доказательство Теоремы. □

## 4. Случай $m = 2$

Для групповых графов в алфавите из двух элементов количество правильных разметок может быть в явном виде выражено через перманент соответствующей ей бинарной матрицы.

**Теорема 5.** *Пусть $A = (a_{i,j})$ — групповая матрица степени 2 порядка $n$. Рассмотрим бинарную матрицу $A' = (a'_{i,j})$ порядка $n$, в которой $a'_{i,j} = \begin{cases} 1, & \text{если } a_{i,j} \neq 0, \\ 0, & \text{если } a_{i,j} = 0, \end{cases}$ где $i, j \in \{1, \ldots n\}$. Тогда количество разметок матрицы равно:*

$$g(A') = \begin{cases} \frac{Per(A')}{2}, & \text{если } A \neq 2E, \\ 1, & \text{если } A = 2E. \end{cases}$$

*Доказательство.* Случай $A = 2E$ очевиден. Заметим, что если $A \neq 2E$, то кратности всех перманентных подматриц равны 1. Пользуясь Теоремой 1 и тем, что для любой подматрицы $P \in P(A)$ степень $g(A - P)$

равна 1, получаем:

$$g\left(A'\right) = \frac{\sum_{P \in P(A)} g(A-P)}{2} = \frac{Per\left(A'\right)}{2}.$$

□

**Теорема 6.** *Количество разметок групповой матрицы $A$ степени 2 не превышает $g(A) \leqslant 2^{\lfloor \frac{n}{2} \rfloor - 1}$, и эта оценка достижима.*

*Доказательство.* Можно показать [9], что перманент групповой матрицы $A$ степени 2 не превышает $2^{\lfloor \frac{n}{2} \rfloor}$, что с учетом Теоремы 5 доказывает верхнюю оценку.

В качестве примеров матриц, на которых эта оценка достижима, для случая четного $n$ рассмотрим групповую матрицу, у которой на диагонали расположены полные подматрицы степени 2, для случая нечетного $n$ — ее незначительную модификацию.

**четное $n$**

$$\begin{pmatrix} 1 & 1 & & & & & \\ 1 & 1 & & & & & \\ & & 1 & 1 & & & \\ & & 1 & 1 & & & \\ & & & & \ddots & & \\ & & & & & 1 & 1 \\ & & & & & 1 & 1 \end{pmatrix}$$

**нечетное $n$**

$$\begin{pmatrix} 1 & 1 & & & & & \\ 1 & 1 & & & & & \\ & & 1 & 1 & & & \\ & & 1 & 1 & & & \\ & & & & \ddots & & \\ & & & & & 1 & 1 & 0 \\ & & & & & 1 & 0 & 1 \\ & & & & & 0 & 1 & 1 \end{pmatrix}$$

Очевидно, что перманент данных матриц равен соответственно $2^{\frac{n}{2}}$ и $2^{\frac{n-1}{2}}$, поэтому количество разметок в обоих случаях равно $2^{\lfloor \frac{n}{2} \rfloor - 1}$. Теорема доказана. □

## 5. Критерий единственности разложения

**Теорема 7.** *Пусть $A$ — групповая матрица степени $m$, $P(A) = \{P_1, \ldots, P_s\}$ — множество подматриц перестановок, при этом для любого $i \in \{1, \ldots, s\}$ кратность подматрицы $P_i$ равна $k_i$, тогда матрица $A$ имеет единственную разметку тогда и только тогда, когда $\sum_{i=1}^{s} k_i = m$.*

*Доказательство.* **Необходимость**. Пусть матрица $A$ имеет единственную правильную разметку $A = p_1 P_1 + \ldots + p_s P_s$, $p_i \in \mathbb{N}_0$ — количество матриц $P_i$ в разложении матрицы $A$, $\sum_{i=1}^{s} p_i = m$. Заметим, что тогда $\sum_{i=1}^{s} k_i \geqslant \sum_{i=1}^{s} p_i = m$.

Предположим, что $\sum_{i=1}^{s} k_i > m$, тогда существует такое $l \in \{1, \ldots, s\}$, что $k_l > p_l$. Рассмотрим матрицу $A' = A - k_l P_l$. Граф $A'$ является групповым степени $m - k_l$ и может быть разложен в сумму $A' = p'_1 P_1 + \ldots + p'_s P_s$, при этом $p'_l = 0$. Тогда разложение $A = p'_1 P_1 + \ldots + k_l P_l + \ldots + p'_s P_s$ отличается от исходного ввиду того, что $k_l > p_l$, противоречие.

**Достаточность**. Пусть $\sum_{i=1}^{s} k_i = m$. Предположим, что граф $G$ имеет не менее двух различных правильных разметок:

$$\begin{cases} A = p_1 P_1 + \cdots + p_s P_s \\ A = p'_1 P_1 + \cdots + p'_s P_s \\ \quad \ldots \end{cases}$$

Существует такое $l \in \{1, \ldots, n\}$, что $p'_l \neq p_l$, без ограничения общности $p'_l > p_l$. Тогда $\sum_{i=1}^{s} k_i \geqslant \sum_{i=1}^{s} \max(p_i, p'_i) \geqslant \sum_{i \in \{1,\ldots,s\} \setminus l} p_i + p'_l > \sum_{i=1}^{s} p_i = m$. Противоречие. Теорема доказана. □

## Список литературы


[1] Кудрявцев В. Б., Алешин С. В., Подколзин А. С., "Введение в теорию автоматов", *Наука*, 1985, 320.

[2] Алешин С. В., "Алгебраические системы автоматов", *МАКС Пресс*, 2016, 192.

[3] Бабин Д. Н., "Особенности схем автоматных функций с операцией суперпозиции", *МАКС Пресс*, 2019, 42.

[4] Ищенко Р. А., "Графы групповых автоматов", *Интеллектуальные системы. Теория и приложения*, **21**:2 (2017), 111–116.

[5] Brègman, L. M., "Some properties of nonnegative matrices and their permanents", *Doklady Akademii Nauk*, **211**:1 (1973), 27–30.

[6] Minc, H., "Upper bounds for permanents of (0, 1)-matrices", *Bulletin of the American Mathematical Society*, **69**:6 (1963), 789–791.

[7] Ostrand, P. A, "Systems of distinct representatives, II", *Journal of Mathematical Analysis and Applications*, **32**:1 (1970), 1–4.

[8] Rosen, K. H., *Handbook of discrete and combinatorial mathematics*, 2000, 1183.

[9] Van Lint J. H., Wilson R. M., Wilson R. M., *A course in combinatorics*, Cambridge university press, Cambridge, 2001, 616 pp.


### Estimation of the number of labelings of group automata graphs


**Ishchenko R.A.**



If we remove symbols of a state diagram, then we get a directed graph. The inverse operation, when this information is restored, is called graph labeling. This article estimates the number of graph labelings that lead to a group automata.



*Keywords:* group automata, transition graph, state diagram, permanent, matrix decomposition.

# О кодовом расстоянии в одном классе квантовых LDPC кодов

Калачев Г.В.[1], Пантелеев П.А.[2]

В работе рассматривается одно семейство квантовых LDPC кодов с весом стабилизатора 6 и двумя логическими кубитами, где имеется фрактальная структура некоторых логических операторов. Эти коды можно представить в виде локальных кодов на трёхмерной решётке $L \times L \times L$ с периодическими граничными условиями. Для этого семейства кодов доказана нижняя оценка кодового расстояния $\Omega(L^\alpha)$, где $\alpha = \log_2(2(\sqrt{5}-1)) \approx 1.306$.

**Ключевые слова:** квантовый LDPC код, локальный квантовый код, кодовое расстояние, линейный клеточный автомат, фрактальная размерность.

## 1. Введение

Одним из основных препятствий на пути к созданию полноценного универсального квантового компьютера является достаточно высокая ненадёжность его компонент. Данное обстоятельство прежде всего связано с невозможностью в процессе вычисления на квантовом компьютере идеально изолировать от окружающей среды элементарные ячейки его памяти — *кубиты*, хранящие текущее состояние. Идея квантовых кодов, предложенная П. Шором [1], теоретически позволяют решить данную проблему за счёт кодирования абстрактного $k$-кубитного состояния квантового компьютера (*логические* кубиты) физически реализуемым $n$-кубитным состоянием (*физические* кубиты). При этом параметр $R = k/n < 1$, называемый *скоростью* кода, характеризует избыточность

[1] *Калачев Глеб Вячеславович* — к.ф.-м.н., м.н.с. лаборатории проблем теоретической кибернетики мех.-мат. ф-та МГУ, e-mail: gleb.kalachev@yandex.ru.

Kalachev Gleb Vyacheslavovich — Candidate of Physical and Mathematical Sciences, Junior Researcher, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Problems of Theorecical Cybernetics Lab.

[2] *Пантелеев Павел Анатольевич* — к.ф.-м.н., н.с. каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: panpavel@yandex.ru.

Panteleev Pavel Anatolyevich — Candidate of Physical and Mathematical Sciences, Researcher, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

такого кодирования. Другой важной характеристикой квантового кода, показывающей его способность исправлять ошибки, возникающие в физических кубитах, является *кодовое расстояние*. Аналогично классическим кодам, *кодовое* (*минимальное*) *расстояние* квантового кода можно определить как максимальное число $d$ такое, что любая ошибка, изменяющая квантовое состояние и затрагивающая менее чем $d$ физических кубитов, может быть обнаружена.

В квантовой механике состояние $n$-кубитной квантовой системы может быть описано вектором из $2^n$-мерного комплексного гильбертова пространства $\mathbb{C}^{2^n}$ и, в общем случае, квантовый код определяется формально как $2^k$-мерное подпространство $Q$ пространства $\mathbb{C}^{2^n}$, где параметры $n$ и $k$ называются его *длинной* и *размерностью*, соответственно. При идеальном функционировании квантового компьютера, защищённого кодом $Q$, его состояние должно быть одним из элементов $Q$, которые по аналогии с классическими кодами называют (*квантовыми*) *кодовыми словами*. Однако, вследствие ошибок, возникающих в кубитах, состоянием квантового компьютера являются искажённые квантовые кодовые слова, и задача декодера для $Q$ состоит в периодическом обнаружении и исправлении этих ошибок.

Легко видеть, что приведённое выше определение квантового кода очень общее, и, вообще говоря, не является конструктивным, так как для фиксированных $n$ и $k$ существует континуум различных квантовых кодов. Поэтому, как правило, рассматриваются какие-то специальные классы квантовых кодов, которые можно задать конструктивно. Один из наиболее известных таких классов — это класс *стабилизирующих квантовых кодов*[1] (анг. stabilizer codes) [2], являющихся, в некотором смысле, квантовым аналогом классических линейных кодов. В данной работе мы рассматриваем частный случай стабилизирующих кодов, называемых кодами Кальдербанка-Шора-Стина (CSS коды) [3, 4].

Важным достоинством CSS кодов, выделяющим их в классе стабилизирующих кодов, является простота их задания и естественная связь с классическими линейными кодами. Фактически, можно рассматривать CSS код как пару классических двоичных линейных кодов одинаковой длины $n$, заданных проверочными матрицами $H_X$ и $H_Z$ такими, что любая строка $H_X$ ортогональна[2] любой строке $H_Z$. В дальнейшим под словом квантовый код мы всегда будем подразумевать CSS код.

Со строками проверочных матриц $H_X$ и $H_Z$ связывают эрмитовы операторы, называемые *стабилизаторами*, описывающие квантовые изме-

[1] В литературе эти коды также еще называются *симплектическими* или *аддитивными*.

[2] Ортогональность двоичных векторов $(u_1, \ldots, u_n)$ и $(v_1, \ldots, v_n)$ понимается как выполнение тождества $\sum_{i=1}^{n} u_i v_i \equiv 0 \pmod 2$.

рения над $n$ физическими кубитами. Случайный вектор $s$, являющийся результатом этих измерений, представляет собой аналог синдрома для классических линейных кодов. Он подаётся на вход декодеру, который пытается определить и исправить ошибку в $n$-кубитном квантовом состоянии. Отметим, что условие ортогональности строк матриц $H_X$ и $H_Z$ эквивалентно попарной коммутативности стабилизаторов, т.е. возможности совместного измерения всех компонент вектора $s$.

По определению, *весом* стабилизатора будем называть число ненулевых компонент в соответствующей строке проверочной матрицы. С физической точки зрения вес стабилизатора — это число физических кубитов с которыми надо провзаимодействовать для выполнения соответствующего квантового измерения. Поэтому с практической точки зрения особый интерес представляют квантовые LDPC коды (QLDPC коды), у которых обе проверочные матрицы $H_X$ и $H_Z$ разрежены. При этом разреженность обычно понимается как существование константы $w$, ограничивающей сверху веса строк и столбцов матриц $H_X$ и $H_Z$ при росте длины кода $n$. Последнее условие эквивалентно тому, что в каждом квантовом измерении участвует не более $w$ физических кубитов, и каждый кубит участвует не более чем в $2w$ измерениях. В отличии от классического случая, подобные ограничения чрезвычайно важны для квантовых кодов, так как в процессе измерения мы взаимодействуем с кубитами, и тем самым вносим в них ошибки.

На данный момент неизвестно, существуют ли семейства QLDPC кодов, имеющие линейно растущее кодовое расстояние $d$ даже при фиксированной размерности кода $k$. Однако имеются некоторые общие конструкции [6], позволяющие строить QLDPC коды с расстоянием, растущим, как $\Theta(\sqrt{n})$ и фиксированной скоростью кода $k/n$. Одно из лучших семейств QLDPC кодов, для которых известны оценки кодового расстояния, основано на специальном семействе метрик на некотором многообразии [7]. У этих кодов кодовое расстояние имеет порядок[3] $\sqrt{n\sqrt{\log n}}$. Отметим, что в недавних работах [8, 9] данный результат был несколько улучшен и получено семейство QLDPC кодов с расстоянием, растущим, как $\Omega(\sqrt{n}\log^k n)$ для любого $k$.

Также для физической реализации очень важно чтобы кубиты можно было расположить в $D$-мерном пространстве, где $D \leqslant 3$, так, чтобы связанные общим проверочным соотношением[4] кубиты были бы расположены локально, т.е. расстояние между ними ограничено константой при увеличении длины кода $n$. Квантовые LDPC коды, для которых такое расположение возможно, называются *$D$-локальными* или просто *ло-*

[3] В оригинальной работе [7] этот порядок ошибочно указан как $\sqrt{n\log n}$.

[4] Проверочные соотношения CSS кода — это проверочные соотношения двух соответствующих классических кодов.

*кальными*, если из контекста ясно о каком $D$ идёт речь. Для локальных кодов известна [10] верхняя оценка $d = O(n^{(D-1)/D})$, которая при $D = 2$ достигается на торическом коде [16, с. 97][17]. Для $D > 2$ неизвестно кодов, на которых эта оценка достигается. При $D = 3$ есть несколько семейств кодов [11, 14, 15], для которых потенциально расстояние может асимптотически расти быстрее чем $\sqrt{n}$, однако все известные нижние оценки на кодовое расстояние этих кодов не превосходят $\Omega(n^{\frac{1}{3}})$. У таких кодов некоторые логические операторы (недетектируемые ошибки, изменяющие квантовое состояние) имеют фрактальную структуру, поэтому их иногда называют *фрактальными*[5].

В работе рассматривается одно семейство QLDPC кодов с весом стабилизатора 6 и двумя логическими кубитами (т.е., $k = 2$), где также имеется фрактальная структура некоторых логических операторов. Эти коды можно представить в виде локальных кодов на трёхмерной решётке $\mathbb{Z}_L^3$ с периодическими граничными условиями, где $\mathbb{Z}_L$ — кольцо вычетов по модулю $L$. При этом в каждом узле решётки $\mathbb{Z}_L^3$ находится по два кубита, и тем самым общее число кубитов $n = 2L^3$. Для этого семейства кодов доказана нижняя оценка кодового расстояния $d = \Omega(L^\alpha) = \Omega(n^{\frac{1}{3}\alpha})$, где $\alpha = \log_2(2(\sqrt{5} - 1)) \approx 1.306$.

## 2. Определения и обозначения

### 2.1. Классические и квантовые коды

Обозначим через $\mathbb{F}_2$ конечное поле из двух элементов, а через $\mathbb{F}_2^n$ — $n$-мерное координатное векторное пространство над $\mathbb{F}_2$, элементы которого мы будем понимать как двоичные векторы-столбцы $(v_1, \ldots, v_n)^T$. *Весом* вектора $v \in \mathbb{F}_2^n$ будем называть количество его ненулевых элементов и обозначать $|v|$. Если у нас имеется $m \times n$ матрица $A$ над $\mathbb{F}_2$, то через $\langle A \rangle$ будем обозначать линейную оболочку строк матрицы $A$, а через $\ker A$ ядро линейного оператора $v \mapsto Av$.

Напомним, что классическим двоичным линейным $C$ кодом *длины* $n$ и *размерности* $k$ называют произвольное $k$-мерное линейное подпространство $n$-мерного векторного пространства $\mathbb{F}_2^n$, элементы которого называются *кодовыми словами*. Важной характеристикой кода, которая описывает его способность исправлять ошибки, является *кодовое* (*минимальное*) *расстояние* $d$, равное, в случае линейных кодов, минимальному весу ненулевого кодового слова, т.е. $d = \min_{v \in C \setminus \{0\}} |v|$. Так как $C$ является $k$-мерным линейным подпространством в $\mathbb{F}_2^n$, его можно задать как $C = \langle G \rangle$, т.е. как линейные комбинации строк некоторой матрицы $G$,

[5]Также широко используется название *квантовая фрактальная жидкость* (анг. quantum fractal liquid).

называемой *порождающей*. Линейное подпространство $C \subseteq \mathbb{F}_2^n$ можно также задать как $C = \ker H$, т.е. как множество решений системы линейных однородных уравнений, где матрица $H$ называется *проверочной*. При этом строки проверочной матрицы $H$ соответствуют уравнениям данной системы, которые мы будем называть *проверочными соотношениями* для кода $C$.

Обычно кодовые слова классического линейного кода $C \subseteq \mathbb{F}_2^n$, заданного проверочной матрицей $H$, понимаются как двоичные $n$-битные последовательности, которые мы передаём по каналу с ошибками. Однако кодовые слова из $C$ можно также интерпретировать и как вектора ошибок $x \in \mathbb{F}_2^n$, возникающие с передаваемыми $n$ битами, которые мы не можем детектировать при помощи проверочной матрицы $H$, т.е. когда $Hx = 0$. При этом самим векторам ошибок $x$ естественно соответствуют операторы ошибок $E_x \colon v \mapsto v + x$ действующие на множестве $n$-битных векторов. Если $c' = E_x(c)$, где $c \in C$, есть искажённое кодовое слово с которым произошла ошибка $x$, то вектор $s = Hc'$ называют *синдромом*. Поскольку $s = H(c + x) = Hx$ мы видим, что синдром не зависит от самого кодового слова $c$, а зависит только от произошедшей с ним ошибки $x$. Поэтому проверочные соотношения, соответствующие строкам проверочной матрицы $H$, можно также понимать как «элементарные измерения», которые мы производим над искажённым кодовым словом $c'$ для выявлении информации о произошедшей с ним ошибке $x$. Результатом такого измерения для $i$-й строки является значение $i$-й компоненты синдрома. При этом очевидно выполняются следующие условия:

- результат всех элементарных измерений равен нулю в точности для кодовых слов;
- для кодовых слов, искажённых оператором ошибки $E_x$, результат измерений описывается синдромом $s = Hx$.

