

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 3 * 2020

МОСКВА

Главный редактор: д.ф.-м.н., профессор В. Б. Кудрявцев

Редакционная коллегия:

д.ф.-м.н., проф. А. Е. Андреев (зам. главного редактора)
д.ф.-м.н., проф. Э. Э. Гасанов (зам. главного редактора)
к.ф.-м.н., доц. А. С. Строгалов (зам. главного редактора)
к.ф.-м.н., м.н.с. В. В. Осокин (ответственный секретарь)
д.ф.-м.н., проф. В. В. Александров, д.ф.-м.н., проф. С. В. Алешин, д.ф.-м.н., проф. Д. Н. Бабин, академик РАН, д.ф.-м.н., проф. Ю. Л. Ершов, академик РАН, д.ф.-м.н., проф. Ю. И. Журавлев, д.ф.-м.н., проф. В. Н. Козлов, чл.-корр. РАН, д.ф.-м.н., проф. А. В. Михалев, к.ф.-м.н., проф. В. А. Носов, д.ф.-м.н., проф. А. С. Подколзин, д.т.н., проф. Д. А. Поспелов, д.ф.-м.н., проф. Ю. П. Пытьев, академик РАН, д.т.н., проф. А. С. Сигов, д.ф.-м.н., проф. А. В. Чечкин

Международный научный совет журнала:

С. Н. Васильев (Россия), К. Вашик (Германия), В. В. Величенко (Россия), А. И. Галушкин (Россия), И. В. Голубятников (Россия), Я. Деметрович (Венгрия), Г. Килибарда (Сербия), Ж. Кнап (Словения), П. С. Краснощеков (Россия), А. Нозаки (Япония), В. Н. Редько (Украина), И. Розенберг (Канада), А. П. Рыжов (Россия) — ученый секретарь совета, А. Саломая (Финляндия), С. Саксида (Словения), Б. Тальхайм (Германия), Ш. Ушчумлич (Сербия), Фан Дин Зиеу (Вьетнам), А. Шайеб (Сирия), Р. Шчепанович (США), Г. Циммерман (Германия)

Секретари редакции: И. О. Бергер

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».

© ООО «Интеллектуальные системы», 2020.

ОГЛАВЛЕНИЕ

Часть 1. Общие проблемы теории интеллектуальных систем

Суворова Е.И., Концевая А.В., Рыжов А.П., Сапунова И.Д., Мырзаматова А.О., Муканеева Д.К., Худяков М.Б., Драпкина О.М. Оценка и мониторинг эффективности популяционных мер профилактики заболеваний 7

Часть 2. Специальные вопросы теории интеллектуальных систем

Бернадотт А. Модификация конечного автомата через применение алгоритмов сжатия 25

Воротников А.С. О синтезе колонии жуков с линейным ростом 43

Часть 3. Математические модели

Быстрыгова А.В. Расшифровка булевых функций фиксированного веса .. 63

Васильев Д.И. Поиск ближайшего соседа на прямой с помощью клеточного автомата с локаторами 99

Кан А.Н. Решетка 1-следов замкнутых классов кусочно-линейных функций
121

Часть 4. Материалы семинаров кафедры МатИС

Доклады семинара «Вопросы сложности алгоритмов поиска» 145

Часть 1.
Общие проблемы теории
интеллектуальных систем

Оценка и мониторинг эффективности популяционных мер профилактики заболеваний

Суворова Е.И.¹, Концевая А.В.², Рыжов А.П.³, Сапунова И.Д.⁴,
Мырзаматова А.О.⁵, Муканеева Д.К.⁶, Худяков М.Б.⁷,
Драпкина О.М.⁸

¹ *Суворова Евгения Игоревна* — младший научный сотрудник отдела укрепления общественного здоровья ФГБУ «Национальный медицинский исследовательский центр терапии и профилактической медицины» МЗ РФ, e-mail: evgen-k@yandex.ru.

² *Концевая Анна Васильевна* — доктор медицинских наук, заместитель директора по научной и аналитической работе ФГБУ «Национальный медицинский исследовательский центр терапии и профилактической медицины» МЗ РФ, e-mail: akontsevaya@gnicpm.ru.

³ *Рыжов Александр Павлович* — доктор технических наук, профессор МГУ имени М.В. Ломоносова механико-математический факультет кафедра математической теории интеллектуальных систем, e-mail: ryjov@mail.ru.

⁴ *Сапунова Ирина Дмитриевна* — кандидат медицинских наук, научный сотрудник отдела укрепления общественного здоровья ФГБУ «Национальный медицинский исследовательский центр терапии и профилактической медицины» МЗ РФ, e-mail: 05121979-79@mail.ru.

⁵ *Мырзаматова Азалия Орозбековна* — кандидат медицинских наук, научный сотрудник отдела укрепления общественного здоровья ФГБУ «Национальный медицинский исследовательский центр терапии и профилактической медицины» МЗ РФ, e-mail: amyrzamatova@gnicpm.ru.

⁶ *Муканеева Динара Кямилловна* — младший научный сотрудник отдела укрепления общественного здоровья ФГБУ «Национальный медицинский исследовательский центр терапии и профилактической медицины» МЗ РФ, e-mail: dmukaneeva@gnicpm.ru.

⁷ *Худяков Михаил Борисович* — ведущий инженер отдела укрепления общественного здоровья ФГБУ «Национальный медицинский исследовательский центр терапии и профилактической медицины» МЗ РФ, e-mail: mkhudyakov@gnicpm.ru.

⁸ *Драпкина Оксана Михайловна* — профессор, доктор медицинских наук, член-корреспондент РАН, директор ФГБУ «Национальный медицинский исследовательский центр терапии и профилактической медицины» МЗ РФ, e-mail: odrapkina@gnicpm.ru.

Принятие эффективных управленческих решений в сфере охраны здоровья и профилактики заболеваний позволяет повысить продолжительность и качество жизни. Процесс принятия таких решений происходит в условиях ограниченной полноты и достоверности доступной информации и имеет существенную экспертную составляющую. В работе представлен один из подходов, позволяющих принимать обоснованные решения в области профилактической медицины.

Ключевые слова: профилактическая медицина, оценка и мониторинг процессов, гибридных интеллект.