В кубитах, в отличии от битов, может возникать континуум различных ошибок[6]. Однако можно показать [5, Глава 10], что для $n$-кубитного состояния защищённого квантовым кодом значение имеют только выделенное конечное подмножество ошибок $\mathcal{E}_n$, состоящее из $2^{2n}$ ошибок $E_{x,z}$, параметризованных всевозможными $x, z \in \mathbb{F}_2^n$. Напомним, что квантовым кодом $C$ называют произвольное $2^k$-мерное подпространство $2^n$-мерного комплексного гильбертова пространства $\mathbb{C}^{2^n}$, элементы которого мы называем (*квантовыми*) *кодовыми словами*.

Основная идея квантового CSS кода $C$ состоит в том, чтобы паре двоичных матриц $H_X$ и $H_Z$ с числом столбцов $n$, играющих роль прове-

[6] С точки зрения квантовой механики ошибкам над $n$-кубитным состоянием соответствуют унитарные операторы действующие на $\mathbb{C}^{2^n}$.

рочных матриц, сопоставить последовательность элементарных квантовых измерений[7] над $n$-кубитным состоянием (по одному измерению для каждой строки $H_X$ и $H_Z$) так, чтобы:

- результат всех элементарных квантовых измерений детерминированный и равен нулю в точности для квантовых кодовых слов;
- для квантовых кодовых слов, искажённых оператором ошибки $E_{x,z}$, результат измерений также детерминированный, и описывается векторами $s_X = H_X z$, $s_Z = H_Z x$, которые, как и в случае классических линейных кодов, называются *синдромами*.

Как видно синдром $s_X$ (соответственно $s_Z$) содержит информацию о компоненте $z$ (соответственно $x$) ошибки $E_{x,z} \in \mathcal{E}_n$ произошедшей с $n$-кубитным квантовым состоянием. На основании данной информации декодер для квантового CSS кода пытается найти наиболее вероятную ошибку $E_{x,z} \in \mathcal{E}_n$ и исправить ее.

Как хорошо известно, в квантовой механике в силу принципа неопределённости Гейзенберга не любые два измерения могут давать полностью детерминированный результат одновременно. Достаточным условием для этого является коммутируемость самосопряжённых операторов, соответствующих этим измерениям. По этой причине для возможности одновременного измерения всех компонент синдромов $s_X$ и $s_Z$ требуется, чтобы операторы, соответствующие элементарным измерениям для матриц $H_X$ и $H_Z$, были попарно коммутативны. Можно показать [5, с. 561], что последнее условие эквивалентно тому, что любая строка в матрице $H_X$ ортогональна любой строке в $H_Z$ с точки зрения стандартного скалярного произведения в векторном пространстве $\mathbb{F}_2^n$. Данное условие можно компактно сформулировать в матричном виде как:

$$H_X H_Z^T = 0. \tag{1}$$

В дальнейшем CSS код $C$, заданный парой матриц $H_X$ и $H_Z$ над полем $\mathbb{F}_2$ с одинаковым числом столбцов $n$, удовлетворяющих соотношению (1), будем обозначать через $\mathrm{CSS}(H_X, H_Z)$ и отождествлять с парой $(C_Z, C_X)$ классических двоичных линейных кодов $C_Z, C_X \subseteq \mathbb{F}_2^n$, заданных проверочными матрицами $H_X$, $H_Z$, соответственно. Заметим, что ошибки вида $E_{0,z}$ и $E_{x,0}$, где $z \in C_Z$, $x \in C_X$ не будут обнаружены квантовым кодом, поскольку в этом случае $s_X = s_Z = 0$. По этой причине далее кодовые слова из $C_Z$ (соответственно $C_X$) мы будем называть *$Z$-ошибками* (соответственно *$X$-ошибками*) кода $\mathrm{CSS}(H_X, H_Z)$.

[7] Результатом каждого элементарного квантового измерения является один бит, причём в общем случае этот результат не является детерминированным.

Отметим, что поскольку, в силу условия (1), строки матриц $H_X$ и $H_Z$ попарно ортогональны друг другу, каждая строка матрицы $H_X$ является $X$-ошибкой, а матрицы $H_Z$ — $Z$-ошибкой. Следовательно $\langle H_X \rangle \subseteq C_X$ и $\langle H_Z \rangle \subseteq C_Z$.

Когда мы рассматривали классические коды мы подмечали, что множество кодовых слов в точности совпадает со множеством всех ошибок $x \in \mathbb{F}_2^n$ которые мы не можем детектировать при вычислении синдрома. Однако, нулевому кодовому слову соответствует оператор ошибки $E_0 : x \mapsto x$, не изменяющий передаваемые по каналу кодовые слова. Оказывается, что в квантовом случае подобная ситуация встречается значительно чаще и можно показать [5, Глава 10], что любой оператор вида $E_{x,z}$, где $x \in \langle H_X \rangle$ и $z \in \langle H_Z \rangle$ тождественно действует на множестве квантовых кодовых слов. При этом любой другой оператор $E_{x,z}$ уже действует нетождественно на этом множестве. Данное обстоятельство мотивирует приведённое ниже определение.

**Определение 1.** $X$-ошибка $x \in C_X$ (соответственно, $Z$-ошибка $z \in C_Z$) называется *вырожденной*, если $x \in \langle H_X \rangle$ (соответственно, $z \in \langle H_Z \rangle$).

Заметим, что две $X$-ошибки $x, x' \in C_X$, отличающиеся на вырожденную $X$-ошибку (т.е. $x - x' \in \langle H_X \rangle$), задают операторы $E_{x,0}$ и $E_{x',0}$, действующие одинакового на множестве квантовых кодовых слов. Аналогичное утверждение справедливо и для $Z$-ошибок. Подобные ошибки, действующие одинаково на множестве квантовых кодовых слов, мы будем называть *эквивалентными*. Соответствующие данному отношению классы эквивалентности обычно называют *логическими ошибками*. Поэтому логические $X$-ошибки (соответственно $Z$-ошибки) представляют собой элементы факторпространства $C_X / \langle H_X \rangle$ (соответственно, $C_Z / \langle H_Z \rangle$).

**Определение 2.** *Размерностью* CSS кода называется размерность пространства его логических ошибок.

Убедимся, что определение корректно, т.е. размерности пространств логических $X$- и $Z$-ошибок совпадают. Действительно:

$$\dim(C_X / \langle H_X \rangle) = \dim(\ker H_Z) - \dim \langle H_X \rangle = n - \operatorname{rank} H_Z - \operatorname{rank} H_X,$$
$$\dim(C_Z / \langle H_Z \rangle) = \dim(\ker H_X) - \dim \langle H_Z \rangle = n - \operatorname{rank} H_X - \operatorname{rank} H_Z.$$

Таким образом, формула для размерности CSS кода имеет вид

$$\dim Q = n - \operatorname{rank} H_X - \operatorname{rank} H_Z. \tag{2}$$

Можно показать [5, с. 553], что введённое сейчас понятие размерности CSS кода согласуется с введённым ранее понятием размерности квантового кода общего вида, т.е. множество квантовых кодовых слов CSS кода образует $2^k$-мерное подпространство в $\mathbb{C}^{2^n}$.

**Определение 3.** *Кодовым* (*минимальным*) расстоянием CSS кода называется минимальный вес невырожденной $X$- или $Z$-ошибки.

В данной работе рассматриваются CSS коды специального вида [21], а именно, когда матрицы $H_X$ и $H_Z$ состоят из двух квадратных блоков одинакового размера. Блочные матрицы будем писать в квадратных скобках. Если матрица состоит из двух блоков $A$ и $B$, будем для наглядности разделять их вертикальной чертой: $[A \mid B]$. В этих обозначениях рассматриваемое семейство кодов задаётся матрицами

$$H_X = [A \mid B],$$
$$H_Z = \left[B^T \mid A^T\right],$$

где матрицы $A$ и $B$ коммутируют, т.е. $AB = BA$. Покажем, что в таком случае матрицы $H_X$ и $H_Z$ задают CSS код. Для этого достаточно убедится, что выполняется равенство (1). Действительно, имеем:

$$H_X H_Z^T = A(B^T)^T + B(A^T)^T = AB + BA = 0.$$

Кубиты полученного квантового кода естественно разбиваются на две группы одинакового размера — левую и правую, в зависимости от того, к какому блоку (левому или правому) матриц $H_X$ и $H_Z$ они относятся. В случаях когда кубиты будут размещаться в узлах какой-либо геометрической решётки мы будем размещать левый и соответствующий ему правый кубит в одном узле.

## 2.2. Задание кодов через групповые алгебры

Важный частный случай кодов из описанного выше семейства получается если матрицы $A$ и $B$ могут быть заданы элементами некоторой групповой алгебры [11, 14]. В связи с этим введём ещё несколько обозначений.

Циклическую группу порядка $\ell$, порождённую элементом $x$ будем обозначать $\langle x \rangle_\ell$.

Пусть $G$ — группа. Через $\mathbb{F}_2^G$ будем обозначать групповую алгебру группы $G$ над полем $\mathbb{F}_2$. Заметим, что если $G$ — абелева, то её можно представить в виде произведения циклических групп $\langle x_1 \rangle_{a_1} \times \cdots \times \langle x_n \rangle_{a_n}$. В этом случае групповая алгебра $\mathbb{F}_2^G$ изоморфна факторкольцу полиномов $\mathbb{F}_2[x_1, ..., x_n]/(x_1^{a_1} + 1, ..., x_n^{a_n} + 1)$.

Через $\mathcal{M}_G(\mathbb{F}_2)$ обозначим алгебру матриц $|G| \times |G|$, у которых строки и столбцы индексируются элементами группы $G$. Определим вложение $\mathbb{F}_2^G \hookrightarrow \mathcal{M}_G(\mathbb{F}_2)$, которое определим на базисе (элементах группы $G$):

$$g \in G \longrightarrow M_G(g) = \{a_{ij}\}_{i,j \in G} \in \mathcal{M}_G(\mathbb{F}_2), \text{ где } a_{ij} = \begin{cases} 1, & \text{если } i = gj, \\ 0, & \text{иначе.} \end{cases}$$

Далее будем отождествлять элемент $g \in \mathbb{F}_2^G$ с соответствующей матрицей $M_G(g)$ в тех случаях, где группа $G$ однозначно определяется из контекста.

Также нам понадобится ставить в соответствие элементам групповой алгебры $\mathbb{F}_2^G$ векторы длины $|G|$. Определим $V_G : \mathbb{F}_2^G \to \mathbb{F}_2^{|G|}$, где $V_G(g)$ — первый столбец матрицы $M_G(g)$. Заметим, что при таком определении выполнено свойство $V_G(gh) = M_G(g)V_G(h)$. Отметим, что в отличие от $M_G$, отображение $V_G$ — биекция.

Введём операцию $\bar{\cdot}: \mathbb{F}_2^G \to \mathbb{F}_2^G$, соответствующую транспонированию матрицы:

$$g = \sum_{\alpha \in G} c_\alpha \alpha \quad \longmapsto \quad \overline{g} = \sum_{\alpha \in G} c_\alpha \alpha^{-1}.$$

Тогда $M_G(\overline{g}) = (M_G(g))^T$.

Классический линейный код с проверочной матрицей $M_G(g)$ будем обозначать через $\mathcal{C}(G, g)$.

Если $G$ — группа и элементы $g, h \in \mathbb{F}_2^G$ коммутируют, то квантовый CSS код, задаваемый матрицами

$$\begin{aligned} H_X &= [M_G(g) \mid M_G(h)], \\ H_Z &= \left[M_G(\overline{h}) \mid M_G(\overline{g})\right] \end{aligned}$$

будем обозначать через $\mathcal{Q}(G, g, h)$.

**Пример 1.** $\mathcal{Q}\left(\langle x\rangle_L \times \langle y\rangle_L, 1 + x, 1 + y\right)$ представляет собой торический код [16, 17] с минимальным расстоянием $L$.

**Пример 2.** Семейство фрактальных кодов

$$\mathcal{Q}\left(\langle x\rangle_L \times \langle y\rangle_L \times \langle z\rangle_L, y + r(x), z + q(x)\right),$$

где $p$ и $q$ — некоторые многочлены, были исследованы в работе Йошиды [14, 15], однако для них неизвестно нижних оценок по порядку выше $L$, но несмотря на это есть гипотеза [14, 15], что расстояние этих кодов может быть выше $L^{3/2} = \Omega(\sqrt{n})$.

**Пример 3.** Кубический код Хааха [11] задаётся следующим образом:

$$\mathrm{Haah}(L) = \mathcal{Q}\left(\langle x\rangle_L \times \langle y\rangle_L \times \langle z\rangle_L, 1 + x + y + z, 1 + xy + xz + yz\right).$$

Торический код и фрактальные коды представляют собой два крайних случая: у торических кодов все логические ошибки являются цепями в графе Таннера, и для них легко находится точное значение минимального расстояния, а у фрактальных кодов логические ошибки минимального веса все имеют фрактальную структуру, и известные нижняя и верхняя оценка очень сильно расходятся.

В данной работе рассматривается промежуточный вариант кодов, у которых часть есть логические ошибки в виде цепей, а есть ошибки фрактального вида. Для этих кодов так же, как и для фрактальных кодов верхняя оценка на кодовое расстояние выше $\sqrt{n}$, а нижняя – ниже $\sqrt{n}$, где $n$ — длина кодового слова, но мы покажем, что для них возможно доказать нижнюю оценку, которая существенно выше, чем известная оценка для фрактальных кодов.

**Пример 4.** Семейство кодов, рассматриваемых в данной работе, а именно:

$$\mathrm{SF}(r,t) = \mathcal{Q}\left(\langle x\rangle_{L^2} \times \langle y\rangle_L, 1+x, y+r(x^L)\right), \qquad \text{где } L = 2^t.$$

Видно, что в данном примере первый полином, как в торическом коде, второй — как во фрактальном коде. Поэтому назовём эти коды *полуфрактальными*.

Отметим, что элементы группы $\langle x\rangle_{L^2} \times \langle y\rangle_L$ имеют вид $x^{i+Lj}y^k = x^i z^j y^k$, где $z = x^L$; $0 \leqslant i,j,k < L$. Поэтому для фиксированного многочлена $r$ семейство кодов $\mathrm{SF}(r,t)$ при $t \to \infty$, заданное многочленами $1+x$ и $y + r(z)$, является локальным при расположении кубитов на трёхмерной решётке $\mathbb{Z}_L^3$ с периодическими граничными условиями (узлы данной решётки можно расположить на трёхмерном торе).

Если $C$ — код (классический или квантовый), то через $d(C)$ будем обозначать его кодовое расстояние.

Посмотрим, как выглядят кодовые слова кода $\mathcal{Q}(G,g,h)$. Уравнение $H_X v = 0$ в терминах групповой алгебры можно переписать, как

$$gs + ht = 0, \qquad \text{где } v = \begin{bmatrix} V_G(s) \\ V_G(t) \end{bmatrix}. \tag{3}$$

Таким образом, каждой $Z$-ошибке $v$ можно поставить в соответствие пару $(s,t)$, удовлетворяющую соотношению (3). Аналогично, каждой $X$-ошибке $u$ можно поставить в соответствие пару $(v,w)$, удовлетворяющую соотношению $\overline{h}v + \overline{g}w = 0$.

Условие вырожденности $v \in \langle H_Z\rangle = \left\langle \left[M_G(\overline{h}) \mid M_G(\overline{g})\right]\right\rangle$ означает, что существует вектор $V_G(a) \in \mathbb{F}_2^{|G|}$ такой, что $v = V_G(a)^T H_Z$, значит

$$v^T = H_Z^T V_G(a) = \begin{bmatrix} M_G(\overline{h})^T V_G(a) \\ M_G(\overline{g})^T V_G(a) \end{bmatrix} = \begin{bmatrix} V_G(ha) \\ V_G(ga) \end{bmatrix},$$

а это значит, что вектору $v$ соответствует пара $(ha, ga)$. Таким образом, множеству вырожденных $Z$-ошибок $\langle H_Z\rangle$ соответствует множество пар $\left\{(ha,ga) \mid a \in \mathbb{F}_2^G\right\}$. Аналогично, множеству вырожденных $X$-ошибок соответствует множество пар: $\left\{(\overline{g}a, \overline{h}a) \mid a \in \mathbb{F}_2^G\right\}$.

В данной работе нас будет интересовать случай, когда группа $G$ абелева. В этом случае выполнено

$$\overline{gs} + \overline{h}\overline{t} = \overline{sg + th} = \overline{gs + ht},$$

поэтому отображение $(s, t) \mapsto (\overline{t}, \overline{s})$ задаёт изоморфизм между пространством $X$-ошибок и пространством $Z$-ошибок и сохраняет вес и вырожденность ошибки. Получим следующее утверждение.

**Утверждение 1.** *Пусть группа $G$ абелева. Тогда для квантового кода $\mathcal{Q}(G, g, h)$ минимальный вес невырожденной $X$-ошибки равен минимальному весу невырожденной $Z$-ошибки, а также* $\operatorname{rank} H_X = \operatorname{rank} H_Z$.

Данное утверждение позволяет несколько упростить вычисление размерности CSS кода, которое даётся формулой (2). Но более важным следствием данного утверждения является возможность в дальнейшем при получении оценок на кодовое расстояние изучать только вес невырожденных $Z$-ошибок. Поэтому далее под словом ошибка мы будем иметь ввиду $Z$-ошибка.

## 3. Формулировка основных результатов

Сформулируем утверждение[8], выражающее размерность квантовых кодов, заданных элементами факторкольца многочленов, через размерность факторкольца многочленов, как векторного пространства над $\mathbb{F}_2$.

**Утверждение 2.** *Пусть* $Q = \mathcal{Q}\left(\langle x_1\rangle_{a_1} \times \cdots \times \langle x_n\rangle_{a_n}, p, q\right)$. *Тогда*

$$\dim Q = 2 \dim \left(\mathbb{F}_2[x_1, ..., x_n]/\left(x_1^{a_1} + 1, ..., x_n^{a_n} + 1, p, q\right)\right).$$

При доказательстве основного результата важной частью доказательства является лемма о выравнивании ошибок, которая может представлять независимый интерес. Рассмотрим код $Q = \mathcal{Q}\left(\langle x\rangle_{ac} \times H, 1 + x, g\right)$, где $g \in \mathbb{F}_2^{G'}$, $G' = \langle x^c\rangle \times H$. Поскольку $x$ коммутирует со всеми элементами группы, в частности, с $G$, то квантовый код $Q$ задан корректно. В лемме доказано, что для каждой логической ошибки минимального веса существует эквивалентная ей ошибка минимального веса, но в некотором смысле выровненная по элементам подгруппы $G'$.

**Определение 4.** Ошибка $e_Z = [p, q]$ кода $Q$ называется *выровненной*, если $(1 + x)p \in \mathbb{F}_2^{G'}$ и $q \in \mathbb{F}_2^{G'}$.

[8] Аналогичное утверждение в значительно более общем виде может быть найдено в работе [12, Следствие 4.5].

Заметим, что выровненная ошибка кода $Q$ всегда имеет вид

$$e_Z = \Big[p_0 \sum_{j=0}^{c-1} x^j,\ q_0\Big], \quad \text{где } p_0, q_0 \in \mathbb{F}_2^{G'}.$$

**Лемма 1** (О выравнивании ошибок)**.** *Пусть $G = \langle x\rangle_\ell \times H$, $\ell = ac \geqslant 2$ и $g \in \mathbb{F}_2^{G'}$, где $G' = \langle x^c\rangle \times H \subseteq G$. Рассмотрим квантовый код $Q = \mathcal{Q}(G, 1+x, g)$. Для любой логической ошибки $e_Z$ среди эквивалентных ей ошибок минимального веса существует выровненная ошибка $e'_Z$.*

Одним из применений этой леммы может быть поиск простых нижних оценок для квантовых кодов, описываемых в этой лемме. В качестве одного из примеров применения этой леммы будет доказано следующее утверждение.

**Утверждение 3.** *Пусть $G = \langle x\rangle_{ac} \times H$, $ac \geqslant 2$, $G' = \langle x^c\rangle \times H \subset G$, $g \in \mathbb{F}_2^{G'}$ имеет чётный вес, и минимальное расстояние классического кода $\mathcal{C}(G', g)$ равно $d$.*

*Тогда $\mathcal{Q}(G, 1+x, g)$ — квантовый код с размерностью, отличной от 0 и минимальным расстоянием не меньше* $\min(d, c)$.

Основным результатом данной работы является нижняя оценка кодового расстояния для полуфрактальных кодов, сформулированная в следующей теореме.

**Теорема 1.** *Если $L = 2^t$, то минимальное расстояние полуфрактального кода*

$$\mathrm{SF}(1+x+x^2, t) = \mathcal{Q}\left(\langle x\rangle_{L^2} \times \langle y\rangle_L, 1+x, y+1+x^L+x^{2L}\right)$$

*ограничено снизу величиной $\Omega(L^\alpha)$, где $\alpha = \log_2(2(\sqrt{5}-1)) \approx 1.306$.*

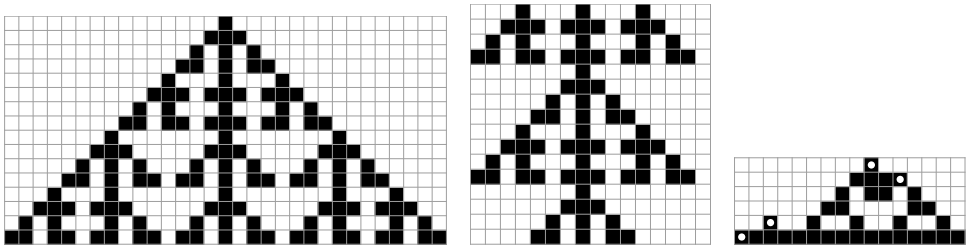
## 4. Доказательство

Для оценки кодового расстояния полуфрактальных кодов $\mathrm{SF}(r, t)$ нам понадобятся вспомогательные величины:

$$D_r(t) = \sum_{j=0}^{2^t-1} |r^j(x)|,$$

$$D_r^*(t) = \min_{\substack{p \in \mathbb{F}_2[x]/(x^{2^t}-1),\\ |p| \equiv 1 \mod 2}} \left| p(x) \sum_{j=0}^{2^t-1} (r(x)/y)^j \bmod (x^{2^t}-1, y^{2^t}-1) \right|,$$

$$\overline{D}_r(t) = \min_{h \in \mathbb{F}_2[x,y]} \left\{ |h(x,y)| \;\middle|\; h(x, r(x)) \equiv 1 + x + .. + x^{2^t-1} \mod x^{2^t}-1 \right\}.$$



(a) $D_{1+x+x^{-1}}(4) = 128$ (b) $D^*_{1+x+x^{-1}}(4) = 80$ (c) $\overline{D}_{1+x+x^{-1}}(4) = 4$

Рис. 1. Интерпретация величин $D$, $D^*$ и $\overline{D}$ в терминах клеточных автоматов.

Величины $D_r$, $D^*_r$ и $\overline{D}_r$ можно интерпретировать в терминах одномерного линейного клеточного автомата $A_r$ с функцией перехода, задаваемой многочленом $r(x)$. Мы не будем использовать эту интерпретацию в формальных рассуждениях, однако иногда удобно её иметь в виду, чтобы удобнее представлять себе объекты, о которых будет идти речь. На рисунке 4 проиллюстрирована такая интерпретация на примере многочлена $r(x) = 1 + x + x^{-1}$, который соответствует элементарному клеточному автомату Rule 150 [19]. Отметим, что с величины $D_r, D^*_r, \overline{D}_r$ для многочленов $1 + x + x^{-1}$ и $1 + x + x^2$ совпадают, но в терминах клеточных автоматов многочлен $1 + x + x^{-1}$ более естественный, поскольку соответствует симметричной окрестности радиуса 1.

Величину $D_r(t)$ можно понимать, как количество единиц в эволюции одномерного линейного клеточного автомата [18, 19], с функцией перехода, задаваемой многочленом $r(x)$, в течение $2^t$ тактов (рисунок 1(a)). Для линейных клеточных автоматов определено понятие предельного множества и есть алгоритм вычисления фрактальной размерности этого множества [20]. Если $a_r$ — фрактальная размерность, то $D_r(t) \asymp 2^{a_r t}$.

Величина $D^*_r(t)$ — минимальное число клеток в эволюции клеточного автомата $A_r$ в полосе ширины $2^t$ с периодическими граничными условиями, на протяжении $2^t$ тактов, начиная с конфигурации с нечётным числом клеток (рисунок 1(b)).

Чтобы интерпретировать величину $\overline{D}_r(t)$, нужно рассмотреть автомат $A_r$ с возможностью подавать на него управляющие сигналы. Один управляющий сигнал переключает состояние одной ячейки на противоположное. $\overline{D}_r(t)$ — минимальное число управляющих сигналов, которые необходимо подать, чтобы перевести состояние 00...00 в состояние 11...11 в полосе шириной $2^t$. На рисунке 1(c) кружками отмечены места, где подаются управляющие сигналы. Для полосы ширины 16 достаточно всего 4 сигналов, чтобы получить конфигурацию из всех единиц.

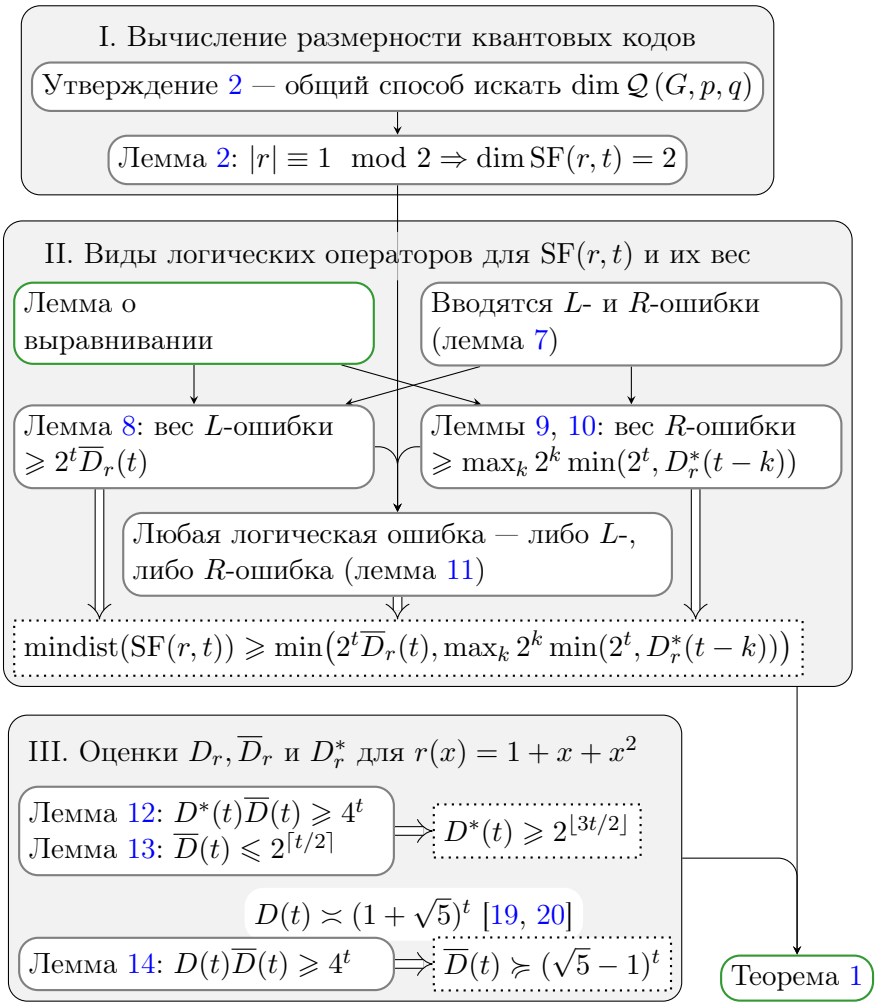



Рис. 2. Структура доказательства теоремы 1

Приведём общий план доказательства теоремы 1.

1) Утверждения о размерности квантовых кодов, заданных через групповые алгебры

   а) Общее утверждение 2, о кодовом расстоянии квантового кода $\mathcal{Q}(G, p, q)$ для коммутативной группы $G$.

   б) Для полуфрактального кода доказываем, что его размерность равна 2 (Лемма 2 с использованием утверждения 2). Как следствие получаем, что у такого кода есть 3 класса эквивалентности невырожденных ошибок.

2) Явный вид логических ошибок полуфрактальных кодов и нижние оценки их веса через величины $\overline{D}_r$ и $D^*_r$.

   а) Лемма о выравнивании, описывает класс, в котором содержатся ошибки минимального веса.

   б) Вводятся ошибки $L1, L2, L3$, и все ошибки делятся на $L$-ошибки и $R$-ошибки: в лемме 7, доказывается, что они являются логическими ошибками.

   в) В лемме 8 показано, что $L$-ошибки имеют вес не менее $2^t\overline{D}_r(t)$.

   г) В леммах 9, 10 показано, что вес любой $R$-ошибки не меньше $\max_k 2^k \min(2^t, D^*_r(t-k))$.

   д) В лемме 11 доказано, что $L1$, $L2$, $L3$ попарно неэквивалентны и невырожденные. Учитывая, что у кода всего 3 неэквивалентных невырожденных ошибки, получаем, что кроме $L1$, $L2$, $L3$ других невырожденных ошибок нет, значит любая ошибка – либо $L$-ошибка, либо $R$-ошибка.

   е) Из 2в, 2г, 2д получаем нижнюю оценку на кодовое расстояние

$$\operatorname{mindist}(\mathrm{SF}(r,t)) \geqslant \min\bigl(2^t\overline{D}_r(t), \max_k 2^k \min(2^t, D^*_r(t-k))\bigr).$$

3) Несколько лемм про соотношения между величинами $D_r$, $\overline{D}_r$ и $D^*_r$:

   а) $D_r(t)\overline{D}_r(t) \geqslant D^*_r(t)\overline{D}_r(t) \geqslant 4^t$ (леммы 12 и 14);

   б) $\overline{D}_{1+x+x^2}(t) \leqslant 2^{\lfloor t/2 \rfloor}$ (лемма 13);

   в) $D_{1+x+x^2}(t) \asymp (1+\sqrt{5})^t$ (известный результат [19, 20]).

4) Соединяя все оценки для $r(x) = 1 + x + x^2$, получаем оценку кодового расстояния в теореме 1.

Одна из причин, по которой было выбрано семейство кодов размерности 2 — то, что можно в явном виде выписать вид логических ошибок из каждого класса эквивалентности и изучать их по одиночке.

### 4.1. Леммы о размерности

*Доказательство утверждения 2.* Пусть $m = a_1 a_2 \cdots a_n$ — размерность групповой алгебры, изоморфной соответствующей факторалгебре многочленов

$$R = \mathbb{F}_2[x_1, ..., x_n]/(x_1^{a_1} + 1, ..., x_n^{a_n} + 1).$$

Длина кодового слова кода $Q$ равна $2m$. Посчитаем размерность с использованием формулы (2). Учитывая, что $\operatorname{rank} H_X = \operatorname{rank} H_Z$ по утверждению 1, имеем $\dim Q = 2m - 2\operatorname{rank} H_X$.

$$\operatorname{rank} H_X = \dim \langle H_X \rangle = \dim \{sp + tq \mid s, t \in R\} = \dim(p, q),$$

где под $(p, q)$ понимается идеал, порождённый элементами $p$ и $q$ в факторалгебре $R$. Тогда $\dim(g, h) = \dim R - \dim(R/(g, h)) = m - \dim(R/(g, h))$. Отсюда

$$\begin{aligned}\dim Q &= 2m - 2(m - \dim(R/(p, q))) = 2\dim(R/(p, q)) = \\ &= 2\dim\big(\mathbb{F}_2[x_1, ..., x_n]/(x_1^{a_1} + 1, ..., x_n^{a_n} + 1)/(p, q)\big) = \\ &= 2\dim\big(\mathbb{F}_2[x_1, ..., x_n]/(x_1^{a_1} + 1, ..., x_n^{a_n} + 1, p, q)\big).\end{aligned}$$

Утверждение доказано. □

С использованием доказанного утверждения посчитаем размерность полуфрактальных кодов.

**Лемма 2.** *Если вес многочлена $r$ нечётный, то* $\dim \mathrm{SF}(r, t) = 2$.

*Доказательство.* Из утверждения 2 имеем

$$\dim \mathrm{SF}(r, t) = 2\dim\big(\mathbb{F}_2[x, y]/(x^{L^2} + 1, y^L + 1, 1 + x, y + r(x^L))\big)$$

Вес $r$ нечётный, значит $r(x^L) \equiv r(1) = 1 \mod (1 + x)$, поэтому

$$(1+x, y+r(x^L), x^{L^2}+1, y^L+1) = (1+x, 1+y, x^{L^2}+1, y^L+1) = (1+x, 1+y).$$

Отсюда

$$\begin{aligned}\mathbb{F}_2[x, y]/(x^{L^2} + 1, y^L + 1, 1 + x, y + r(x^L)) = \mathbb{F}_2[x, y]/(1 + y)/(1 + x) \simeq \\ \simeq \mathbb{F}_2[x]/(1 + x) \simeq \mathbb{F}_2.\end{aligned}$$

Значит $\dim \mathrm{SF}(r, t) = 2\dim \mathbb{F}_2 = 2$. Лемма доказана. □

### 4.2. Лемма о выравнивании и её следствия

В этом разделе мы рассматриваем код $Q = \mathcal{Q}\left(\langle x \rangle_{ac} \times H, 1 + x, g\right)$ из леммы о выравнивании. Также используем обозначения: $G = \langle x \rangle_{ac} \times H$, $G' = \langle x^c \rangle \times H$.

**Идея доказательства.** Элементы $\mathbb{F}_2^G$ можно отождествить с подмножествами элементов группы $G$. Напомним, что ошибка $[p, q]$ является выровненной, если $(1 + x)p$ и $q$ лежат в $G'$.

Для слова $[p, q] \in (\mathbb{F}_2^G)^2$ элемент $(1 + x)p$ назовём *левым синдромом*, а элемент $gq$ назовём *правым синдромом*. Если $[p, q]$ — ошибка, то левый синдром должен совпасть с правым, давая в сумме 0, и в этом случае левый (и правый) синдромы назовём *полусиндромами*. Легко видеть, что ошибка является выровненной тогда и только тогда, когда соответствующий полусиндром лежит в $\mathbb{F}_2^{G'}$.

*Срезом* множества $X \subset G$ назовём множество вида $x^iG' \cap X$. Заметим, что срез правого синдрома, лежащий в $x^iG'$, порождается элементом $q \in x^iG'$.

Основная идея состоит в том, что срез $X = gq_X$ полусиндрома $S = gq$ можно «сдвигать», постепенно прибавляя к кодовому слову вырожденные кодовые слова вида $[gq_X, (1+x)q_X]$ таким образом, что компонента $X$ умножается на $x$. За $i$ шагов можно «сдвинуть» $X$ на $i$ позиций (при этом $X$ заменится на $x^iX$). И до тех пор, пока в $S \setminus X$ не пересекается с $x^iX$, с изменением $i$ кусок границы $X$ левой части кодового слова будет сдвигаться на $i$ позиций, поэтому вес левой части ошибки меняется линейно по $i$ (пусть он равен $w_0 + i\Delta w$), а вес правой части ошибки не меняется. В точке, где $x^iX$ перестаёт пересекаться с $S \setminus X$, вес левой части кодового слова по-прежнему равен $w_0 + i\Delta w$, а вес правой части может быть такой же, как у исходного слова, а может быть меньше.

Таким образом, у нас получается семейство эквивалентных кодовых слов, параметризованных целым числом $i$, изменяющимся в пределах некоторого отрезка. При этом вес кодового слова в пределах этого отрезка линейно зависит от $i$, а на краях значение меньше или равно этой линейной функции. Отсюда можно заключить, что всегда можно взять $i$ на одном из концов отрезка так, чтобы вес ошибки не увеличился.

То, что мы взяли $i$ из края отрезка, означает, что у нас два среза полусиндрома «склеились», и число таких компонент уменьшилось на 1. Повторяя эту процедуру можно добиться, чтобы был лишь один непустой срез в полусиндроме. На последнем шаге следует так сдвинуть срез, чтобы он попал в $G'$. Вес на каждом шаге не увеличивается, и в результате мы получим искомое выровненное кодовое слово с меньшим или равным весом.

Введём обозначение

$$I(a, b) = \sum_{j=\min(a,b)}^{\max(a,b)-1} x^j.$$

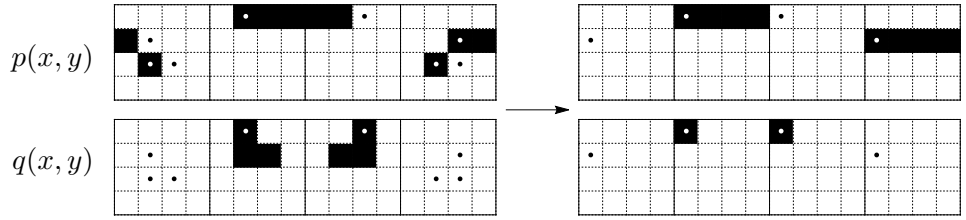



Рис. 3. Выравнивание логической ошибки $[p(x,y), q(x,y)]$ для кода $\mathrm{SF}(1+x+x^{-1}, 2)$, $L=4$. Каждая картинка изображает многочлен таким образом: чёрная клетка с координатами $(i,j)$ соответствует слагаемому $x^i y^j$. Точка с координатами $(i,j)$ соответствует слагаемому $x^i y^j$ в полусиндроме.