1. Введение

Продолжительность и качество жизни является интегральным показателем здоровья популяции и одним из важных критериев оценки качества управления на уровне государств, регионов, муниципальных образований. Этот показатель зависит от множества параметров, как контролируемых (например, потребление алкоголя) так и не контролируемых (например, стихийные бедствия). Некоторыми контролируруемыми параметрами можно управлять, однако зависимости, описывающие влияние конкретного параметра на целевой показатель, носят качественный характер; влияние конкретного параметра зависит от состояния других параметров, которые могут как усиливать, так и ослаблять влияние выбранного. Все это не позволяет описать функционирование системы в виде непрерывных или дискретных моделей. Отметим, что подобные попытки предпринимались в рамках популяционной динамики, начиная с работ Т. Мальтуса (Thomas Robert Malthus, 1766-1834) и П. Ферхюльста (Pierre François Verhulst, 1804 - 1848) и заканчивая современными многоагентными системами (см., например, [14]). Надежных и работоспособных моделей, позволяющих понимать какие и параметры и как влияют на продолжительность жизни в конкретной популяции, так и не появилось. Это связано с рядом причин, к основным из которых можно отнести:

- фрагментарность доступных данных: проведение замеров значений большинства параметров, имеющих отношение к продолжительности жизни, является дорогостоящей процедурой, поэтому они проводятся редко и не по всей популяции; полученные же данные считаются справедливыми для всей популяции, что является достаточно спорным;

- наличие большого числа косвенных параметров, влияние которых на целевой показатель до конца не изучено, носит опосредованный характер через серию агрегированных параметров, взаимовлияние которых при агрегации также не изучено в достаточной для разработки моделей степени;
- значительные изменения в глобальной экономике (например, увеличение транспортных потоков товаров и людей, приводящее к быстрому распространению инфекций типа COVID 19) и образе жизни (например, повседневное использование гаджетов и продуктов быстрого питания) оказывают существенное влияние даже на известные зависимости.

Целью настоящей работы является описание нового подхода к прогнозированию эффективности популяционных мер профилактики на основе человеко-машинной технологии оценки и мониторинга сложных процессов. Системы оценки и мониторинга (СОиМ) позволяют обрабатывать ненадежную, фрагментарную, нечетко представленную информацию и получать максимально надежные для такого качества информации выводы.

Дальнейшее содержание работы организовано следующим образом. В разделе 2 приведен обзор существующих результатов моделирования эффективности популяционных мер профилактики, в разделе 3 представлен краткий обзор технологии оценки и мониторинга, в разделе 4 описан разработанный прототип модели и обсуждаются перспективы его развития и использования, в заключении подведены итоги проведенного исследования.

2. О моделировании эффективности популяционных мер профилактики

Моделирование – общепринятый подход построения прогноза состояния здоровья популяции для оценки влияния тех или иных профилактических мер или программ в средне- и долгосрочной перспективе. Кроме того, это распространенный подход к прогнозированию не только здоровья популяции, но и социально-экономического бремени болезней, что является важным аргументом при принятии решений в сфере укрепления здоровья [1].

Популяционное моделирование необходимо для планирования стратегий профилактики с учетом экономической целесообразности их внедрения. Существует ряд публикаций с результатами моделирования эффективности популяционных мер профилактики. Например, в Великобритании показана возможность экономии 100 млн. фунтов в системе здравоохранения за счет значительного сокращения заболеваемости и смертности при полной элиминации промышленных транс-жиров из продуктов питания [2]. Тем временем, в модели Thow A.M. и др. спрогнозировано снижение смертности от ишемической болезни сердца (ИБС) на 1,8-2,6% при повышении налога на добавленную стоимость на основные источники насыщенных жиров [3]. В США по результатам моделирования показано, что субсидия в размере 10%, направленная на снижение стоимости овощей и фруктов, может предотвратить 150500 смертей от сердечно-сосудистых заболеваний (ССЗ) [4]. Mozaffarian и др. предложили сценарий субсидирования за счет использования дохода в размере 580 млн. долларов от налогообложения менее полезных продуктов питания, что вероятно приведет к предотвращению около 6000 случаев смерти от ССЗ и онкологических заболеваний ежегодно [5].

Однако, зарубежные модели не всегда могут быть применимы в России, а их адаптация ассоциирована с очень большим количеством допущений, что снижает их прогностическую ценность. В РФ также неоднократно обсуждались аспекты моделирования социально-экономического эффекта популяционных мер для принятия управленческих решений в области профилактики хронических неинфекционных заболеваний (ХНИЗ), однако, в настоящий момент публикаций с результатами такого моделирования нет [1].

Таким образом, в настоящее время существует необходимость разработки модели популяционной профилактики поведенческих факторов риска ХНИЗ с учетом национальной специфики и доступности данных, которая бы обладала максимально возможной точностью для российской популяции.

3. О системах оценки и мониторинга сложных процессов

Системы оценки и мониторинга сложных процессов - специализированные человеко-компьютерные интеллектуальные информационные системы, предназначенные для оценки состояния некоторого процесса на

базе фрагментарной, ненадежной, возможно недостоверной и противоречивой информации о нем и моделирования возможных сценариев его развития. Системы оценки и мониторинга возникли в начале 90-х годов и достаточно широко представлены в научной литературе. В качестве обзорной работы можно порекомендовать [15], детали представлены в [16]. Основными научными проблемами разработки таких систем являются:

- Проблема 1 (Perception-modelling): как человек описывает объекты? Можем мы описывать объекты наиболее надежным и эффективным для последующей обработки образом?
- Проблема 2 (Perception-based computing): как мы обрабатываем perception-based описания (например, ищем или обобщаем)? Можем мы оптимизировать такие вычисления?
- Проблема 3 (Aggregation of perception-based data): как происходит агрегирование оценок в нечетких иерархических динамических системах, минимизирующее противоречивость модели процесса в системах оценки и мониторинга?

Эти проблемы обсуждаются в упомянутых выше работах. Детали представлены в работах [17] (Проблема 1), [18] (Проблема 2) и [19] (Проблема 3). Учитывая доступность перечисленных работ, мы позволим себе не останавливаться на них. Отметим лишь, что системы оценки и мониторинга эффективны, когда нет (нельзя построить) математической модели процесса в виде уравнений, автоматов, и т.п. Разработка таких систем возможна, когда можно построить «семантическую модель» процесса в виде набора понятий и их взаимосвязей (рис. 1), и когда поступает и анализируется реальная информация (возможно обучение или настройка). Возможна разработка оптимальных систем с точки зрения удобства ввода информации аналитиком, согласованности мнений аналитиков, информационного обеспечения ввода информации, и моделирования.