$I(a,b)$ обладает следующими свойствами:

$$I(l+a, r+a) = x^a I(l,r), \qquad I(l,r) + I(l',r') = I(l,l') + I(r,r'),$$
$$(1+x)I(l,r) = x^l + x^r, \qquad I(l,l) = 0.$$

Если $0 \leqslant a \leqslant b \leqslant \ell c$, то $|I(a,b)| = b - a$.

Пусть задан набор $v$ длины $c$, $v[i] \in \mathbb{Z}$. Введём оператор сдвига $S([p,q], v)$ для ошибки $[p,q]$. Пусть $q = \sum_{i=0}^{c-1} q_i x^i$, тогда

$$S([p,q], v) = [p,q] + [g, 1+x] \sum_{i=0}^{c-1} q_i I(i, v[i]).$$

**Лемма 3.** *Если $[p,q]$ — ошибка, то $S([p,q], v)$ — эквивалентная ей ошибка.*

*Доказательство.* Утверждение следует из того, что $[p,q] + S([p,q], v]) = [g, 1+x]a$ для $a = \sum_{i=0}^{c-1} q_i I(i, v[i]) \in \mathbb{F}_2^G$. □

Для удобства введём доопределённый вектор $v^*[i] := v[a] + cb$, если $i = a + cb$, $0 \leqslant a \leqslant c-1$. Следующая лемма утверждает, что в представлении $p$ в виде суммы «интервалов» $h_i I(l_i, r_i)$ после применения оператора $S$ концы интервалов сдвигаются под действием отображения $v^*$.

**Лемма 4.** *Пусть $[p,q]$ — ошибка и*

$$p = \sum_{i=1}^{N} h_i I(l_i, r_i), \ \textit{где } h_i \in H, \ 0 \leqslant l_i < r_i \leqslant \ell c. \tag{4}$$

*Тогда*

$$S([p,q],v) = \Big[\sum_{i=1}^{N} h_i I(v^*[l_i], v^*[r_i]), \sum_{i=0}^{c-1} q_i x^{v[i]}\Big].$$

*Доказательство.* Пусть $[p',q'] = S([p,q],v)$. Тогда

$$q' = q + (1+x)\sum_{i=0}^{c-1} q_i I(i, v[i]) = \sum_{i=0}^{c-1} q_i x^i + \sum_{i=0}^{c-1} q_i (x^i + x^{v[i]}) = \sum_{i=0}^{c-1} q_i x^{v[i]}.$$

Чтобы проверить компоненту $p'$, понадобится несколько обозначений. Представим $l_i$ и $r_i$ в виде

$$l_i = a_{2i-1} + cb_{2i-1}, \qquad r_i = a_{2i} + cb_{2i} \qquad \text{где } 0 \leqslant a_j \leqslant c-1,\ 0 \leqslant b_j \leqslant \ell.$$

Определим множества $M_j = \{i \mid a_i = j\}$. Пусть также $h'_{2i-1} = h'_{2i} = h_i$. Тогда

$$\begin{aligned}(1+x)p &= \sum_{i=1}^{N} h_i (1+x) I(l_i, r_i) = \\ &= \sum_{i=1}^{N} h_i (x^{l_i} + x^{r_i}) = \sum_{i=1}^{2N} h'_i x^{a_i} x^{cb_i} = \sum_{j=0}^{c-1} x^j \sum_{i \in M_j} h'_i x^{cb_i}.\end{aligned}$$

С другой стороны, поскольку $[p,q]$ — кодовое слово, имеем

$$(1+x)p = gq = \sum_{j=0}^{c-1} g q_j x^j.$$

Учитывая, что $gq_j \in \mathbb{F}_2^{G'}$ и $h'_i, x^{cb_i} \in \mathbb{F}_2^{G'}$, то $gq_j = \sum_{i \in M_j} h'_i x^{cb_i}$. Вычислим $\Delta p = p' + p$:

$$\begin{aligned}\Delta p &= \sum_{i=1}^{N} h_i \big(I(l_i, r_i) + I(v^*[l_i], v^*[r_i])\big) = \sum_{i=1}^{N} h_i \big(I(l_i, v^*[l_i]) + I(r_i, v^*[r_i])\big) = \\ &= \sum_{i=1}^{2N} h'_i\, I(a_i + cb_i, v[a_i] + cb_i) = \sum_{i=1}^{2N} h'_i x^{cb_i} I(a_i, v[a_i]) = \\ &= \sum_{j=0}^{c-1} I(j, v[j]) \sum_{i \in M_j^l} h'_i x^{cb_i} = \sum_{j=0}^{c-1} I(j, v[j]) q_j g.\end{aligned}$$

Лемма доказана. □

Покажем, что при условии $0 \leqslant v[i] \leqslant c$ отображение $v \mapsto v^*$ сохраняет монотонность.

**Лемма 5.** *Пусть*

$$0 = v[0] \leqslant v[1] \leqslant v[2] \leqslant \cdots \leqslant v[c-1] \leqslant c. \tag{5}$$

*Тогда $v^*$ монотонно на $\mathbb{Z}$.*

*Доказательство.* Пусть $a < b$. Рассмотрим 2 случая.

1) Если $a < ct \leqslant b$ для некоторого $t \in \mathbb{Z}$, то

$$v^*[a] \leqslant c(t-1) + c = ct \leqslant v^*[b].$$

2) Иначе $ct \leqslant a \leqslant b < c(t+1)$. Тогда

$$v^*[a] = ct + v[a - ct] \leqslant ct + v[b - ct] = v^*[b].$$

Итак, в обоих случаях $v^*[a] \leqslant v^*[b]$. Лемма доказана. □

*Доказательство леммы 1 о выравнивании ошибок.* Зафиксируем произвольную ошибку $w = [p, q]$. Сначала посмотрим, как действует оператор сдвига на её левую часть $p$.

Представим $p$ в виде суммы (4) таким образом, чтобы $\operatorname{supp} h_i I(l_i, r_i)$ не пересекались. Тогда $|p| = \sum_{i=1}^{N}(r_i - l_i)$.

Подберём такой вектор $v$, чтобы минимизировать величину

$$L(v) = \sum_{i=1}^{N}(v^*[r_i] - v^*[l_i])$$

при ограничении (5). Заметим, что $v^*$ линейно по $v$, а целевая функция линейна по $v^*$, а значит и по $v$.

Для начальных значений $v_0[i] = i$ по построению $L(v_0) = |p|$.

Множество, задаваемое неравенствами (5), является $(c-1)$-мерным симплексом с вершинами $(0, \underbrace{0, 0, ..., 0}_{k}, \underbrace{c, ..., c}_{c-k-1})$, $k = 0, ..., c-1$. Поэтому минимум линейной функции $L(x)$ при ограничении (5) достигается в некоторой вершине $v$ этого симплекса, причём $v[i] \in \{0, c\}$, значит $v[i]$ — целое и делится на $c$.

Положим $[p', q'] = S([p, q], v)$. Поскольку $v^*[l_i], v^*[r_i]$ делятся на $c$, то $I(v^*[l_i], v^*[r_i])$ делится на $I(0, c)$, а значит и $p'$ делится на $I(0, c)$; также поскольку $v[i]$ делится на $c$, то $x^{v[i]} \in G'$, значит $q' \in \mathbb{F}_2^{G'}$, поэтому ошибка $[p', q']$ является выровненной, а по лемме 3 она эквивалентна $[p, q]$.

Осталось оценить вес $[p', q']$. Поскольку (5) выполнено, то $v^*$ монотонно по лемме 5, значит $|I(v^*[l_i], v^*[r_i])| = v^*[r_i] - v^*[l_i]$. Отсюда, используя лемму 4, получим

$$|p'| = \Big|\sum_{i=1}^{N} h_i\, I(v^*[l_i], v^*[r_i])\Big| \leqslant \sum_{i=1}^{N} |h_i I(v^*[l_i], v^*[r_i])| =$$
$$= \sum_{i=1}^{N} (v^*[r_i] - v^*[l_i]) = L(v) \leqslant L(v_0) = |p|,$$
$$|q'| = \Big|\sum_{j=0}^{c-1} q_j x^{v[j]}\Big| \leqslant \sum_{j=0}^{c-1} |q_j| = |q|.$$

Отсюда $\big|[p', q']\big| \leqslant \big|[p, q]\big|$. Лемма доказана. □

**Лемма 6.** *Пусть $G = \langle x\rangle_\ell \times H$, $g \in \mathbb{F}_2^G$. Для того, чтобы размерность квантового кода $\mathcal{Q}(G, 1 + x, g)$ была отлична от 0, достаточно, чтобы вес $g$ был чётным.*

*Доказательство.* Пусть $g = \sum_{i=1}^{2k} g_i$, $g_i \in G'$. Тогда

$$\Big(\sum_{v\in G} v\Big) g = \sum_{i=1}^{2k} \sum_{v\in G} v g_i = \big\langle v' = v g_i^{-1}\big\rangle = \sum_{i=1}^{2k} \sum_{v'\in G} v' = 2k \sum_{v'\in G} v' = 0.$$

Аналогично, поскольку $1 + x$ – чётного веса, то

$$\Big(\sum_{v\in G} v\Big)(1 + x) = 0.$$

Значит матрицы $H_X = [1 + x, g]$ и $H_Z = [\overline{g}, 1 + x^{-1}]$ вырожденные и $\operatorname{rank}[\overline{g}, 1 + x^{-1}] = \operatorname{rank}[1 + x, g] \leqslant |G| - 1$, значит размерность кода $\geqslant 2|G| - 2(|G| - 1) = 2$. Лемма доказана. □

*Доказательство утверждения 3.* Из того, что вес $g$ чётный по лемме 6 получаем, что размерность кода $\mathcal{Q}(G, 1 + x, g)$ отлична от 0.

Рассмотрим ошибку минимального веса. По лемме 1 мы можем выбрать ошибку вида $e = \Big[p_0 \sum_{j=0}^{c-1} x^j,\ q_0\Big]$, где $p_0, q_0 \in \mathbb{F}_2^{G'}$. Рассмотрим 2 случая:

1) Если $p_0 = 0$, то $g q_0 = 0$, значит $q_0$ — кодовое слово классического кода, задаваемого элементом $G$, а по условию его расстояние $d$. Значит $|q_0| \geqslant d$.

2) Если $p_0 \neq 0$, тогда $|e| = |p_0|c + |q_0| \geqslant c$.

Лемма доказана. □

На самом деле, в случае $G' = H$, то есть, $\ell = 1$, код $\mathcal{Q}(G, 1+x, g)$ является [6] *гиперграфовым кодом-произведением* (анг. hypergraph product code) классических кодов $\mathcal{C}(\langle x \rangle_c, 1+x)$ и $\mathcal{C}(H, g)$, и оценка, даваемая утверждением 3, совпадает с оценкой из [6].

**Пример 5.** Положим $G' = H = \langle y \rangle$, $y^c = 1$, $g = 1 + y$. Получим торический код с матрицей $H_X = [1+x, 1+y]$ и кодовым расстоянием $c$. Легко видеть, что классический код, задаваемый элементом $1 + y$, является $[c, 1, c]$-кодом, где кодируемый бит просто повторяется $c$ раз. Даваемая леммой нижняя оценка кодового расстояния $c$ в данном случае является точной.

Чтобы получить коды, отличные от кодов из [6], нужно брать $\ell \geqslant 2$.

### 4.3. Виды логических ошибок $\mathrm{SF}(r,t)$ и их вес

Далее будем рассматривать полуфрактальные коды

$$\mathrm{SF}(r,t) = \mathcal{Q}\left(\langle x \rangle_{4^t} \times \langle y \rangle_{2^t}, 1+x, y + r(x^{2^t})\right),$$

в этом случае $H = \langle y \rangle_L$, $G = \langle x \rangle_{L^2} \times H$, $G' = \langle x^L \rangle \times H$, где $L = 2^t$.

Учитывая соотношения $x^{L^2} = y^L = 1$, будем использовать $1/x = x^{-1} = x^{L^2-1}$ и $1/y = y^{-1} = y^{L-1}$.

В лемме 2 мы показали, что размерность кода $\mathrm{SF}(r,t)$ равна 2. Таким образом, существует 4 логических ошибки, включая нулевую. Найдём 3 невырожденные ошибки. Введём следующие ошибки:

$$\Big[\sum_{j=0}^{L^2-1} x^j,\ \mathbf{0}\Big], \tag{$L1$}$$

$$\Big[\mathbf{0},\ \sum_{j=0}^{L-1} \big(r(x^L)/y\big)^j\Big], \tag{$L2$}$$

$$(L1) + (L2) = \Big[\sum_{j=0}^{L^2} x^j,\ \sum_{j=0}^{L-1} \big(r(x^L)/y\big)^j\Big] \tag{$L3$}.$$

Здесь ошибки заданы вектором из многочленов. Подразумевается, что каждому многочлену ставится в соответствие ошибка, в которой на позициях, соответствующих ненулевым коэффициентам многочлена стоит 1, на оставшихся позициях — 0.

**Лемма 7.** *$(L1)$, $(L2)$ и $(L3)$ являются ошибками для кода $\mathrm{SF}(r,t)$.*

*Доказательство.* Проверим, что ошибки $H_X L1 = H_X L2 = H_X L3 = 0$. Учитывая, что $s$ и $a$ — степени 2, получим

$$H_X L1 = (1+x)\sum_{i=0}^{L^2-1} x^j = 1 + x^{L^2} = 0.$$

$$H_X L2 = (r(x^L)+y)\sum_{j=0}^{L-1}\left(r(x^L)/y\right)^j =$$

$$= y(r(x^L)/y+1)\sum_{j=0}^{L-1}\left(r(x^L)/y\right)^j = y\left(\left(r(x^L)/y\right)^L + 1\right) = 0.$$

Здесь мы учли, что $y^L = 1$, $x^{L^2} = 1$ и

$$r^L(x^L) = r(x^{L^2}) = r(1) = |r| \mod 2 = 1,$$

поскольку в $r$ — нечётное число ненулевых слагаемых. □

Ошибки, эквивалентные $L1$ будем называть *левыми* или $L$*-ошибками*, а эквивалентные $L2$ или $L3$ — *правыми* или $R$*-ошибками*. Ниже, в лемме 11, мы докажем, что их классы эквивалентности различны, но удобнее сначала получить оценки снизу на веса $L$- и $R$-ошибок.

**Лемма 8.** *Вес любой $L$-ошибки не меньше $L\overline{D}_r(t)$.*

*Доказательство.* Возьмём $L$-ошибку минимального веса вида

$$e = [p(x^L, y)(1 + x + ... + x^{L-1}), q(x^L, y)]$$

(по лемме 1 такая существует) и покажем, что $|p| \geqslant \overline{D}_r(t)$.

Поскольку $e$ эквивалентна $L1$, то существует такой $f_0(x, y) \in \mathbb{F}_2[x, y]$, что

$$p(x^L, y)(1+x+...+x^{L-1}) + f_0(x, y)(y + r(x^L)) = \sum_{i=0}^{L-1} x^{iL}(1+x+...+x^{L-1}).$$

Здесь равенство в кольце $\mathbb{F}_2[x, y]/(x^{L^2} - 1, y^L - 1)$. Поскольку $r(x^L)^L = 1$, то мы можем подставить $y = r(x^L)$, тогда получим

$$p(x^L, r(x^L))(1 + x + ... + x^{L-1}) = \sum_{i=0}^{L-1} x^{iL}(1 + x + ... + x^{L-1}).$$

Отсюда, оставляя лишь коэффициенты при степенях $x^{iL}$ и делая замену $z = x^L$, получим

$$p(z, r(z)) = 1 + z + ... + z^{L-1},$$

значит $\overline{D}_r(t) \leqslant |p|$, отсюда $|w| \geqslant L|p| \geqslant L\overline{D}_r(t)$, что и требовалось. □

**Лемма 9.** *Для любой R-ошибки $(p, q)$ для всех $j = 0, ..., L-1$ вес $q_j$ нечётный.*

*Доказательство.* Для любой вырожденной ошибки $[p', q']$ вес $q'_j$ чётный, поскольку $q' = (1+x)s(x, y)$ для некоторого $s \in \mathbb{F}_2[x, y]/(y^L - 1, x^{L^2} - 1)$. Из любой $R$-ошибки прибавлением вырожденной ошибки можно получить либо $[p'', q''] = L2$, либо $[p'', q''] = L1 + L2$. В обоих случаях $q''_j(1) = r(1)^j = 1$, то есть $q''_j$ имеет нечётный вес. А значит и $q_j = q'_j + q''_j$ имеет нечётный вес, что и требовалось. □

**Следствие 1.** *Для любой R-ошибки $[p, q]$ для всех $j = 1, ..., L-1$ выполнено $q_j \neq 0$.*

**Лемма 10.** *Вес любой R-ошибки не меньше*

$$\max_{1 \leqslant k \leqslant t} 2^k \min(L, D^*_r(t-k)).$$

*Доказательство.* Рассмотрим $R$-ошибку

$$w = [p(x^L, y)(1 + x + ... + x^{L-1}), q(x^L, y)]$$

минимального веса, ($R$-ошибка такого вида существует по лемме 1).

Зафиксируем произвольное $k \in \overline{1, t}$ и разделим левую и правую части ошибки на $2^k$ горизонтальных полос высоты $2^u$, где $u = t - k$. Соответствующее представление в виде полиномов:

$$p(x) = \sum_{i=0}^{2^k-1} p_i(x, y), \qquad q(x) = \sum_{i=0}^{2^k-1} q_i(x, y),$$

где $p_i(x, y)$ и $q_i(x, y)$ содержат лишь степени $y$ от $i2^u + 1$ до $(i+1)2^u$.