На основе описанных методов разработаны системы оценки и мониторинга процессов создания странами специальных видов технологий и материалов в интересах управления гарантией Международного Агентства по Атомной Энергии [20], риска развития сердечно-сосудистых заболеваний [21], разработки изделий микроэлектроники (оценка технологических стартапов) в интересах компании Cadence Design Systems [22]. Воз-

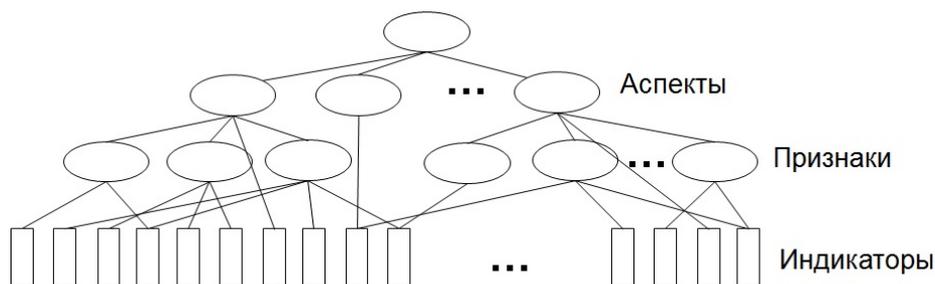


Рис. 1. Структура модели процесса.

возможности применения подобных систем в оценке и мониторинге кризисных ситуаций и в стратегическом управлении компаниями описаны в [23] и [24] соответственно. Уникальные аналитические возможности систем оценки и мониторинга обсуждаются в [15] и суммированы на рис. 2. Текущий уровень почти всех компаний – OLAP аналитика; лишь немногие компании используют Big Data аналитику; аналитические инструменты систем оценки и мониторинга используются в единичных международных и правительственных организациях, а также в крупном международном высокотехнологическом бизнесе. Использование аналитических возможностей систем оценки и мониторинга – это естественный эволюционный этап развития аналитических инструментов.

4. Прототип модели эффективности популяционных мер профилактики на основе СОиМ

Как было отмечено выше (раздел 3), разработка систем оценки и мониторинга возможна, когда можно построить «семантическую модель» процесса в виде набора понятий и их взаимосвязей (рис. 1). Разработка прототипа такой модели была проведена коллективом авторов.

Разработка включала в себя следующие этапы:

- 1) Выбор основных групп ХНИЗ с высокой заболеваемостью и смертностью в РФ
- 2) Выбор поведенческих факторов риска, ассоциированных с основными ХНИЗ

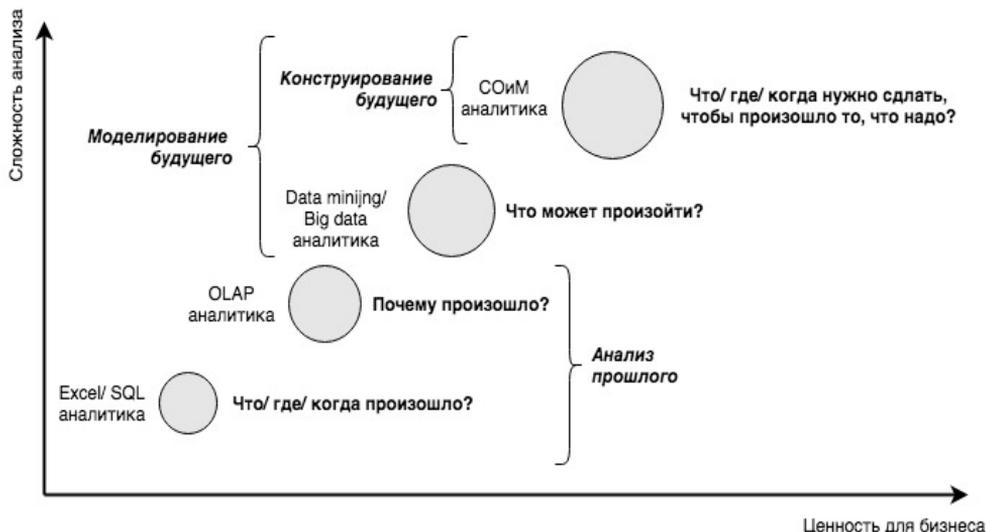


Рис. 2. Эволюция аналитических инструментов.

- 3) Поиск мер популяционной профилактики, влияющих на распространенность поведенческих факторов риска
- 4) Анализ степени внедрения/возможности внедрения мер популяционной профилактики, влияющих на распространенность поведенческих факторов риска и отбор мер для моделирования

На первом этапе были отобраны основные ХНИЗ, ассоциированные с высокой заболеваемостью и смертностью в РФ, в отношении которых есть эффективные меры профилактики. К таким ХНИЗ как во всем мире, так и в РФ, относятся сердечно-сосудистые заболевания (ССЗ), рак, сахарный диабет (СД), хроническая обструктивная болезнь легких (ХОБЛ), которые также ассоциированы со значительным экономическим ущербом [6].

На следующем этапе на основе анализа литературы были отобраны факторы, ассоциированные с увеличением риска развития данных ХНИЗ, т.е. факторов, которые определяют количественные изменения эпидемиологических и демографических показателей в популяции, создана гипотеза в отношении возможных причин изменения поведения кривой распространенности, заболеваемости и смертности в РФ. Отбор факторов проведен и описан ранее [7]. К таким факторам отнесли куре-

ние, избыточное потребление алкоголя, элементы нерационального питания и низкую физическую активность. В процессе анализа поведенческих факторов риска, онкологические заболевания были сокращены до 10 локализаций, в отношении которых существуют доказательные данные их ассоциации с поведенческими факторами риска. Отбор мер популяционной профилактики производился на основании анализа литературы и экспертным путем: эксперты предполагали перечень категорий профилактических мер, которые, по данным литературы, оказывают влияние на изучаемые ФР ХНИЗ и имеют потенциал внедрения в РФ.

В рамках отобранных категорий мер был проведен аналитический обзор исследований по оценке эффективности реализации мер популяционной профилактики, направленных на коррекцию изучаемых поведенческих факторов риска (курение, потребление алкоголя, нерациональное питание) основных ХНИЗ с целью выделения мер с доказанной эффективностью, в том числе рекомендованных ВОЗ, с потенциалом внедрения в РФ. На этом этапе был исключен такой фактор риска как физическая активность, в связи с дефицитом исследований эффективности мер популяционной профилактики.

В результате проведения обзора был накоплен достаточно большой объем научных знаний об эффективности популяционных мер профилактики в разных странах, однако, полученная из различных источников информация была фрагментарной, разноуровневой и с разной степенью надежности, что существенно усложняло процесс построения модели и полностью исключало возможность применения таких математических инструментов как статистические методы моделирования.

В результате выполнения этапов 1 – 4 получилась модель, структурная (или декларативная в терминах представления знаний) часть которой представлена на рис. 3. Фрагмент модели, отражающий ее структуру и достаточный для ее понимания и анализа, представлен на рис. 4.

Поведенческая (или процедуральная) часть модели состоит из набора правил, приписанных каждому узлу модели. Часть правил формулируется напрямую из результатов исследований (например, «Полный запрет на рекламу в Интернет приводит к снижению потребления алкоголя на 16%»), часть правил формируется как результат обработки данных известных исследований (например, известно, что (1) «Снижение цен на 1% приводит к увеличению потребления овощей и фруктов на 1%» и (2) «Снижение цен на 10% приводит к увеличению потребления овощей и фруктов на 15%», формулируется общее правило «Снижение цен на

$x\%$ приводит к увеличению потребления овощей и фруктов на $\frac{14x-5}{9}\%$; используемая гипотеза о линейной зависимости снижения цен и увеличения потребления для овощей и фруктов согласуется с экспертами).

Имея такую модель и оценки стоимости мер профилактики (субсидии, снижение цен, реклама, маркировка, запреты и пр.), а также известные затраты на лечение ХНИЗ и потери от преждевременной смертности, становится возможным оценивать эффективность различных мер профилактики и/ или их комбинаций в терминах снижения бюджетных затрат на лечение и потерь от смертности или в терминах снижения ХНИЗ. Такой сценарий использования системы (изменить значение меры профилактики и вычислить изменение целевого параметра) называется прямой задачей. Возможно также решение более сложных и интересных для практики обратных задач.

Обратная задача позволяет оптимизировать бюджет на достижение определённого уровня целевого параметра. Если задан бюджет, и мы знаем стоимость изменения состояния меры профилактики, то возможно нахождение тех мер, изменение которых находится в рамках заданного бюджета и даёт максимальной эффект повышения целевого параметра модели. Формализовать такую постановку можно следующим образом. Пусть модель содержит N мер профилактики $\{a_1, \dots, a_N\}$, и целевой параметр g (это может быть корневой узел модели - ХНИЗ в целом, элемент второго уровня модели - конкретное ХНИЗ, или фактор риска). К i -й мере приписано m_i значений: $a_i = \{a_i^1, a_i^2, \dots, a_i^{m_i}\} (i = 1, \dots, N)$. Пусть задано: X - бюджет, c_i - стоимость изменения состояния i -й меры (перевода ее из состояния a_i^k в состояние a_i^{k+1} ($k = 1, \dots, m_i - 1$)); будем для простоты записи считать такую стоимость одинаковой для всех k). Обозначим через Δg силу изменения целевого параметра g .

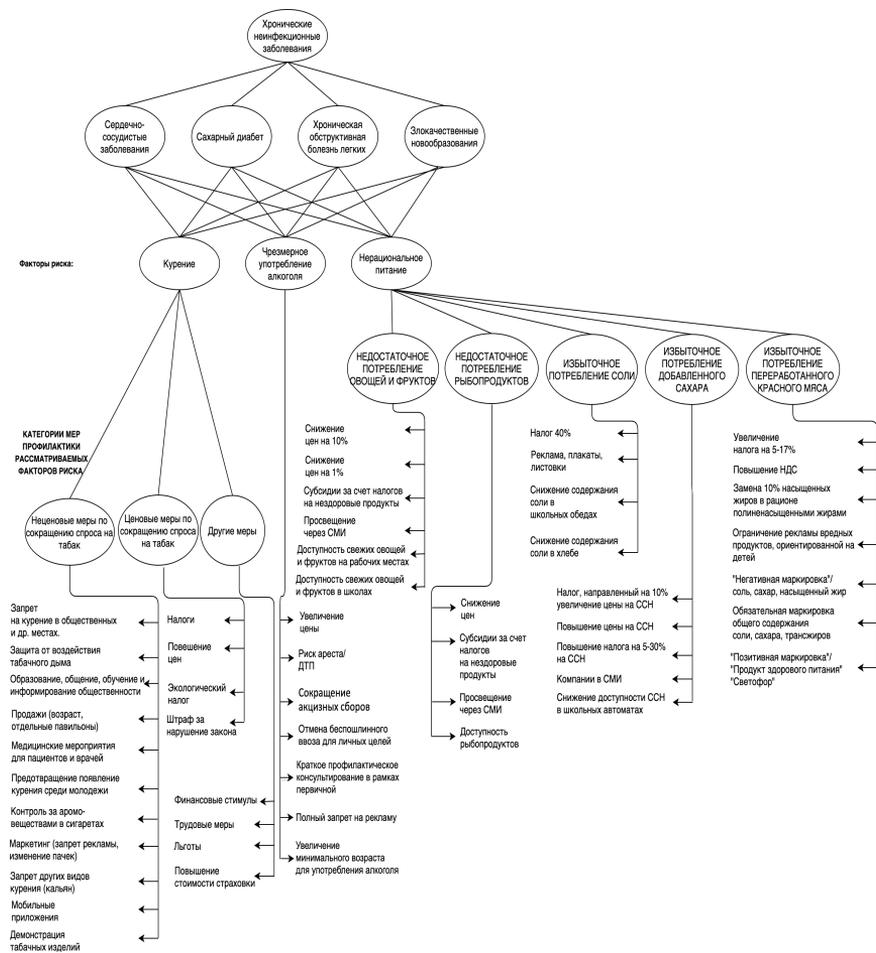


Рис. 3. Структура прототипа модели.

что оптимальный для некоторого региона набор мер профилактики может не являться таковым для других регионов. Последнее означает, что разрабатываемая система может быть использована для оптимизации эффективности бюджетных расходов в части здравоохранения в регионах, городах и всех других органах власти – владельцев бюджетов.

5. Заключение

Принятие эффективных управленческих решений в сфере охраны здоровья является важной задачей, так как позволяет повысить продолжительность и качество жизни. В настоящее время такие решения принимаются специалистами без применения средств моделирования; свойства доступной информации таковы, что построение классических математических моделей (как непрерывных, так и дискретных) затруднено.