Покажем, что в каждой полосе либо $|q_i| \geqslant D^*_r(u)$, либо $|p_i| \neq 0$. Предположим, что $|p_i| = 0$. Тогда синдром

$$s_i = q_i(x, y)(y + r(x)) + p_i(x, y)(1 + x) = q_i(x, y)(y + r(x))$$

может содержать лишь $y$ в степенях $y^{i2^u}$ и $y^{(i+1)2^u}$, чтобы сократиться с $s_{i+1}$ и $s_{i-1}$. Пусть

$$q_i(x, y) = y^{(i+1)2^u - 1} \sum_{j=0}^{2^u-1} q_i^j(x)/y^j.$$

Тогда при $0 < j < 2^k$ имеем $q_i^j(x) + r(x)q_i^{j-1} = 0$, отсюда $q_i^{j+1}(x) = q_i^j(x)r(x)$, значит

$$q_i(x, y) = y^{(i+1)2^u - 1} q_i^0(x) \sum_{j=0}^{2^u-1} (r(x)/y)^j,$$

Поскольку мы рассматриваем $R$-ошибку, то по лемме 9 вес всех $q_i^j$ нечётный, в частности, нечётный вес имеет $q_i^0$. Тогда

$$|q_i(x,y)| = \left|q_i^0(x)\sum_{j=0}^{2^u-1}\frac{r^j(x)}{y^j}\right| \geqslant \left|q_i^0(x)\sum_{j=0}^{2^u-1}\frac{r^j(x)}{y^j} \bmod (x^{2^u}+1)\right| \geqslant D_r^*(u).$$

Итак, поскольку $|q_i| \geqslant D_r^*(t-k)$, либо $|p_i| \neq 0$, то

$$L|p_i| + |q_i| \geqslant \min(L, D_r^*(t-k)).$$

Суммируя по всем $i = 0, ..., 2^k - 1$, получим $|w| \geqslant 2^k \min(L, D_r^*(t-k))$. Поскольку $k$ мы брали произвольным, то из этой оценки сразу следует утверждение леммы. □

**Лемма 11.** *Код* $\mathrm{SF}(r,t)$ *имеет размерность* 2, *и ошибки* $(L1)$, $(L2)$, $(L3)$ *являются различными невырожденными логическими ошибками.*

*Доказательство.* Для всех пар ошибок из $\{0, L1, L2, L3\}$ покажем, что они различны.

1) По лемме 9 у любой $R$-ошибки $[p,q]$ при каждой степени $y$ в $q$ многочлен от $x$ имеет нечётный вес. А для 0-ошибки и $(L1)$ правая часть нулевая, поэтому $L2, L3 \not\sim 0, L1$.

2) $L1 \not\sim 0$ следует из леммы 8, поскольку минимальный вес $L$-ошибки больше 0.

3) $L3 + L2 = L1$, значит $L2$ и $L3$ — также различны.

Лемма доказана. □

Из леммы 11 следует, что любая невырожденная ошибка является либо $L$-ошибкой, либо $R$-ошибкой. Поэтому объединяя леммы 8 и 10, получим

**Следствие 2.** *Если многочлен* $r \in \mathbb{F}_2[x]$ *имеет нечётный вес, то*

$$\operatorname{mindist}\mathrm{SF}(r,t) \geqslant \min\bigl(2^t\overline{D}_r(t), \max_{1\leqslant k\leqslant t} 2^k \min(2^t, D_r^*(t-k))\bigr).$$

### 4.4. Соотношения между $D$, $\overline{D}$ и $D^*$ и доказательство основной теоремы

**Лемма 12.** $D_r^*(t)\overline{D}_r(t) \geqslant 4^t$.

*Доказательство.* По определению $\overline{D}_r(t)$ существует многочлен $p(x,y)$ такой, что $|p| = \overline{D}_r(t)$ и $1 + x + ... + x^{2^t-1} = p(x, r(x))$ в факторкольце $\mathbb{F}_2[x]/(x^{2^t} - 1)$.

Обозначим $c_0(x,y) = \sum_{i=0}^{2^t-1}(r(x)/y)^j$. Далее все действия производятся в факторкольце $\mathbb{F}_2[x,y]/(x^{2^t} - 1, y^{2^t} - 1)$.

Рассмотрим многочлен $q \in \mathbb{F}_2[x]$ нечётного веса такой, что вес многочлена $w(x,y) = q(x)c_0(x,y)$ равен $D_r^*(t)$. Заметим, что если разложить многочлен $p(x,y)c_0(x,y)$ по степеням $y$, то при $y^0$ будет множитель $p(x, r(x)) = 1 + x + ... + x^{2^t-1}$. Поскольку $c_0(x,y)r(x)/y = c_0(x,y)$ в этом факторкольце, то при $y^{L-j}$ будет множитель $r^j(x)p(x, r(x)) = 1 + x + ... + x^{2^t-1}$, значит

$$p(x,y)c_0(x,y) = \sum_{i=0}^{2^t-1}(1 + x + ... + x^{2^t-1})r(x)^j/y^j = \sum_{i,j=0}^{2^t-1} x^i/y^j,$$

то есть $|pc_0| = 4^t$. Отсюда

$$p(x,y)w(x,y) = q(x)c_0(x,y)p(x,y) = q(x)\sum_{i,j=0}^{2^t-1} x^i y^j = |q|\sum_{i,j=0}^{2^t-1} x^i y^j =$$

$$= \sum_{i,j=0}^{2^t-1} x^i y^j,$$

поскольку вес $q$ нечётный. Теперь можем оценить вес $w$:

$$D_r^*(t)\overline{D}_r(t) = |w||p| \geqslant |pw| = 4^t,$$

что и требовалось. □

**Лемма 13.** $\overline{D}_{1+x+x^2}(t) \leqslant 2^{\lceil t/2 \rceil}$.

*Доказательство.* Определим последовательность многочленов $h_t(x,y)$ индуктивно:

$$h_0(x,y) = 1, \tag{6}$$

$$h_{t+1}(x,y) = \left(y^{4^t} + x^{3\cdot 4^t}\right) h_t(x,y). \tag{7}$$

Легко видеть, что $|h_{t+1}| \leqslant 2|h_t|$, значит $|h_t| \leqslant 2^t$. Проверим индукцией по $t$, что $h_t(x, 1 + x + x^2) = \sum_{i=0}^{4^t-1} x^i$.

**База индукции:** $t = 0$. $h_0(x, 1 + x + x^2) = 1 = \sum_{i=0}^{4^0-1} x^i$.

**Шаг индукции** $t \to t+1$. Поскольку всё происходит в поле характеристики 2, то $(1+x+x^2)^{4^t} = 1 + x^{4^t} + x^{2\cdot 4^t}$, значит

$$h_{t+1}(x, 1+x+x^2) = \left((1+x+x^2)^{4^t} + x^{3\cdot 4^t}\right) h_t(x, 1+x+x^2) =$$
$$= \left(1 + x^{4^t} + x^{2\cdot 4^t} + x^{3\cdot 4^t}\right) \sum_{i=0}^{4^t-1} x^i = \sum_{i=0}^{4^{t+1}-1} x^i.$$

Отсюда сразу следует, что $\overline{D}_{1+x+x^2}(2t) \leqslant |h_t| \leqslant 2^t$. Поскольку

$$\sum_{i=0}^{2^{2t+1}} x^i = (1+x^{4^t}) \sum_{i=0}^{2^{2t}} x^i = (1+x^{4^t}) h_t(x, 1),$$

то $\overline{D}_{1+x+x^2}(2t+1) \leqslant 2|h_t| \leqslant 2^{t+1}$. Лемма доказана. □

**Лемма 14.** $D_r(t)\overline{D}_r(t) \geqslant 4^t$.

*Доказательство.* В определении $D_r^*(t)$ подставим $p(x) = 1$, получим

$$D_r^*(t) \leqslant \left| \sum_{j=0}^{2^t-1} r^j(x)/y^j \bmod (x^{2^t}-1, y^{2^t}-1) \right| \leqslant \sum_{j=0}^{2^t-1} |r^j(x)| = D_r(t).$$

Используя лемму 12 получим $D_r(t)\overline{D}_r(t) \geqslant D_r^*(t)\overline{D}_r(t) \geqslant 4^t$. □

*Доказательство теоремы 1.* Известно [19, 20], что

$$D_{1+x+x^2}(t) \asymp (1+\sqrt{5})^t, \tag{8}$$

отсюда по лемме 14

$$\overline{D}_{1+x+x^2}(t) \succcurlyeq \left(\frac{4}{1+\sqrt{5}}\right)^t = \left(\sqrt{5}-1\right)^t. \tag{9}$$

По лемме 13 имеем

$$\overline{D}_{1+x+x^2}(t) \leqslant 2^{\lceil t/2 \rceil}, \tag{10}$$

отсюда по лемме 12 получаем

$$D^*_{1+x+x^2}(t) \geqslant 2^{\lfloor 3t/2 \rfloor}. \tag{11}$$

По лемме 8, учитывая (9), вес любой $L$-ошибки не меньше

$$2^t \overline{D}_{1+x+x^2}(t) \succcurlyeq 2^t(\sqrt{5}-1)^t = 2^{\alpha t}. \tag{12}$$

Учитывая (11) и подставляя $k = \lfloor t/3 \rfloor$ в лемму 10, получим, что вес любой $R$-ошибки не меньше

$$2^{\lfloor t/3 \rfloor} \min\left(2^t, D_r^*(\lceil 2t/3 \rceil)\right) \geqslant 2^{t/3-1} \min\left(2^t, 2^{\left\lfloor \frac{3}{2} \lceil \frac{2}{3} t \rceil \right\rfloor}\right) = 2^{\frac{4}{3}t-1} \succcurlyeq 2^{\alpha t}. \quad (13)$$

Последнее асимптотическое неравенство следует из числового неравенства $\alpha < 4/3$.

Учитывая (12) и (13), получаем, что вес любой логической ошибки по порядку не меньше $2^{\alpha t}$, что и требовалось. □

## 5. Заключение

В данной работе введён класс полуфрактальных квантовых кодов, для которых доказана нижняя оценка кодового расстояния. Мы здесь не приводим верхней оценки, но она выше $\sqrt{n}$, и есть основания полагать, что кодовое расстояние полуфрактальных кодов может оказаться выше, чем $\sqrt{n}$.

Для получения нижней оценки использовались простые соотношения между величинами $D$, $\overline{D}$ и $D^*$. Потенциально нижняя оценка может быть улучшена, если будут улучшены оценки на величины $\overline{D}(t)$ и $D^*(t)$.

Интерес представляет также рассмотрение других многочленов $r(x)$, отличных от $1 + x + x^2$.

Для рассмотрения более широких классов кодов необходимо обобщить лемму о выравнивании на случай кодов $\mathcal{Q}(G, p, q)$, когда многочлен $p$ отличен от $1 + x$.

Далее перечислим некоторые гипотезы и соображения, как можно улучшить полученную в статье оценку, а также некоторые гипотезы.

Один из простейших способов улучшить оценку — рассмотреть код $\mathrm{SF}'(r, t, L') = \mathcal{Q}\left(\langle x \rangle_{LL'} \times \langle y \rangle_L, 1 + x, y + r(x^{L'})\right)$, $L = 2^t$, на решётке в виде параллелепипеда $L' \times L \times L$ вместо кубической решётки. Все оценки будут очень похожими, а именно, вес $L$-ошибки $\geqslant L'\overline{D}_r(t)$, а вес $R$-ошибки $\geqslant \max_k 2^k \min(L', D_r^*(t-k))$. При фиксированной длине кодового слова $L'L^2$ изменяя соотношение между $L'$ и $L$ можно максимизировать нижнюю оценку кодового расстояния.

Другим естественным шагом является улучшение оценок на $D^*(t)$ и $\overline{D}(t)$. Приведём здесь несколько гипотез относительно поведения этих величин.

**Гипотеза 1.** $D_r^*(t) \asymp D_r(t)$.

Заметим, что для величины $D_r$ есть алгоритм вычисления асимптотики [20, Теорема 5.5], а для $D_r^*(t)$ асимптотика неизвестна даже для

конкретного случая $r(x) = 1 + x + x^2$. Соответствующая задача была явно сформулирована в статье [13], где исследовалось кодовое расстояние классических фрактальных кодов.

Отметим, что проведённые нами компьютерные эксперименты подтверждают правдоподобность данной гипотезы для многих многочленов $r(x)$ и параметра $t \leqslant 5$. Её также ещё можно переформулировать в терминах линейных клеточных автоматов.

**Гипотеза 2** (переформулировка гипотезы 1)**.** *Для любого двоичного линейного клеточного автомата с периодическими граничными условиями плотность узора, порождаемого эволюцией одноклеточной начальной конфигурации, минимальна по порядку.*

Следующая гипотеза является в некотором смысле индикатором перспективности рассмотрения полуфрактальных кодов, как кандидатов на преодоление барьера кодового расстояния $\sqrt{n}\operatorname{polylog} n$.

**Гипотеза 3.** $\overline{D}_{1+x+x^2}(t) = 2^{\lceil t/2 \rceil}$.

Или в более слабой форме:

**Гипотеза 4.** *Существует многочлен $r \in \mathbb{F}_2[x]$ нечётного веса и $\varepsilon > 0$ такие, что*

$$D_r(t)\overline{D}_r(t) \succcurlyeq 4^{t(1+\varepsilon)} \qquad \textit{при } t \to \infty.$$

Гипотеза 3 основана лишь на верхней оценке и результатах компьютерных экспериментов для $t \leqslant 4$. Эту гипотезу можно пытаться опровергнуть, найдя экспериментально контрпример для $t \geqslant 5$, либо, если не получится, искать подходы к ее доказательству. В случае, если гипотеза верна, будут уже веские основания полагать, что кодовое расстояние кода $\mathrm{SF}(1 + x + x^2, t)$ превосходит $n^\gamma$, где $n = 8^t$ — длина кодового слова, $\gamma > 1/2$. Если же гипотеза 4 неверна, то кодовое расстояние полуфрактальных кодов будет не выше $\sqrt{n}\operatorname{polylog} n$.

Задача проверки гипотез 3 и 4 представляется ключевой и самой сложной частью исследования, связанного с кодовым расстоянием полуфрактальных кодов.

## Список литературы


[1] P. Shor, “Scheme for reducing decoherence in quantum computer memory”, *Phys. Rev. A*, **52**:4 (1995), R2493–R2496.

[2] D. Gottesman, *Stabilizer Codes and Quantum Error Correction*, Ph.D. Thesis, California Institute of Technology, Pasadena, California, 2004 (Submitted May 21, 1997), 114 pp., arXiv: quant-ph/9705052.

[3] A. R. Calderbank, P. Shor, "Good quantum error-correcting codes exist", *Phys. Rev. A*, **54**:2 (1996), 1098–1105.

[4] A. M. Steane, "Error Correcting Codes in Quantum Theory", *Phys. Rev. Lett.*, **77**:5 (1996), 793–797.

[5] М. Нильсен, И. Чанг, *Квантовые вычисления и квантовая информация*, Мир, М., 2006, 824 с.

[6] J. Tillich, G. Zémor, "Quantum LDPC Codes With Positive Rate and Minimum Distance Proportional to the Square Root of the Blocklength", *IEEE Transactions on Information Theory*, **60**:2 (Feb 2014), 1193–1202.

[7] M. H. Freedman, D. A. Meyer, F. Luo, "$Z_2$-systolic freedom and quantum codes", *Mathematics of quantum computation*, Computational Mathematics, ed. R. K. Brylinski, G. Chen, Chapman & Hall/CRC, 2002, 287–320.

[8] Sh. Evra, T. Kaufman, G. Zémor, *Decodable quantum LDPC codes beyond the $\sqrt{n}$ distance barrier using high dimensional expanders*, 2020, arXiv: quant-ph/2004.07935.

[9] T. Kaufman, R. J. Tessler, *Quantum LDPC codes with $\Omega(\sqrt{n}\log^k n)$ distance, for any $k$*, 2020, arXiv: quant-ph/2008.09495.

[10] S. Bravyi, B. Terhal, "A no-go theorem for a two-dimensional self-correcting quantum memory based on stabilizer codes", *New Journal of Physics*, **11**:4 (Apr 2009), 043029.

[11] J. Haah, "Local stabilizer codes in three dimensions without string logical operators", *Phys. Rev. A*, **83**:4 (Apr 2011), 042330.

[12] J. Haah, "Commuting Pauli Hamiltonians as Maps between Free Modules", *Communications in Mathematical Physics*, **324**:2 (Oct 2013), 351–399.

[13] B. Yoshida, "Information storage capacity of discrete spin systems", *Annals of Physics*, **338** (Nov 2013), 134–166.

[14] B. Yoshida, "Exotic topological order in fractal spin liquids", *Phys. Rev. B*, **88**:12 (Sep 2013), 125122.

[15] B. Yoshida, *Classical and quantum fractal code*, `slides` (XVII Conference on Quantum Information Processing (QIP 14), Barcelona, Spain, Feb. 3–7, 2014).

[16] А. Ю. Китаев, "Квантовые вычисления: алгоритмы и исправление ошибок", *УМН*, **52**:6(318) (1997), 53–112; *Russian Math. Surveys*, **52**:6 (1997), 1191–1249.

[17] A. Yu. Kitaev, "Fault-tolerant quantum computation by anyons", *Annals of Physics*, **303**:1 (Jan 2003), 2–30.

[18] S. Amoroso, G. Cooper, "Tessellation structures for reproduction of arbitrary patterns", *Journal of Computer and System Sciences*, **5**:5 (Oct 1971), 455–464.

[19] S. Wolfram, "Statistical mechanics of cellular automata", *Rev. Mod. Phys.*, **55**:3 (Jul 1983), 601–644.

[20] S. J. Willson, "Computing fractal dimensions for additive cellular automata", *Physica D: Nonlinear Phenomena*, **24**:1 (1987), 190–206.

[21] A. A. Kovalev, L. P. Pryadko, "Quantum Kronecker sum-product low-density parity-check codes with finite rate", *Phys. Rev. A*, **88**:1 (Jul 2013), 012311.

**On the minimum distance in one class of quantum LDPC codes**
**Kalachev G.V., Panteleev P.A.**

We consider a family of quantum LDPC codes with weight-6 stabilizer generators and two logical qubits, where some logical operators have a fractal structure. These codes can be considered as local quantum codes on the $L \times L \times L$ cubic lattice with periodic boundary conditions. We prove that the minimum distance of codes from this family is bounded below by $\Omega(L^\alpha)$, where $\alpha = \log_2(2(\sqrt{5} - 1)) \approx 1.306$.

*Keywords:* quantum LDPC code, local quantum code, minimum distance, linear cellular automaton, fractal dimension.

# О порядках линейных над полем рациональных чисел автоматов

Муравьев Н.В.[1]

Рассматривается задача определения порядка инициального линейного над полем рациональных чисел автомата относительно операции суперпозиции. Выведена верхняя оценка на порядок автомата, зависящая от размерности.

**Ключевые слова:** линейные автоматы, порядок в полугруппе.

## 1. Введение

Если входной и выходной алфавиты инициального автомата совпадают, то число различных автоматов, получаемых суперпозицией исходного с самим собой, может быть как конечным, так и бесконечным. Задача определения порядка конечного инициального автомата относительно суперпозиции алгоритмически неразрешима в общем случае [1]. Но для некоторых классов автоматов удается найти алгоритмы определения порядка. Например, Алешин С.В. показал [3], что в группе одномерных (вход и выход - элементы поля) линейных автоматов над полем из двух элементов автомат имеет конечный порядок тогда и только тогда, когда его переходы безусловны. Этот результат обобщается на одномерные автоматы над любым полем.