В работе описывается прототип модели на основе человеко-машинной технологии оценки и мониторинга сложных процессов, позволяющей совместно обрабатывать разнородную, фрагментарную, плохо измеримую информацию по проблеме. Разработка системы на основе такой модели позволит оценивать эффективность различных мер профилактики и/или их комбинаций в терминах снижения бюджетных затрат на лечение и потерь от смертности или в терминах снижения ХНИЗ. Возможность решения обратных задач позволяет находить такой набор профилактических мер, который дает максимальный эффект сокращения ХНИЗ в рамках заданного бюджета и/или такой набор профилактических мер, который обеспечивает заданный эффект сокращения ХНИЗ в рамках минимального возможного бюджета. Решение перечисленных задач позволит максимально эффективно расходовать бюджеты различных уровней в части здравоохранения.

Список литературы

- [1] A.V. Kontsevaya, S.A. Shalnova, E.I. Suvorova et al., “The Prediction Model of Cardiovascular Events Among the Russian Population: Methodological Aspects”, *Kardiologiya*, **12** (2016), 54–62; русский перевод: А.В. Концевая, С.А.Шальнова, Е.И. Суворова и др., “Модель прогнозирования сердечно-сосудистых событий в российской популяции: методологические аспекты”, *Кардиология*, **12**, 2016, 54–62 <https://doi.org/10.18565/cardio.2016.12.54-62>.
- [2] Pearson-Stuttard J., Hooton W., Critchley J., et al., “Cost-effectiveness analysis of eliminating industrial and all trans fats in England and

- Wales: modelling study”, *Public Health (Oxf)*, **39**:3 (2017), 574–582
<https://doi.org/10.1093/pubmed/fdw095>.
- [3] Thow A.M., Jan S., Leeder S., Swinburn B., “The effect of fiscal policy on diet, obesity and chronic disease: a systematic review”, *Bull World Health Organ*, **88** (2010), 609–614.
 - [4] Pearson-Stuttard J., Bandosz P., Rehm C.D., et al., “Reducing US cardiovascular disease burden and disparities through national and targeted dietary policies: A modelling study”, *PLoS Med*, **14**:6 (2017).
 - [5] Mozaffarian D., Afshin A., Benowitz N.L., et al., “Population approaches to improve diet, physical activity, and smoking habits: a scientific statement from the American Heart Association”, *Circulation*, **126** (2012), 1514–1563.
 - [6] Bloom D.E., Cafiero E.T., Jane-Lopis E., Abrahams-Gessel S., Bloom L.R., Fathima S., Feigl A.B., Gaziano T., Mowafi M., Pandya A., Prettner K., Rosenberg L., Seligman B., Stein A., Weinstein C., “The Global Economic Burden of Non-communicable Diseases”, *Geneva 2011: World Economic Forum*, http://www.weforum.org/docs/WEF_Harvard_HE_GlobalEconomicBurdenNonCommunicableDiseases_2011.pdf.
 - [7] Мырзаматова А.О., Концевая А.В., Баланова Ю.А., Муқанеева Д.К., “Аналитический обзор ассоциации поведенческих факторов риска с хроническими неинфекционными заболеваниями”, *Профилактическая медицина*, **22**:5 (2019), 117–123.
 - [8] Chen Yang, Henry Ambayo, Bernard De Baets et al., “An Ontology to Standardize Research Output of Nutritional Epidemiology: From Paper-Based Standards to Linked Content”, *Nutrients*, **11** (2019), 1300
<https://doi.org/10.3390/nu11061300>.
 - [9] Vitali F., Lombardo R., Rivero D. et al., “ONS: an ontology for a standardized description of interventions and observational studies in nutrition”, *Genes Nutr*, **13** (2018), 12 <https://doi.org/10.1186/s12263-018-0601-y>.
 - [10] Michie S., Thomas J., Johnston M. et al., “The Human Behaviour-Change Project: harnessing the power of artificial intelligence and machine learning for evidence synthesis and interpretation”, *Implement Sci*, **12**:1 (2017), 121
<https://doi.org/10.1186/s13012-017-0641-5>.
 - [11] Norris E., Finnerty A.N., Hastings J., Stokes G., Michie S., “A scoping review of ontologies related to human behaviour change”, *Nat Hum Behav*, **3**:2 (2019), 164–172 <https://doi.org/10.1038/s41562-018-0511-4>.
 - [12] Riaño D., Real F., López-Vallverdú J.A. et al., “An ontology-based personalization of health-care knowledge to support clinical decisions for chronically ill patients”, *J Biomed Inform*, **45**:3 (2012), 429–460
<https://doi.org/10.1016/j.jbi.2011.12.008>.
 - [13] S.A. Boytsov, S.A. Shalnova, A.V. Kontsevaya, A.D. Deev, Yu.A. Balanova, A.V. Kapustina, “Trends in simulated 10-year mortality rates and the evaluation of the socioeconomic efficiency of different scenarios of prevention”, *The Russian Journal of Preventive Medicine and Public Health*, **19**:3 (2016), 12–18; русский перевод: Бойцов С.А., Шальнова С.А., Концевая А.В., Деев А.Д., Баланова Ю.А., Капустина А.В., “Динамика моделированной 10-летней смертности и оценка социально-экономической эффективности

- различных сценариев профилактики”, *Профилактическая медицина*, **19:3** (2016), 12–18 <https://doi.org/10.17116/profmed201619312-18>.
- [14] Селяков И.С., “Определение структуры многоагентной системы для моделирования популяции животных”, *Труды СПИИРАН*, т. 4, 2017, 437–444 <https://doi.org/10.15622/sp.4.33>.
- [15] Рыжов А.П., “Математические задачи систем оценки и мониторинга сложных процессов. Обзор постановок и результатов”, *Интеллектуальные системы. Т. 19, Вып. 1*, 2015, 5–20.
- [16] Рыжов А.П., “Информационный мониторинг сложных процессов: технологические и математические основы”, *Интеллектуальные системы, Том 11, вып. 1-4*, 2008, 101–136.
- [17] Рыжов А.П., “Элементы теории нечетких множеств и измерения нечеткости”, Москва, *Диалог-МГУ*, 1998, 116, <http://www.intsys.msu.ru/staff/ryzhov/FuzzySetsTheoryApplications.html>.
- [18] Рыжов А.П., “Модели поиска информации в нечеткой среде”, Москва, *Издательство Центра прикладных исследований при механико-математическом факультете МГУ*, 2004, 96, <http://www.intsys.msu.ru/staff/ryzhov/FuzzyRetrieval2010.html>.
- [19] Рыжов А.П., “Об агрегировании информации в нечетких иерархических системах”, *Интеллектуальные системы, Том 6, Вып. 1-4*, 2001, 341–364, [http://www.intsys.msu.ru/magazine/archive/v6\(1-4\)/ryzhov.pdf](http://www.intsys.msu.ru/magazine/archive/v6(1-4)/ryzhov.pdf).
- [20] Ryjov, A., Belenki, A., Hooper, R., Pouchkarev, V., Fattah, A. and Zadeh, L.A., “Development of an Intelligent System for Monitoring and Evaluation of Peaceful Nuclear Activities (DISNA)”, *IAEA, STR-310, Vienna*, 1998, 122.
- [21] Ахмеджанов Н.М., Жукоцкий А.В., Кудрявцев В.Б., Оганов Р.Г., Расторгуев В.В., Рыжов А.П., Строгалов А.С., “Информационный мониторинг в задаче прогнозирования риска развития сердечно-сосудистых заболеваний”, *Интеллектуальные системы, Т.7, вып. 1-4*, 2003, 5–38.
- [22] Лебедев А.А., Рыжов А.П., “Оценка и мониторинг проектов разработки высокотехнологических изделий микроэлектроники”, *Известия ТРТУ, Тематический выпуск "Интеллектуальные САПР ISBN 5-8327-0249-2*, **8** (2006), 93–99.
- [23] Ryjov A., “Basic principles and foundations of information monitoring systems. In: Monitoring, Security, and Rescue Techniques in Multi-Agent Systems”, *Springer. 3-540-23245-1, ISSN 16-15-3871*, 2005, 147–160.
- [24] Ryjov A., “Information Monitoring Systems as a Tool for Strategic Analysis and Simulation in Business”, *International Conference on Fuzzy Sets and Soft Computing in Economics and Finance FSSCEF 2004 Proceedings, Saint-Petersburg, Russia, June 17-20, ISBN 968-489-028-1*, **2** (2004), 511–520.

**Evaluation and monitoring of the effectiveness of
population-based disease prevention measures**
Suvorova E. I., Kontsevaya A.V., Ryjov A. P., Sapunova I. D.,

**Myrzamatova A. O., Mukaneeva D. K., Khudyakov M. B.,
Drapkina O.M.**

The adoption of effective management decisions in the field of healthcare and disease prevention can increase the duration and quality of life. The process of making such decisions takes place in conditions of limited completeness and reliability of the available information and has a significant expert component. The paper presents one of the approaches that allow making informed decisions in the field of preventive medicine.

Keywords: reventive medicine, process evaluation and monitoring, hybrid intelligence.

Часть 2.
Специальные вопросы теории
интеллектуальных систем

Модификация конечного автомата через применение алгоритмов сжатия

Бернадотт А.¹

Аннотация

Решение вопроса о принадлежности слова регулярному языку находит приложение в областях, где осуществляется поиск определенных паттернов в данных различной природы. Актуальной является проблема роста числа состояний распознающего детерминированного конечного автомата (ДКА) от числа регулярных выражений распознаваемого языка — проблема экспоненциального взрыва. В данной статье рассматривается модификация конечного автомата через применение алгоритмов сжатия, работающих без изменения распознаваемого языка и без добавления дополнительных структурных элементов автомата.

Ключевые слова: ДКА, НДКА, регулярный язык, экспоненциальный взрыв, алгоритм сжатия.

1. Введение

Решение вопроса о принадлежности слова регулярному языку находит приложение в разных областях, где является актуальным поиск определенных паттернов в данных различной природы. Так, алгоритмы проверки принадлежности слова регулярному языку применяются в системах определения сетевого протокола, системах обнаружения вторжений с поиском по сигнатурам, сетевых процессорах, в анализе биохимических данных (в частности для поиска лиганда рецептора в протеоме) [1, 2, 3, 4, 5, 6, 7].

¹*Бернадотт Александра* — к.м.н., аспирантка каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: alexandra.bernadotte@gmail.com.

Bernadotte A. — PhD in Med, graduate student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

Решение вопроса о принадлежности слова регулярному языку в классическом варианте осуществляется детерминированным конечным автоматом (ДКА), обеспечивающим быстроту работы, но большую пространственную сложность для некоторых регулярных языков. Классической альтернативной ДКА является недетерминированный конечный автомат (НДКА), представляющий регулярный язык. Согласно теореме о эквивалентности детерминированных и недетерминированных конечных автоматов, всякий язык принимается некоторым НДКА тогда и только тогда, когда этот язык принимается некоторым ДКА [8]. Кроме того, существует алгоритм построения из НДКА эквивалентного ему ДКА [8, 9]. При этом в худшем случае НДКА требует удержания в памяти всех состояний.

Таким образом, актуальной остается проблема экспоненциального взрыва — проблема роста числа состояний распознающего детерминированного конечного автомата (ДКА) от числа регулярных выражений распознаваемого языка. В общем случае при входном алфавите мощности более или равной двум мощность множества состояний экспоненциально возрастает [10, 11, 12]. К признакам, вызывающим экспоненциальный взрыв, относятся: использование в регулярных выражениях сочетаний “.” и “*” и “считающих” выражений, типа “ $\{n\}$ ” и “ $[-a_i]\{n\}$ ”, представленных в виде PCRE (Perl Compatible Regular Expressions) [13].

Глобально можно выделить три алгоритмических подхода к решению проблемы экспоненциального взрыва:

- 1) модификация распознаваемого регулярного языка;
- 2) модификация структуры конечного автомата через добавление специальных структурных элементов;
- 3) модификация конечного автомата через применение алгоритмов сжатия.

Первый подход — подход через модификацию распознаваемого регулярного языка основан на изменении распознаваемого языка с целью упрощения распознающего автомата. Александровым была предложена оптимизация языков вида $\bigcup_{i=1}^n \cdot * R_i \cdot * R'_i \cdot *$ за счет замены пары слагаемых на слагаемое $(R_{i_1} | R_{i_2}) \cdot * (R'_{i_1} | R'_{i_2})$. Такая замена очевидным образом расширяла язык, однако практически в системах обнаружения вторжений модификация снижала требование к памяти на порядок. Оценка числа состояний конечного автомата для такого вида языка и ошибка погрешности при распознавании принадлежности слова искомому регулярному языку при данном расширении языка приведены Александровым в следующих работах [14, 15, 16].