Ранее автором была доказана верхняя граница на порядок линейного автомата над конечным полем [7], что позволило получить алгоритм решения задачи для автоматов произвольной размерности над конечными полями. В данной работе этот результат будет распространен на случай поля рациональных чисел. А именно, будет выведена верхняя оценка на порядок линейного над $\mathbb{Q}$ автомата, зависящая от его размерности.

Работа существенно использует известные результаты в теории линейных автоматов [2, 3, 4, 5, 6], в частности метод передаточных функций.

[1] *Муравьев Никита Валерьевич* — студент каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: ne-ki-tos@yandex.ru .

Muravev Nikita Valerevich — student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, The Department of Mathematical Theory of Intellectual Systems.

## 2. Основные определения и утверждения

**Определение 2.1.** Линейным автоматом над полем рациональных чисел $\mathbb{Q}$ называется инициальный абстрактный автомат $(\Sigma, Q, \Omega, \phi, \psi, q_0)$, чьи множество состояний $Q$, входной $\Sigma$ и выходной $\Omega$ алфавиты есть подмножества конечномерных векторных пространств над $\mathbb{Q}$, а канонические уравнения имеют следующий вид:

$$\begin{cases} q(t+1) = Aq(t) + Bx(t) \\ y(t) = Dq(t) + Lx(t) \\ q(0) = q_0, \end{cases}$$

где $q(t) \in Q, x(t) \in \Sigma, y(t) \in \Omega$; $A, B, D, L$ - линейные операторы между соответствующими пространствами.

Без ограничения общности везде далее считаем, что размерности линейных оболочек алфавитов и линейных оболочек множеств состояний совпадают с размерностями соответствующих векторных пространств, подмножествами которых они являются.

**Определение 2.2.** Размерностью линейного автомата будем называть размерность линейной оболочки его входного-выходного алфавита.

Будем обозначать поле частных кольца многочленов над $\mathbb{Q}$ от переменной $z$ как $Frac(\mathbb{Q}[z])$.

Следующая лемма есть обобщение известных результатов [2, 6, 7]. Мы приводим ее без доказательства.

**Лемма 1.** *Сопоставим каждому слову $x = x_0x_1x_2x_3... \in \Sigma^\infty$ формальный ряд*

$$x(z) = \sum_{v=0}^{\infty} x_v z^v,$$

*А каждому слову $y = y_0y_1y_2y_3... \in \Omega^\infty$ - формальный ряд*

$$y(z) = \sum_{v=0}^{\infty} y_v z^v.$$

*Тогда для любого n-мерного линейного автомата $G$ существуют передаточная функция $M_G(z) \in (Frac(\mathbb{Q}[z]))^{n\times n}$ и сдвиг $S_G(z) \in (Frac(\mathbb{Q}[z]))^n$, такие что для любых $x \in \Sigma^\infty, y \in \Omega^\infty$*

$$y = G(x) \Leftrightarrow y(z) = M_G(z)x(z) + S_G(z).$$

**Определение 2.3.** Порядком автомата будем называть порядок его автоматной функции относительно суперпозиции.

## 3. Основные результаты

Введем следующие обозначения:

$\phi(m)$ - функция Эйлера, считающая количество натуральных чисел меньших $m$, которые взаимнопросты с $m$;

НОК - наименьшее общее кратное;

$\psi(n) := \max\{m \in \mathbb{N} : \phi(m) \leq n\}$.

**Теорема 1.** *Если порядок $n$-мерного линейного над $\mathbb{Q}$ автомата конечен, то он не превосходит*

$$\max_{1 \leq v_1 < \ldots < v_n \leq \psi(n)} HOK(v_1, ..., v_n).$$

*Доказательство.* Пусть порядок $n$-мерного линейного над $\mathbb{Q}$ автомата $G$ конечен. Тогда конечен порядок его передаточной функции $M_G(z)$. По лемме 1 передаточная функция $M_G(z)$ есть линейный оператор над полем $Frac(\mathbb{Q}[z])$, а значит она есть линейный оператор и над полем $Frac(\mathbb{C}[z])$. Порядок оператора конечен, а значит конечны и порядки по умножению его собственных значений (ведь при возведении матрицы в степень собственные значения тоже возводятся в степень). Следовательно собственные значения, лежащие в алгебраическом замыкании поля $Frac(\mathbb{C}[z])$, имеют конечные порядки. Но тогда они либо нули, либо корни из единицы. Все корни из единицы в алгебраическом замыкании поля $Frac(\mathbb{C}[z])$ лежат в $\mathbb{C}$, так как $\mathbb{C} \subset Frac(\mathbb{C}[z])$ алгебраически замкнуто, и при его расширении новых корней добавиться не может. То есть коэффициенты характеристического многочлена с одной стороны лежат в $\mathbb{C}$ (так как получаются сложением и умножением собственных значений), а с другой стороны лежат в $Frac(\mathbb{Q}[z])$ (так как получаются сложением и умножением элементов матрицы передаточной функции). Значит коэффициенты принадлежат $\mathbb{C} \cap Frac(\mathbb{Q}[z]) = \mathbb{Q}$.

Получили, что коэффициенты характеристического многочлена передаточной функции есть константы из $\mathbb{Q}$, а корни этого многочлена есть комплексные корни из единицы и, возможно, ноль. Но минимальный многочлен над $\mathbb{Q}$ для корня из единицы $k$-й степени это круговой многочлен

$$\prod_{\substack{1 \leq m \leq k \\ \text{НОД}(m,k)=1}} (\lambda - e^{2i\pi m/k})$$

порядка $\phi(k)$, где НОД означает наибольший общий делитель. То есть, если собственное значение передаточной функции $n$-мерного автомата $G$ есть корень из единицы степени $k$, то $\phi(k) \leq n$. А значит

$$k \leq \psi(n) := \max\{m \in \mathbb{N} : \phi(m) \leq n\}.$$

Таким образом, получена верхняя оценка на порядок собственных значений передаточной функции автомата. Теперь рассмотрим Жорданову нормальную форму передаточной функции. Это блочно-диагональная матрица с клетками вида

$$\begin{pmatrix} \lambda & 1 & 0 & 0 & ... \\ 0 & \lambda & 1 & 0 & ... \\ ... & ... & ... & ... & ... \\ ... & ... & 0 & \lambda & 1 \\ ... & ... & 0 & 0 & \lambda \end{pmatrix}$$

на диагонали.

Если $\lambda = 0$, порядок клетки равен ее размерности. Если $\lambda$ есть корень $k$-й степени из единицы, порядок клетки конечен тогда и только тогда, когда ее размерность равна единице, и равен $k$. Порядок передаточной функции есть наименьшее общее кратное порядков клеток.

Из вышесказанного следует, что порядок передаточной функции не превосходит

$$\max_{1 \le v_1 < ... < v_n \le \psi(n)} \text{НОК}(v_1, ..., v_n).$$

Однако автомат определяется не только своей передаточной функцией, но и сдвигом. Сдвиг автомата $G^n$ имеет вид

$$(M_G^{n-1}(z) + ... + M_G(z) + I)S_G(z).$$

Если $k, l$ наименьшие натуральные числа, для которых $M_G^k(z) = M_G^l(z), \quad k < l,$ то либо $(M_G^{l-1}(z) + ... + M_G^k(z))S_G(z) = 0$ и порядок автомата совпадает с числом различных степеней его передаточной функции, либо $(M_G^{l-1}(z) + ... + M_G^k(z))S_G(z) \neq 0$ и порядок автомата бесконечен (так как $\mathbb{Q}$ есть поле характеристики 0).

То есть, если порядок $n$-мерного линейного над $\mathbb{Q}$ автомата конечен, то он не превосходит

$$\max_{1 \le v_1 < ... < v_n \le \psi(n)} \text{НОК}(v_1, ..., v_n).$$

□

Полученную оценку можно огрубить, чтобы получить более лаконичное неравенство. Известно [8], что $\forall n > 6 \ \phi(n) \ge \sqrt{n}$. Значит $\forall n > 6 \ \psi(n) \le n^2$. Тогда $\forall n > 6$

$$\max_{1 \le v_1 < ... < v_n \le \psi(n)} \text{НОК}(v_1, ..., v_n) \le \max_{1 \le v_1 < ... < v_n \le n^2} \text{НОК}(v_1, ..., v_n) \le$$

$$\le n^2 \cdot ... \cdot (n^2 - (n-1)) = \frac{(n^2)!}{(n^2 - n)!}.$$

То есть порядок $n$-мерного линейного над $\mathbb{Q}$ автомата либо бесконечен, либо не превышает $\frac{(n^2)!}{(n^2-n)!}$ при $n > 6$.

Для $n \leq 6$ верхнюю границу на порядок можно вычислить по теореме. Обозначив за $L(n)$ максимальный порядок $n$-мерных линейных над $\mathbb{Q}$ автоматов, получаем следующее следствие из теоремы 1:

**Следствие 1.1.**

$$L(1) \leq 2$$
$$L(2) \leq 30$$
$$L(3) \leq 60$$
$$L(4) \leq 4\,620$$
$$L(5) \leq 13\,860$$
$$L(6) \leq 2\,450\,448$$
$$L(n) \leq \frac{(n^2)!}{(n^2-n)!},\ n > 6.$$

Заметим, что обобщение полученных результатов на случай автоматов над полями вещественных и комплексных чисел невозможно. Уже среди двумерных автоматов с одним состоянием над полем вещественных чисел имеются автоматы сколь угодно больших порядков.

В самом деле, возьмем примитивный корень $n$-й степени из единицы $a + ib$ в поле комплексных чисел. Рассмотрим двумерный линейный над $\mathbb{R}$ автомат с одним состоянием:

$$\begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} = \begin{pmatrix} a & b \\ -b & a \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}.$$

Очевидно, порядок этого автомата равен порядку матрицы

$$\begin{pmatrix} a & b \\ -b & a \end{pmatrix}$$

по умножению.

Ее Жорданова нормальная форма имеет вид

$$\begin{pmatrix} a+ib & 0 \\ 0 & a-ib \end{pmatrix}.$$

То есть порядок матрицы равен $n$.

Таким образом показано, что для линейных автоматов над $\mathbb{R}$ не существует верхней границы на порядок, зависящей от размерности.

Автор выражает благодарность своему научному руководителю Бабину Д.Н. за помощь на всех этапах подготовки и написания данной работы.

## Список литературы


[1] P. Gillibert, “An automaton group with undecidable order and Engel problems”, *preprint, available online at arxiv.org/abs/1710.09733*, 2017.

[2] Кудрявцев В.Б., Алешин С.В., Подколзин А.С., *Введение в теорию автоматов*, "Наука", Москва, 1985.

[3] Алешин С.В., *Алгебраические системы автоматов.*, "МАКС Пресс", Москва, 2016.

[4] Бабин Д.Н., “Автоматы с линейными переходами”, *Интеллектуальные системы. Теория и приложения*, **23**:3 (2019), 87-95.

[5] Часовских А.А., “О полноте в классе линейных автоматов”, *Математические вопросы кибернетики*, 1995, № 3, 140–166.

[6] Ронжин Д.В., “Линейные автоматы над полем рациональных чисел”, *Интеллектуальные системы. Теория и приложения*, **21**:4 (2017), 144–155.

[7] Муравьев Н.В., “Разрешимость задачи определения порядка линейного автомата”, *Интеллектуальные системы. Теория и приложения*, **24**:2 (2020), 145-155.

[8] Kendall D.G., Osborn H.B., “Two Simple Lower Bounds for Euler’s Function”, *Texas Journal of Science*, **17** (1965).


### About orders of linear over rationals automata

**Muravev N.V.**



We consider the order problem with respect to the superposition operation for linear automata over rational numbers. An upper bound of automata orders is proved.



*Keywords:* linear automata, order in semigroup.

# Часть 4.
# Материалы семинаров кафедры МаТИС

# Доклады семинара «Теория автоматов»

С конца 2019 года на научном семинаре «Теория автоматов» под руководством академика Валерия Борисовича Кудрявцева состоялось 13 докладов.

4 декабря 2019 года

## Формальные модели безопасности и скрытые каналы

с.н.с. Галатенко А. В.

Наличие формальной модели и доказательства безопасности является обязательным требованием к системам с высоким классом защиты. В качестве классического примера верифицируемой безопасности можно привести модель Белла-Лападула. Теоремы, позволяющие свести проверку глобальной безопасности к проверке локальных условий, получили название теорем раскрутки. В 1976 году Харрисон, Руззо и Ульман доказали, что при достаточно естественных предположениях задача проверки безопасности становится алгоритмически неразрешимой. Как следствие, актуальной становится задача выявления моделей с разрешимой безопасностью. С другой стороны, при интеграции разнородных систем желательно уметь выражать одни модели в терминах других. В первой части доклада будет сделан обзор результатов о проблеме раскрутки и взаимной выразимости моделей, полученных совместно с учениками.

При переходе от реальной системы к математической абстракции неизбежно теряются какие-то свойства. В результате даже при наличии формального доказательства безопасности в реальной системе могут возникать информационные потоки, нарушающие безопасность. Такие потоки получили название скрытых каналов передачи информации. Во второй части доклада будет рассмотрен ряд примеров скрытых каналов, проанализирована возможность обеспечения заданной надежности и максимизации скорости передачи.

18 декабря 2019 года

## О методах и алгоритмах функционального построения подстановок элементарных абелевых групп и их комбинаторных, алгебраических и дифференциальных свойствах

в.н.с. Носов В. А.

В докладе рассматриваются результаты, связанные с функциональным построением подстановок элементарных абелевых групп с целью обеспечения у них определенных комбинаторных, алгебраических или дифференциальных свойств. К данному направлению относятся классические работы Шеннона К., Холла М., Клосса Б. М. и других авторов.

В докладе приводятся результаты, устанавливающие связи между различными разделами дискретной математики. Критерии существования важных для приложений комбинаторных, алгебраических и дифференциальных свойств конструируемых подстановок определяются критериями наличия у координатных функций табличных, аналитических (условия на коэффициенты Фурье) или алгебраических (условия на группы инерции используемых функций). Данные результаты представляют интерес для синтеза кодирующих автоматов.

19 февраля 2020 года

## Обзор математических результатов в геометрическом подходе к распознаванию визуальных образов

профессор Козлов В. Н.

В докладе излагается теорема о кодах изображений (коды двумерных, трехмерных и большей размерности изображений, эллипс-коды и пропорциональные изображения). Приводятся результаты, обеспечивающие оптимальное наложение изображений (минимизирующее «степень» их несовпадения) для преобразований параллельного переноса, изометрических, подобия и аффинных. Рассказывается об эскизах и остовах изображений, их оптимальном взаиморасположении. Суммируются результаты для восстановления трехмерных изображений по их двумерным проекциям. В заключении рассказывается о планируемых направлениях продолжений и обобщений подхода.

26 февраля 2020 года

## О свойствах логического вывода в пропозициональных исчислениях

доцент Боков Г. В.

В докладе будут изложены результаты исследования свойств логического вывода в пропозициональных исчислениях. При этом акцент будет сделан на функциональных, алгоритмических и сложностных аспектах логического вывода:

- *Функциональный аспект.* С функциональной точки зрения пропозициональные исчисления представляют собой множества формул, замкнутые относительно правил вывода. Структура таких замкнутых классов по вложенности задаёт решётку пропозициональных исчислений. Данное представление порождает корпус проблем, связанных с описанием свойств этой решётки и базисности её элементов.

- *Алгоритмический аспект.* Логический вывод в пропозициональных исчислениях представляет собой процедуру получения одних формул через другие с помощью правил вывода. Существуют неразрешимые исчисления, для которых данная процедура не может быть выполнена алгоритмически. В докладе будет рассказано, при каких необходимых и достаточных условиях это возможно.

- *Сложностной аспект.* В последние десятилетия процедуры решения задач, извлекаемые из логического вывода в классических исчислениях, повсеместно используются во многих компьютерных системах. При этом первостепенное значение приобретает простота данных процедур. С этой точки зрения представляется интересным описание свойств минимального логического вывода.

4 марта 2020 года

## О применении алгоритмов обработки больших данных в задаче прогнозирования риска неблагоприятного клинического исхода

профессор Кудрявцев В. Б., профессор Рыжов А. П.,
доцент Строгалов А. С., Журавлев А. А., Шергин И. А.

В современных высокотехнологичных клиниках, благодаря внедрению медицинских информационных систем (МИС) фиксируется и накапливается большой объем данных о каждом пациенте (демографические, клинические данные, результаты лабораторных и инструментальных методов диагностики, характер оперативного лечения и пр.). Часть данных о пациенте могут быть представлены в динамике. Учитывая разнородность информации, ее нечеткость и, порой, неполноту, необходимо исследование возможности прогноза риска неблагоприятного клинического исхода на основе имеющихся данных и определение качества такого прогноза.

11 марта 2020 года

## Автоматы и вычисления в лабиринтах

н.с. Волков Н. Ю.

Автоматы и вычисления в лабиринтах. Аннотация: В докладе рассматриваются два подхода к описанию автоматных систем в лабиринтах (независимые системы, коллективы автоматов и автоматы с красками): через их траектории и через вычисляемые ими функции (или последовательности). Вводятся понятие лабиринтного агента (обобщающее автоматные системы в лабиринте). Определяются операции над агентами: сумма и обратная ей операция проецирования. Аналогичные операции определяются над автоматами в лабиринтах. Устанавливается взаимосвязь этих операций. Рассматриваются случаи, когда применение этих операций позволяет решить задачу преследования. Так же определяется понятие вычисления целочисленных функций автоматными системами. Далее излагается подход, при котором коллективы автоматов характеризуются функциями, которые они вычисляют. Приводятся результаты, полученные автором совместно с В.В.Ушаковой и гипотезы о классификации вычислимых функций через число автоматов, необходимое для их вычисления. Приводятся приложения к задаче преследования.

14 октября 2020 года

## Дискуссия на тему «Искусственный интеллект — проблемы и перспективы»

Тематика дискуссии восходит к классическим работам А.Тьюринга "Может ли машина мыслить" и Дж. фон Неймана "Вычислительная машина и мозг", которые возникли на заре становления кибернетики как науки. С тех пор дискуссии на тему "Может ли машина мыслить" то возникали, то затухали и в понимании этого вопроса особой ясности они не вносили. В последние годы, в связи с мощным развитием технологической базы компьютеростроения, тематика стала вновь актуальной, но вместе с научной базой, наработанной в теории искусственного интеллекта и прикладных программ в этой области, возникло множество работ спекулятивного типа, готовых объявить любое устройство со встроенными в его систему управления тривиальными алгоритмическими добавками "системой искусственного интеллекта". На наш взгляд настало время отделить "зерна от плевел" и обсудить эту тематику с разных точек зрения. Важно отметить существенные пересечения по данной тематике в математике, информатике и кибернетике.

21 октября 2020 года

## Коды изображений, инвариантные к аффинным преобразованиям

профессор Кудрявцев В. Б., профессор Козлов В. Н.