К второму подходу относятся следующие решения: двойной конечный автомат, представляющий собой комбинацию конечных автоматов, один из которых несет вспомогательную функцию хранения метасостояния другого; расширенный ДКА (в первоисточнике — extended finite automata), где дополнительно хранится битовый массив метасостояний автомата и используются счетчики для языков типа $\{n\}$; ДКА с счетчиками для решения проблемы экспоненциального взрыва для языка $\bigcup_{i=1}^n \cdot * \alpha_i \cdot * \beta_i \cdot *$, где α_i, β_i — непустые слова в алфавите A [5, 7, 17].

В данной статье нам бы хотелось глубже рассмотреть вопросы, описанные в работах Александра [18, 19], и сфокусироваться на третьем подходе — модификации конечного автомата через применение алгоритмов сжатия. Подход заключается в модификации структуры конечного автомата без изменения распознаваемого языка и без добавления дополнительных структурных элементов.

2. Обзор модификаций конечного автомата через применение алгоритмов сжатия

2.1. Методы сжатия с использованием недетерминированных переходов на случай ошибки

Большинство методов сжатия без использования дополнительных структурных элементов основаны на добавлении недетерминированных переходов в распознающий заданный язык ДКА.

2.1.1. Алгоритм Ахо-Корасик — НДКА с переходами в случае ошибки [20]

В 1975 году Альфред Ахо и Маргарет Корасик (Alfred Aho и Margaret Corasick) разработали алгоритм, осуществляющий поиск слов над алфавитом A из заданного словаря в строке. Алгоритм строит модифицированный инициальный ДКА на основе префиксного дерева [21]. Сжатие осуществляется через использование “переходов в случае ошибки” — переходов, которые отсутствуют в предварительно построенном префиксном дереве. То есть для каждого состояния нужно хранить информацию только о двух исходящих переходах: о детерминированном переходе по соответствующему символу из алфавита A (переход соответствует переходу префиксного дерева) и о переходе в случае “иначе”. Такая мо-

дификация ДКА позволяет сократить количество переходов, оставляя количество состояний исходного ДКА интактным.

Формально, полученный НДКА с переходами в случае ошибки, описывается следующим образом: $V = (A, Q, F, \varphi_{Q \times A}, \varphi_{er}, q_0)$, где A — входной алфавит, Q — алфавит состояний, F — алфавит принимающих состояний, $F \in Q$, $\varphi_{Q \times A}$ — функция перехода, определенная на $Q \times A$, φ_{er} — функция перехода в случае ошибки, определенная на Q . Алгоритм построения функции в случае ошибки представлен ниже — см. Algorithm 1. В алгоритме построения функции в случае ошибки используется термин “глубина состояния”, что есть минимальное число переходов от данного состояния до начального состояния конечного автомата. Аналогично используется термин “глубина вершины”. Вычислительная

Algorithm 1 Алгоритм построения функции в случае ошибки, φ_{er}

- 1: Вход: префиксное дерево G для соответствующего множества слов S над алфавитом A . Q — множество вершин префиксного дерева, которые соответствуют состояниям искомого НДКА с переходами в случае ошибки, для которого не определена φ_{er} . Выход: φ_{er} НДКА с переходами в случае ошибки.
 - 2: **for** q_i из Q **do**
 - 3: Определить глубину вершины (глубину состояния искомого НДКА с переходами в случае ошибки), d_i
 - 4: **if** $d_i \leq 1$ **then**
 - 5: $\varphi_{er}(q_i) = 0$
 - 6: **end if** $k = 1$
 - 7: **for** q_i из Q , при условии $d_i \geq k$ **do**
 - 8: **for** q_j из Q , при условии $d_j \geq k$ **do**
 - 9: **for** a_k из A , для которых не определена $\varphi(q_i, a_k)$ **do**
 - 10: Найти q_j , которое соответствует самому длинному суффиксу подслова, состоящего из букв, являющимися непустыми метками подряд идущих переходов, ведущих в состояние q_i , и для которого определена $\varphi(q_j, a_k)$
 - 11: $\varphi(q_i, a_k) == \varphi(q_j, a_k)$
 - 12: **end for**
 - 13: **end for**
 - 14: увеличиваем k на единицу
 - 15: **end for**
 - 16: **end for**
-

сложность алгоритма Ахо-Корасик напрямую зависит от организации

данных. Пространственно, НДКА Ахо-Корасик с переходами в случае ошибки имеет $\sum_{i=1}^n |s_i|$ состояний, где s_i — слово из множества распознаваемых слов S , а число переходов сокращается с $|A||Q|$ для исходного ДКА ($V = (A, Q, F, \varphi, q_0)$) до $2 \sum_{i=1}^n |s_i|$ в худшем случае для искомого НДКА Ахо-Корасик с переходами в случае ошибки.

2.1.2. Сжатие ДКА на основе общего префикса регулярных выражений

Логичным переходом от алгоритма Ахо-Корасик, работающего на строках, является поиск и использование префикса на регулярных выражениях, позволяющий сжать ДКА на основе нахождения эквивалентных вершин, соответствующих регулярным выражениям, Perl-совместимая запись которых имеет один префикс. Иллюстрация данного перехода от строкового алгоритма Ахо-Корасик на регулярные выражения приведена в работе [2] (см. рис 1). Однако более подробно данный переход на регулярные выражения был осуществлен Кумаром и соавторами и представлен в виде НДКА с задержками.

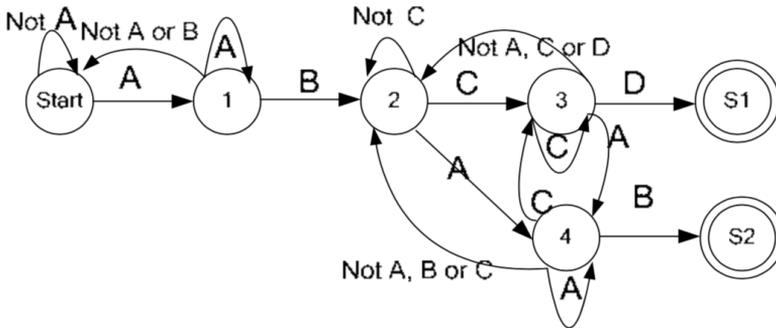


Рис. 1. ДКА, принимающий регулярные выражения $.*ABCD$ и $.*ABAB$ с общим префиксом. Источник: [2].