Изображения трактуются как конечные (непустые) множества точек в евклидовых пространствах разной размерности. Чаще рассматриваются двумерные изображения, хотя возможно рассмотрение и трехмерных (объемных изображений), и четырехмерных (объемные изображения в динамике). Код двумерного изображения можно представлять как совокупность всех чисел, получаемых отношением площадей треугольников с вершинами в точках изображения. Числам приписаны индексы, которыми помечены точки изображения. Такой код, как доказано, определяет изображение с точностью до аффинных преобразований. Позже возник вариант кода, в котором числа получаются не отношением площадей

треугольников, а отношением площадей эллипсов наименьшей площади, включающих эти треугольники. Такой код тоже определяет изображение с точностью до аффинных преобразований. Наконец, третья вариация возникает из доказанного утверждения о том, если для двух изображений, в рамках некоторой биекции на точках этих изображений, отношения площадей сопоставляемых треугольников с вершинами в точках изображений есть константа, то эти изображения аффинно эквивалентны. Есть более частные виды кодов, определяющие изображения с точностью до изометрических преобразований, и до преобразований подобия.

21 октября 2020 года

## О существовании изображения с заданным кодом

с.н.с. Алексеев Д. В.

Вводится кодирующая функция, инвариантная относительно аффинных отображений, и исследуются ее свойства. Найдены необходимые и достаточные условия того, что данный набор чисел является кодом невырожденного изображения.

28 октября 2020 года

## Введение в теорию автоматов

профессор Подколзин А. С.

В докладе представлен краткий обзор результатов по теории автоматов, полученный за длительный период группой исследователей под руководством академика В.Б. Кудрявцева. Теория автоматов представляет исключительный интерес как дискретный аналог классических моделей, описывающих непрерывные процессы. В частности, она находит широкое применение в проектировании современных вычислительных средств и разработке программного обеспечения.

28 октября 2020 года

## О классах автоматов с операцией суперпозиции, расширяющихся до максимальных

профессор Бабин Д. Н.

Решение задачи выразимости автоматов с операцией суперпозиции наталкивается на существенные трудности. Как показал Валерий Борисович Кудрявцев, предполных классов автоматов – континуум. Более того, как показал Кратко М.И., сама задача выразимости в самой общей постановке являтеся алгоритмически неразрашимой. Известно, что система бесконечно порожденная. Оставался открытым вопрос о расширяемости замкнутых классов до предполных.

Автор показал, что для автоматов с суперпозициями есть классы, не расширяющиеся до предполных. В докладе исследованы классы автоматов, расширяющиеся до предполных. Таких классов — континуум. В их числе оказались все конечнопорожденные классы автоматов.

11 ноября 2020 года

## Введение в устройство мозга и естественных нейронных сетей[1]

м.н.с. Ивашкина О. И.
(Институт перспективных исследований мозга МГУ)

Доклад будет посвящен наиболее важным концепциям и принципам современной нейробиологии. Я расскажу про основную функциональную единицу мозга – нейрон, о принципах работы нейронов на разных уровнях: электрическом, биохимическом и геномном. Также мы обсудим закономерности, по которым из отдельных клеток формируются нейронные сети и структуры мозга, а также коснемся эволюции и развития мозга. Особое внимание я уделю фундаментальному свойству мозга: пластичности в нейронных сетях.

[1] Видеозапись семинара доступна по ссылке
`https://drive.google.com/file/d/1JeuCXR4p0rKI-4-lM5fjNqQxehThJHHh`

25 ноября и 2 декабря 2020 года

## Введение в когнитивные нейросети и естественный интеллект[2]

академик Анохин К. В.
(Институт перспективных исследований мозга МГУ)

На протяжении миллионов лет некоторые природные системы развили в себе свойство интеллекта – способность вырабатывать и успешно достигать множественные цели в широком диапазоне условий окружающей среды. Все такие системы обладают глубокой когнитивной сетью, вершины и ребра которой кодируют значимые для агента соотношения с его средой. В докладе будут разобраны примеры интеллектуального поведения таких естественных систем и принципы организации стоящих на этим когнитивных нейросетей. Особое внимание будет уделено свойствам и структурам, отсутствующим пока в системах искусственного интеллекта и искусственных нейросетях.

[2]Видеозаписи семинаров доступны по ссылкам
https://drive.google.com/file/d/1yPO5DVDPvMRxsabhp-p_lQA7-bPrh83W и
https://drive.google.com/file/d/1DVnWaA3cyeqLe_22AHBmtjOH_zq13e_L

# Доклады семинара «Вопросы сложности алгоритмов поиска»

В осеннем семестре 2020 – 2021 учебного года на научном семинаре «Вопросы сложности алгоритмов поиска» под руководством профессора Эльяра Эльдаровича Гасанова состоялось 11 докладов.

9 сентября 2020 года

## О верхней оценке сложности расшифровки функций фиксированного веса запросами на сравнение

асп. Быстрыгова А.В.

Доклад посвящен задаче точной расшифровки функций фиксированного веса запросами на сравнение: рассматривается класс функций, у которого в векторе значений ровно $k$ единиц, каждый запрос, используемый при расшифровке, представляет собой пару наборов, ответ на запрос — знак разности значений функций на этих наборах, цель — восстановить вектор значений функции при помощи как можно меньшего числа запросов. В докладе представлен алгоритм нахождения единиц функции, а также приводятся точные оценки сложности расшифровки для $k = 1, 2, 3$.

16 сентября 2020 года

## Поиск ближайшего соседа на прямой с помощью клеточного автомата с локаторами

инженер-разработчик НКБ «НИР» Васильев Д.И.

В докладе рассматривается применение модели клеточного автомата с локаторами к задаче поиска ближайшего соседа на прямой.

Модель клеточного автомата с локаторами подразумевает возможность каждой ячейки автомата передавать через эфир сигнал на сколь угодно большие расстояния. В докладе демонстрируется, что эта возможность позволяет уменьшить сложность рассматриваемой задачи с линейной до логарифмической по сравнению с классической моделью клеточного автомата.

23 сентября 2020 года

## Семантический анализ Правил дорожного движения

инженер-разработчик НКБ «НИР» Менькин М. И.

Доклад посвящён семантическому анализу текста юридического документа на примере Постановления Правительства "О правилах дорожного движения". Вводится модель семантики правил дорожного движения, а именно теоретико-графовая модель дорожной ситуации, являющаяся основой для моделирования правил движения. Приводятся синтаксические шаблоны текста документа для манёвра "Уступить дорогу". Также определяются шаблоны правил как результат отображения отдельных правил из текста в модель дорожной ситуации.

30 сентября 2020 года

## Верхние оценки энергопотребления в классе объёмных схем

инженер-разработчик ООО «СТЦ» Ефимов А. А.

Ещё в середине XX века в связи с интенсивным развитием вычислительной техники возникла задача синтеза схем, вычисляющих булевы функции и операторы. Одной из основных и наиболее подробно исследованных моделей схем является схема из функциональных элементов (СФЭ). В качестве характеристики оптимальности СФЭ можно рассматривать потенциал — мера мощности, равная количеству элементов схемы, выдающих единицу на данном входном наборе.

При этом часто рассматривались схемы, в которых не учитывались вполне естественные ограничения на размещение элементов схемы в плоскости или пространстве, способы их соединения, разводка проводов и т.п. В действительности, любая схема состоит из отдельных элементарных частей (функциональных элементов), которые имеют определенную длину, ширину и соединяются проводниками, размеры которых следует учитывать.

Также одной из моделей схем, учитывающих данные ограничения, являются плоские схемы, которые были введены С.С. Кравцовым в 1967 году. В данном докладе рассматриваются объёмные схемы, являющиеся обобщением плоских схем в пространстве. Была получена верхняя

оценка потенциала. Отдельно был рассмотрен класс $T_1$ схем, где длина дерева выходов минимальна. В данном классе получены верхние и нижние оценки, совпадающие по порядку.

7 октября 2020 года

## Нижняя оценка сложности задачи поиска ближайшего соседа, с применением модели клеточного автомата с локаторами

инженер-разработчик НКБ «НИР» Васильев Д.И.

Рассматривается применение модели клеточного автомата с локаторами к задаче поиска ближайшего соседа на прямой.

Модель клеточного автомата с локаторами подразумевает возможность каждой ячейки автомата передавать через эфир сигнал на сколь угодно большие расстояния. Показано, что эта возможность не позволяет уменьшить сложность рассматриваемой задачи до менее чем логарифмической.

14 октября 2020 года

## Теорема о персистентных гомологиях графов внимания модели машинного обучения BERT

инженер-разработчик Московского исследовательского центра Хуавэй Кушнарева Л. П.

В данный момент в сфере обработки текстов на естественном языке широко используются модели машинного обучения, основанные на механизме «внимания». Этот механизм помогает модели учитывать влияние контекста на семантику слов в тексте, что является важным условием для решения задач машинного перевода, детекции предложений, написанных со смысловыми ошибками, составления ответов на вопросы, сформулированные на естественном языке и т.п. Когда некоторое представление конкретного текста подается на вход такой модели, она строит взвешенные направленные графы, вершинами которых являются «токены» (слова/части слов и знаки препинания, из которых составлен текст). Вес каждого ребра (A, B) между двумя токенами A и B в таком графе представляет собой некую числовую оценку того, насколько смысловое значение токена A «влияет» на семантику значения токена B в данном контексте. Эти графы и называются графами «внимания»

(attention graphs). В связи с этим возникает естественный интерес к построению математических моделей этих графов, исследованию вопроса о том, какие их свойства можно выделить в рамках данных моделей, чему и посвящено данное исследование.

В данном докладе веса графов внимания смоделированы случайными величинами, и, соответственно, показана эмпирическая оценка параметров порождающего их вероятностного распределения. С помощью простых методов теории случайных графов дана асимптотическая оценка вероятности появления циклов и путей длины $k$ в таких графах. Также дано краткое введение в персистентные гомологии — топологический инвариант, который широко используется в современном анализе данных как инструмент для выделения свойств графов и комплексов, построенных на основе данных вероятностной природы. Показано, как из оценки количества путей легко следует асимптотическая оценка размерности групп персистентных гомологий этих графов, как еще одной из их описательных характеристик.

11 ноября 2020 года

## Расшифровка функций ограниченного веса запросами на сравнение

асп. Быстрыгова А.В.

В докладе рассматривается задача точной расшифровки функций ограниченного веса запросами на сравнение. Класс функций ограниченного веса $F(n, k, i)$ — множество булевых функций арности $n$, у которых число единиц в векторе значений ограничено снизу числом $i$, а сверху — числом $k$. В докладе приводится точная оценка сложности расшифровки класса $F(n, k, 0)$, а также нижняя оценка сложности расшифровки класса $F(n, k, k)$. Обе эти оценки позволяют получить верхнюю и нижнюю оценку для класса $F(n, k, i)$, совпадающие по порядку — $2^n$.

18 ноября 2020 года

## Верхняя оценка числа состояний клеточного автомата, реализующего двунаправленное движение на луче с половинной скоростью движения вперёд

соискатель Кузнецова Е.В.

В докладе рассматривается движение точки на экране, который реализован, как клеточный автомат на бесконечной в правую сторону полосе шириной в одну клетку. Законом движения назовём последовательность, состоящую из символов f, s, b (f-forward, s-stop, b-back), кодирующих перемещение точки в каждый момент времени. Если в момент времени t точка сместилась на одну клетку вправо, то t-ый член последовательности примет значение f, если влево, то b, если никуда не сместилась – s.

Изучается класс законов движения этого автомата, для которых движение вперёд возможно со скоростью, не большей, чем 1/2. При движении вперёд на одну клетку будет хотя бы одна пауза, то есть в законе движения перед f обязательно стоит s. Также возможно движение назад со скоростью 1.

Построен клеточный автомат с пятью состояниями, реализующий законы движения из рассматриваемого класса.

25 ноября 2020 года

## Нижняя оценка числа состояний клеточного автомата, реализующего двунаправленное движение на луче с половинной скоростью движения вперёд

соискатель Кузнецова Е.В.

В докладе рассматривается движение точки на экране, который реализован, как клеточный автомат на бесконечной в правую сторону полосе шириной в одну клетку. Законом движения назовём последовательность, состоящую из символов f, s, b (f-forward, s-stop, b-back), кодирующих перемещение точки в каждый момент времени. Если в момент времени t точка сместилась на одну клетку вправо, то t-ый член последовательности примет значение f, если влево, то b, если никуда не сместилась – s.

Изучается класс законов движения этого автомата, для которых движение вперёд возможно со скоростью, не большей, чем 1/2. При движении вперёд на одну клетку будет хотя бы одна пауза, то есть в законе движения перед f обязательно стоит s. Также возможно движение назад со скоростью 1.

Доказано, что невозможно построить клеточный автомат с количеством состояний, меньшим пяти, реализующий законы движения из рассматриваемого класса.

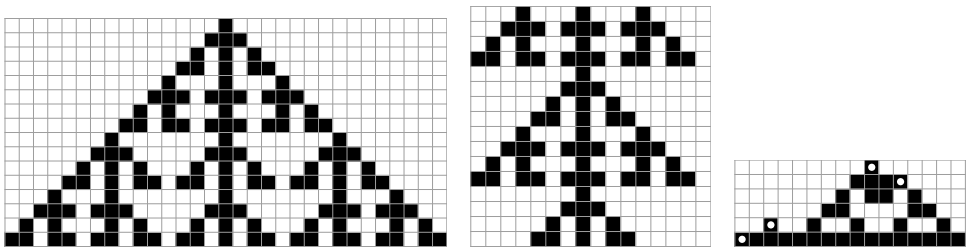
2 декабря 2020 года

## О характеристиках линейных клеточных автоматов, связанных с оценками минимального расстояния фрактальных квантовых кодов

к.ф.-м.н. Калачев Г.В.

Рассматриваются линейные клеточные автоматы (ЛКА) с двумя состояниями. Локальную функцию перехода ЛКА в $k$-мерном пространстве можно задать многочленом от $k$ переменных над полем из двух элементов $\mathbb{F}_2$. Конфигурацию клеточного автомата также можно задать многочленом, и тогда глобальная функция перехода соответствует умножению на многочлен локальной функции переходов.

Для линейного клеточного автомата $A_r$, задаваемого многочленом $r(x)$, рассматриваются 3 характеристики, связанные со скоростью роста числа единиц в конфигурации: $D_r(t)$, $D_r^*(t)$ и $\overline{D}_r(t)$. Ниже приведены определения этих величин в терминах многочленов.



(a) $D_{1+x+x^{-1}}(4) = 128$ (b) $D^*_{1+x+x^{-1}}(4) = 80$ (c) $\overline{D}_{1+x+x^{-1}}(4) = 4$

Рис. 1. Интерпретация величин $D$, $D^*$ и $\overline{D}$ в терминах клеточных автоматов.

1) $D_r(t) = \sum_{j=0}^{2^t-1} |r^j(t)|$ — число единиц в эволюции клеточного автомата на протяжении времени $2^t$, начиная с одной единичной клетки (рис. 1(a)).

2) $D_r^*(t) = \min\limits_{\substack{p\in\mathbb{F}_2[x]/(x^{2^t}-1),\\ |p|\equiv 1 \mod 2}} \left|p(x)\sum_{j=0}^{2^t-1}(r(x)/y)^j \bmod (x^{2^t}-1, y^{2^t}-1)\right|$ — минимальное число клеток в эволюции клеточного автомата $A_r$ в полосе ширины $2^t$ с периодическими граничными условиями, на протяжении $2^t$ тактов, начиная с конфигурации с нечётным числом единиц.

3) $\overline{D}_r(t) = \min\limits_{h\in\mathbb{F}_2[x,y]} \left\{|h(x,y)| \;\middle|\; h(x,r(x)) \equiv 1+x+..+x^{2^t-1} \mod x^{2^t}-1\right\}$. Чтобы интерпретировать величину $\overline{D}_r(t)$, нужно рассмотреть автомат $A_r$ с возможностью подавать на него управляющие сигналы. Один управляющий сигнал переключает состояние одной ячейки на противоположное. $\overline{D}_r(t)$ — минимальное число управляющих сигналов, которые необходимо подать, чтобы перевести состояние 00...00 в состояние 11...11 в полосе шириной $2^t$. На рисунке 1(c) кружками отмечены места, где подаются управляющие сигналы. Для полосы ширины 16 достаточно 4 сигналов, чтобы получить конфигурацию из всех единиц.

Будут показаны связи между этими величинами и доказаны оценки для случая $r(x) = 1+x+x^2$, а также приведены некоторые гипотезы. В конце доклада будет рассказано о связи этих величин с квантовыми кодами.

9 декабря 2020 года

## Нижние оценки энергопотребления объемных схем

инженер-разработчик ООО «СТЦ» Ефимов А. А.

Ещё в середине XX века в связи с интенсивным развитием вычислительной техники возникла задача синтеза схем, вычисляющих булевы функции и операторы. Одной из основных и наиболее подробно исследованных моделей схем является схема из функциональных элементов (СФЭ). В качестве характеристики оптимальности СФЭ можно рассматривать сложность — количество функциональных элементов, содержащихся в схеме. Таким образом, под сложностью булевой функции или оператора

будем понимать минимальную сложность схемы, реализующую данную функцию или оператор.

При этом часто рассматривались схемы, в которых не учитывались вполне естественные ограничения на размещение элементов схемы в плоскости или пространстве, способы их соединения, разводка проводов и т.п. В действительности, любая схема состоит из отдельных элементарных частей (функциональных элементов), которые имеют определенную длину, ширину и соединяются проводниками, размеры которых следует учитывать.

Доклад посвящен объёмным схемам, которые определяются аналогично плоским схемам, но в манхэттенском пространстве. Под объёмной схемой понимается укладка схемы из функциональных элементов в пространстве. Объёмная схема состоит из кубических элементов. Каждый кубический элемент реализует булев оператор, у которого в сумме не более 6 входов и выходов. Также используется такая мера сложности схемы, как потенциал. Он равен среднему значению количества единиц на всех внутренних узлах схемы. Неформально говоря, потенциал играет роль средней “энергии” схемы, необходимой для её функционирования.

В докладе приводится нижняя оценка потенциала в классе булевых частичных операторов, которая совпадает с верхней оценкой по порядку.

# Доклады семинара «Теория дискретных функций и приложения»

В осеннем семестре 2020 – 2021 учебного года на научном семинаре «Теория дискретных функций и приложения» под руководством профессора Дмитрия Николаевича Бабина и с.н.с. Ивана Леонидовича Мазуренко состоялся следующий доклад.

## О классах языков, состоящих из кодов булевых функций

студ. специалитета Нужков Н.Ю.

Кодом булевой функции, как обычно, называется вектор из нулей и единиц, состоящий из значений функции на лексикографически упорядоченных значениях аргументов. Если функции добавить фиктивную переменную слева, то код функции удваивается (дважды повторяется). Если добавить переменную справа, то удваивается каждый из символов кода. Таким образом, возникают новые операции над словами: удвоение слов и удвоение каждого символа. Сохраняют ли регулярные языки эти операции.

Автор показал, что замыкание относительно этой операции регулярных и контекстно-свободных языков, выводит результат за рамки регулярных и контекстно свободных языков. Языки, выдерживающие указанные операции, суть контекстно-зависимые языки или языки вычислимые на линейно-ограниченной машине Тьюринга.

Зафиксируем класс Поста и рассмотрим множество кодов функций из этого класса. Автор показал, что для любого класса Поста это множество распознаётся линейно-ограниченной машиной Тьюринга.

Классы же Поста распознаваемые конечными автоматами – суть класс сохранения нулей, класс сохранения единиц и классы, состоящие из констант.

Таким образом, возник естественный и простой пример языка, не распознаваемого конечными автоматами и автоматами с магазинной памятью.

# Доклады семинара «Автоматы и алгоритмы»

С начала 2020 года на научном семинаре «Автоматы и алгоритмы» под руководством к.ф.-м.н., н.с. Волкова Н.Ю. и аспирантки Ушаковой В.В. состоялось 15 докладов.

23 января 2020 года

## Поведение коллектива автоматов со связью на целочисленном луче

Студент бакалавриата Пулатов Э.

В ходе доклада были озвучены основные определения связанные с коллективом автоматов со связью. Понятия коллективов автоматов со связью и “лабиринтного монстра” являются синонимами.

В дальнейшем лабиринтный монстр с $m$ головками и радиусом обзора $R$ и скоростью перемещения $V$ будет обозначаться $M = (W_1, ...W_m)(R, V)$. Ставится задача реализации таким монстром вычислений на луче целочисленных функций. Аргументы задаются начальными координатами определённых головок, а результат вычисления – координатой определённой головки в момент остановки монстра.

В зависимости от времени вычисления функций, определяются быстрые и сверхбыстрые вычисления функций лабиринтным монстром. Доказана лемма о том, что последовательность выходных символов монстра с двумя головками на целочисленной прямой, является периодической. Доказана другая схожая лемма о том, что последовательность выходных символов лабиринтного монстра с одной головкой на луче - периодична.

6 февраля 2020 года

## Поведение в квадранте непериодических автоматов

Студентка бакалавриата Хегай Ю.

Доклад посвящен изучению перемещения в лабиринте, представляющем квадрант, автоматов, имеющих в этом лабиринте непериодическую последовательность выходных символов. Этот тип автоматных траекторий был ранее практически не изучен. Для изучения было введено понятие манёвра, как локально-периодического движения автомата. Разным типам движения в локальном периоде соответствуют разные манёвры. Целью проделанной работы является проверка гипотезы о том, что любой автомат в квадранте с непериодической последовательностью выходных символов имеет периодическую последовательность манёвров. Были введены понятия x- и y- вычислимости автоматом последовательностей. Удалось выявить два класса автоматов, имеющих в квадранте периодическую последовательность маневров.

За время изучения данного вопроса стало ясно, что это свойство имеет место для автоматов, которые x- и y-вычисляют последовательности, являющиеся арифметическими прогрессиями с произвольными разностями p1 и p2. Также удалось доказать, что если автомат x- и y-вычисляет последовательности, разности соседних членов которых являются периодическими последовательностями, то последовательность маневров автомата является периодической.

Доказано, что существует автомат, который x- вычисляет любую последовательность, являющуюся арифметической прогрессией с разностью , и при этом y-вычисляет некоторую последовательность, являющуюся арифметической прогрессией с той же разностью p. Также удалось выяснить, что существует автомат, который x- и y- вычисляет любые две последовательности, которые являются геометрическими прогрессиями с одинаковым натуральным показателем $q$.

19 марта 2020 года

## О вычислимости функций коллективами из двух автоматов

Аспирантка Ушакова В.

В работе рассматривается понятие вычислимости одноместных частичных функций $f : N_0' \to N_0 (N_0' \subseteq N_0)$ коллективом автоматов на целочисленной прямой. Значение аргумента задаётся расстоянием между определёнными автоматами коллектива в его начальной расстановке,

а результат равен расстоянию между определёнными автоматами коллектива в его финальной расстановке, при условии, что коллектив останавливается. Аналогично определяется вычислимость частичных многоместных функций: значение каждого аргумента задается расстоянием между определёнными автоматами коллектива в его начальной расстановке, а результат равен расстоянию между определёнными автоматами коллектива в его финальной расстановке, при условии, что коллектив останавливается.

Малые коллективы автоматов имеют более ограниченные вычислительные возможности, чем машины Тьюринга. Соответственно, классы функций, вычислимые коллективами автоматов, являются подклассами вычислимых функций.

Работа нацелена на расслоение всех одноместных вычислимых функций на подклассы по минимальному числу $n$ автоматов в коллективе, необходимом для вычисления данной функции. Класс частичных (многоместных) функций, вычислимый коллективами из $n$ автоматов, обозначается, как $F_n$. Начиная с некоторого $n$, коллектив из $n$ автоматов может моделировать любую, в том числе универсальную машину Тьюринга, т.е. соответствующий подкласс функций $F_n$ становится равным классу всех вычислимых функций $F$. Можем считать, что функция тем сложнее, чем больше автоматов требуется для её вычисления.

Найден класс функций, вычислимых коллективами из двух автоматов. Это периодические функции и простейшие линейные функции, которые, начиная с некоторого значения аргумента $x$ ведут себя, как $f(x) = x + C$.

9 апреля 2020 года

## О конструируемых множествах

Студент бакалавриата Миндуллин М.

В докладе рассматривается перемещение конечного инициального автомата с краской на плоскости. Функционирование автомата воспроизводит на ней цветную конфигурацию, включающую в себя те клетки, которые, начиная с определенного момента, приобретают цвет, отличный от 0 (белого), и не включающую клетки, остающиеся белыми всегда, а также становящиеся белыми сколь угодно часто.

Множество конструируемо, если существует автомат с краской, его конструирующий. Ставится задача исследовать конструируемость различных классов множеств.

Приведено доказательство конструируемости пустого множества, всей плоскости, всех конечных множеств, а также их алгебраических дополнений. Доказано, что эти множества образуют алгебру множеств из $Z^2$. Также доказана конструируемость бордюра (множества на $Z^2$, имеющего группу самосовмещений, содержащую параллельный перенос на ненулевой вектор, причем все векторы параллельного переноса в этой группе коллинеарны) при условии конечности его примитивной ячейки (подмножества бордюра, образующего весь бордюр при всевозможных самосовмещениях из группы, причем для собственных подмножеств ячейки это неверно). Во всех случаях оценена сложность конструирования указанных множеств автоматом с красками.

16 апреля 2020 года

## Операции над автоматами и агентами

Студентка магистратуры Маметниязова Н.

В докладе рассматривается сумма автоматов, введённая Н.Ю. Волковым, для решения задачи преследования на плоскости. Использование суммы автоматов помогло Н.Ю. Волкову свести задачу поимки произвольной жертвы с периодической последовательностью выходных символов на плоскости к задаче обхода лабиринта. Однако, данная конструкция, в общем случае, неприменима для бесконечной полосы.

В ходе изучения способов суммирования траекторий в лабиринтах было введено понятие лабиринтного агента. Также были показаны свойства лабиринтных агентов и связь этих свойств с аналогичными свойствами автоматов, перемещающихся в лабиринтах. Были представлены свойства операций автоматной и агентной суммы, автоматной и агентной проекций и доказаны свойства операций.

8 мая 2020 года

## Сверхбыстрые вычисления функций

Студент бакалавриата Пулатов Э.

В ходе доклада были представлены примеры сверхбыстрого вычисления двух элементарных функций. Доказана теорема о том, что функция суммы двух переменных вычислима лабиринтным монстром с двумя головками за сверхбыстрое время. Также была доказана теорема о том, что функция произведения двух переменных вычислима лабиринтным монстром с четырьмя головками за сверхбыстрое время.

21 мая 2020 года

## Однородные лабиринты

Студентка магистратуры Маметниязова Н.

В докладе рассматриваются простейшие $i-$мерные агенты, где число $i$ равно 1 или 2. Одномерные агенты – это агенты, у которых все выходные символы колинеарны. Двумерные агенты – это агенты, у которых существуют неколинеарные выходные символы.

Были введены понятия $i-$мерного однородного лабиринта, как лабиринта, допустимого для некоторого $i-$мерного простейшего агента и операции сужения агента на лабиринт. Показан способ получения лабиринта, порожденного простейшим агентом. Также были представлены примеры некоторых $i-$мерных агентов.

28 мая 2020 года

## Компьютерный симулятор автоматов в лабиринтах. Редактор автомата.

Студент бакалавриата Зияев А.

В ходе доклада был рассмотрен функционал симулятора инициального конечного автомата в шахматном лабиринте. В программе симулируется поведение автомата на плоскости, задаваемого таблицей переходов. Была рассмотрена реализация редактора автомата, а также проводились обсуждения по его пользовательскому интерфейсу. Реализация редактора осуществлялась возможностями языка C#.

Эта программа позволяет создавать лабиринты, которые в дальнейшем будут использоваться для обхода заданным автоматом. Работая в автономном режиме, программа позволяет рисовать лабиринт любой

сложности, находить кратчайший путь между двумя точками, задавать автомат, который в дальнейшем сможет обойти данный лабиринт. Для произвольно заданного лабиринта обеспечивается визуальное наблюдение за поведением автомата в нем. Программа может быть использована в качестве удобного "вспомогательного инструмента" для решения проблем, связанных с поведением автоматов в лабиринтах.

24 сентября 2020 года

## О вычислимости функций коллективами из трёх автоматов

Аспирантка Ушакова В.

В работе рассматривается понятие вычислимости одноместных частичных функций $f : N_0' \to N_0 (N_0' \subseteq N_0)$ коллективом автоматов на целочисленной прямой. Класс частичных (многоместных) функций, вычислимый коллективами из $n$ автоматов, обозначается, как $F_n$.

Ранее был найден класс функций, вычислимых коллективами из двух автоматов. Это периодические функции и простейшие линейные функции, которые, начиная с некоторого значения аргумента $x$ ведут себя, как $f(x) = x + C$. Очевидно, коллектив из трех автоматов моделирует коллектив из двух автоматов, то есть $F_2 \subseteq F_3$.

Установлено также, что коллективами из трех автоматов вычисляются функции вида $f(x) = [(C_1 * x + C_2)/C_3]$.

1 октября 2020 года

## Компьютерный симулятор автоматов в лабиринтах. Редактор автомата.

Студент бакалавриата Зияев А.

В ходе доклада был обсужден дальнейший план действий по улучшению программы. А именно, высказаны требования к функциональности симулятора автоматов, требования по улучшению пользовательского интерфейса и кода. Было составлено краткое техническое задание по доработке симулятора. Слушателями было предложено добавление определенного ряда функциональных особенностей, а именно реализация независимой системы автоматов и коллектива автоматов.

8 октября 2020 года

## О взаимно однозначном соответствии конфигурации на плоскости состоянию ленты машины Тьюринга

Студент бакалавриата Миндуллин М.

Рассматривается пустая лента машины Тьюринга. Каждая ячейка ленты нумеруется следующим образом. Ячейке, в которой находится головка в нулевой момент времени, присваивается номер 0. Клетки, расположенные правее клетки с номером 0, нумеруются нечетными натуральными числами, а расположенные левее – четными.

Далее рассматривается неокрашенная плоскость. Клетке, в которой находится автомат с красками в нулевой момент времени, присваивается номер 0. Оставшиеся клетки плоскости нумеруются натуральными числами по спирали.

При этом каждая клетка плоскости получает свой уникальный номер, который соответствует уникальному номеру некоторой ячейки ленты машины Тьюринга.

Сопоставив каждому символу входного алфавита машины Тьюринга некоторый символ из входного алфавита автомата с красками (пустому символу сопоставим белую краску, а остальным символам – цвета с соответствующим номером), получаем взаимно однозначное соответствие любого состояния ленты машины Тьюринга некоторой конфигурации плоскости.

16 октября 2020 года

## О реализациях элементарных операций

Студент специалитета Хайруллин А.

Рассматривается множество (частично-определённых) функций счётнозначной логики $P_{\aleph_0}$. Отображения $\varphi(n) : E_\varphi \to P_{\aleph_0}$, где $E_\varphi \subset P_{\aleph_0} \times ... \times P_{\aleph_0}$ назовём элементарными операциями. Каждая такая операция по $n$ частичным счётно-значным функциям строит новую функцию. Значения самой операции определены на некоторых наборах из $n$ фунций. Реализацией назовем произвольное подмножество элементарных операций. Показаны представления классических операций суперпозиции, минимизации и примитивной рекурсии в виде реализаций.

Каждая реализация порождает на множествах функций из $P_{\aleph_0}$ свой оператор замыкания, по аналогии с тем, как классическое замыкание порождается операциями суперпозиции, примитивной рекурсии и минимизации. Показано, что оператор замыкания, порождённый любой не более чем счетной реализацией, при применении к любому не более чем счетном подмножеству $P_{\aleph_0}$ даст множество, не равное $P_{\aleph_0}$. Т.е. показана неполнота любых счётных систем функций относительно любого конечно-порождённого или счётно-порождённого замыкания.

29 октября 2020 года

## Доклад по результатам курсовой работы "Компьютерный симулятор автомата в лабиринте. Графическая часть".

Студентка бакалавриата Абдуллаева К.

В докладе были продемонстрированы возможности компьютерного симулятора автомата в лабиринте, а именно его графическая часть. Были обсуждены планы дальнейшего развития программы для написания дипломной работы, а именно создание нового графического интерфейса с добавлением дополнительного функционала лабиринта и оптимизирование уже имеющегося.

12 ноября 2020 года

## О конструируемости двумерного орнамента

Студент бакалавриата Миндуллин М.

В докладе рассматривается двумерный орнамент – множество на $Z^2$, имеющее группу самосовмещений, содержащую параллельный перенос на ненулевой вектор, причем в этой группе содержится два неколлинеарных вектора параллельного переноса. Доказана конструируемость всех двумерных орнаментов автоматом с красками. Алгоритм представляет с собой модификацию алгоритма обхода плоскости автоматом по спирали, где перемещения на единичные векторы заменены последовательными перемещениями на векторы параллельного переноса, а окрашивание

клетки плоскости – на конструирование макроячейки, содержащей примитивную ячейку двумерного орнамента. Приведена оценка сложности конструирования орнамента автоматом с красками.

26 ноября 2020 года

## Расширение вычислительных возможностей машины Тьюринга. Простейшая и универсальная модель сверхтьюринговых вычислений

Научный сотрудник Волков Н.Ю.

Классическая машина Тьюринга задаётся программой конечного автомата, являющегося её головкой, и работает с изначально пустой лентой. Пользователь перед применением наносит на ленту конечную информацию (как правило, непосредственно правее начального расположения головки на ленте), запускает машину Тьюринга и, в случае её остановки, считывает с ленты конечную информацию, которую интерпретирует, как результат вычислений.

В классической теории алгоритмов алгоритмом считается именно программа для машины Тьюринга, а вычислимость любой функции понимается, как возможность её вычисления на некой машине Тьюринга.

Известно, что множество частичных счётнозначных функций $f : (N_0)^m \to N_0$, вычислимых по Тьюрингу, есть множество частично-рекурсивных функций.

Более того, несложно показать, что для любого определения алгоритма (не обязательно понимаемого, как программа машины Тьюринга), подразумевающего, что конкретный алгоритм задаётся конечным словом в конечном алфавите, множество функций, вычислимых при помощи алгоритмов такого вида, будет не более, чем счётно. А, значит, большая часть частичных счётнозначных функций будет невычислима.

Это означает, что любой вычислитель, универсальный для класса частичных счётнозначных функций $f : (N_0)^m \to N_0$, должен иметь программу бесконечной длины.

Предлагается простейшая модель универсального вычислителя в виде машины Тьюринга с лентой, правая половина которой (от начального расположения головки) изначально заполнена произвольной последовательностью в конечном алфавите. Будем называть такие машины **машинами Тьюринга с непустой (изначально) лентой**. Как и в

классическом случае, пользователь перед применением наносит на ленту конечную информацию (теперь, непосредственно левее начального расположения головки на ленте), запускает машину Тьюринга и, в случае её остановки, считывает с ленты конечную информацию, находящуюся в клетке, где остановилась головка и в клетках, левее её, которую интерпретирует, как результат вычислений.

Таким образом, теперь алгоритм задаётся конечной программой головки и бесконечной полулентой, содержащей последовательность букв конечного алфавита. Определённое таким образом множество алгоритмов – континуально.

Показано, что в данной вычислительной модели являются вычислимыми все частичные счётнозначные функции вида $f : (N_0)^m \to N_0$, все словарные функции вида $f : (A^m)^* \to A^*$ и, вообще, все конструктивные объекты.

Показано, что существует классическая машина Тьюринга (т.е. машина с изначально пустой лентой), которая по любому набору входных данных и любой программе для машины Тьюринга с непустой лентой моделирует работу этой машины на этих входных данных, т.е. существует классическая машина Тьюринга, универсальная для класса машин Тьюринга с непустой лентой.

## К сведению авторов публикаций в журнале «Интеллектуальные системы. Теория и приложения»

В соответствии с требованиями ВАК РФ к изданиям, входящим в перечень ведущих рецензируемых научных журналов и изданий, в которых могут быть опубликованы основные научные результаты диссертаций на соискание ученой степени доктора и кандидата наук, статьи в журнал «Интеллектуальные системы. Теория и приложения» предоставляются авторами в следующей форме:

1. Статьи, набранные в пакете LaTeX, предоставляются к загрузке через WEB-форму http://intsysjournal.org/generator_form .
2. К статье прилагаются файлы, содержащие название статьи на русском и английском языках, аннотацию на русском и английском языках (не более 50 слов), список ключевых слов на русском и английском языках (не более 20 слов), информация об авторах: Ф.И.О. полностью, место работы, должность, ученая степень и/или звание (если имеется), контактные телефоны (с кодом города и страны), e-mail, почтовый адрес с индексом города (домашний или служебный).
3. Список литературы оформляется в едином формате, установленном системой Российского индекса научного цитирования.
4. За публикацию статей в журнале «Интеллектуальные системы. Теория и приложения» с авторов (в том числе аспирантов высших учебных заведений) статей, рекомендованных к публикации, плата не взимается. Оттиски статей авторам не предоставляются. Журнал распространяется по подписке, экземпляры журнала рассылаются подписчикам наложенным платежом. Условия подписки публикуются в каталоге НТИ «Роспечать», индекс журнала 64559.
5. Доступ к электронной версии последнего вышедшего номера осуществляется через НЭБ «Российский индекс научного цитирования». Номера, вышедшие ранее, размещаются на сайте http://intsysjournal.org, и доступ к ним бесплатный. Там же будут размещены аннотации всех публикуемых статей.

Подписано в печать: 11.12.2020
Дата выхода: 21.12.2020
Тираж: 200 зкз.
Цена свободная
Свидетельство о регистрации СМИ: ПИ № ФС77–58444 от 25 июня 2014 г., выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций(Роскомнадзор).