2.1.3. НДКА с задержками [1, 22, 23]

НДКА с задержками (в первоисточнике — Delayed Input DFA, или сокращенно — D2FA) представляет собой инициальный автомат, переходы которого помечены символами входного алфавита или пустым словом: $D2FA = \{A \cup \{\lambda\}, F, Q, \varphi, q_0\}$, A — входной алфавит, λ — пустое слово,

F — алфавит выходных состояний, φ — функция перехода, q_0 — начальное состояние (рис. 2) [1].

Ребра, помеченные пустым словом, носят название — “переход по умолчанию”. Каждое состояние данного ДКА с задержками может иметь не более одного перехода по умолчанию. Само построение НДКА с задержками из ДКА опирается на лемму:

Лемма 1. *Если $\varphi(a, q_i) = \varphi(a, q_j)$ для всех a_k из алфавита A , то множество переходов из состояния q_i эквивалентно множеству, состоящему из перехода по умолчанию в состояние q_j .*

При построении НДКА с задержками учитывается такая характеристика, как “самый длинный δ -путь” — самый длинный путь, состоящий исключительно из переходов по умолчанию. При выборе автомата предпочтение отдается автомату с минимальной данной характеристикой из эквивалентных НДКА с задержками. При этом эквивалентность НДКА₁ с задержками и НДКА₂ с задержками это выполнимость для любой строки x следующих условий $\varphi_{1i}(x) = \varphi_{2j}(x)$, где φ_1, φ_2 — функции переходов НДКА₁ и НДКА₂ соответственно, и эквивалентность множеств принимающих состояний НДКА₁ и НДКА₂.

Алгоритм построения оптимального НДКА с задержками состоит из нескольких стадий:

1) Построение ненаправленного графа.

Вершины графа соответствуют состояниям НДКА, а ненаправленные ребра, соединяющие каждые два (пусть q_m и q_k) состояния, имеют вес равный числу переходов по непустому символу a_i из алфавита, для которых $\varphi(a_i, q_m) = \varphi(a_i, q_k)$. Ребра с наибольшим весом являются потенциальными кандидатами на переход по умолчанию при построении НДКА с задержками, вес ребра $w_{q_m q_k} - 1$ сообщает, на сколько можно сократить количество переходов при выборе в качестве перехода по умолчанию перехода между состояниями q_m и q_k .

2) Построение остовных деревьев алгоритмом Крускала с ограничением на длину δ -пути (см. далее).

При выборе графа накладывается ограничение на появления циклов, состоящих из переходов по умолчанию, и обеспечивающих бесконечность самого длинного δ -пути. Очевидно, что построенный из ненаправленного графа НДКА с задержками в общем случае не единственен, а число возможных НДКА с задержками в крайнем случае достигает числа всех направленных графов без циклов при заданном числе вершин.

Из полученных ненаправленных графов строят множество остовных деревьев с использованием алгоритма Крускала (Kruskal algorithm), требующего $O(n^2 \log n)$, итерационно при каждой вершине графа добавляя в остовное дерево ребро с максимальным весом, если дочерняя вершина уже не принадлежит остовному дереву [24]. Длина δ -пути является одним из параметров, ограничивающим эффективность сжатия и скорости работы НДКА с задержками, поэтому при построении дерева накладывается ограничение на его диаметр. Корнем дерева выбирается вершина графа, являющаяся серединой диаметра дерева, а направления переходов по умолчанию выбираются в сторону корня дерева.

Построение остовного дерева, удовлетворяющего ограничению на максимальный диаметр, является NP-трудной задачей.

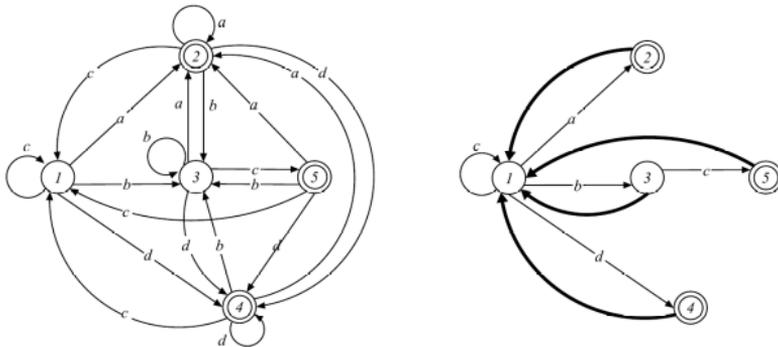


Рис. 2. Примеры НДКА (слева) и НДКА с задержками (справа), принимающих язык $L = \{a^+ \cup b^+c \cup c^*d^+\}$. Источник: [1].

Как и НДКА Ахо-Корасик с переходами в случае ошибки НДКА с задержками позволяет сократить таблицу переходов, при этом не затрагивая число состояний. Применение такого подхода авторами статьи к сокращению числа переходов, требуемого для реализации ДКА, через конструирование автомата с задержками на практике (на базах Cisco System (2008), Snort (2008), Bro NIDS (2008)) показало, что сокращение числа переходов может достигать 99 % от начального числа переходов [1, 22, 23].

2.1.4. δ -КА — НДКА с задержками и ссылкой на родительское состояние [25]

Продолжением идеи НДКА с задержками (D2FA) является сокращение таблицы переходов, и тем самым уменьшение объема необходимой памяти. Идея Фикара (Ficara) и соавторов модификации D2FA, названная δ -КА, заключается в том, что для каждого состояния автомата в таблице переходов хранятся не все значения переходов, а лишь те, которые отличаются от родительского состояния (при его наличии). Так, авторы δ -КА исходили из следующих предположений, основанных как на особенностях построения остоного дерева для D2FA, так и на эмпирических данных баз сигнатур систем обнаружения вторжений:

- большинство переходов по умолчанию НДКА с задержками имеют направление в состояния, близкие к начальному;
- большинство переходов по одному и тому же входящему символу идут в направлении одного и того же состояния автомата [25].

Авторы наглядно модифицируют пример автомата, изначально приведенный авторами НДКА с задержками (рис. 3). δ -КА обладает тем же недостатком, что и его предшественник ДКА с задержками (D2FA) — задержкой, которая обоснована тем, что при отсутствии в таблице состояний переходов по определенным символам для данного состояния, требуется перейти к участку таблицы, описывающего родительское состояние. Последнее необходимо повторять столько раз, пока в таблице переходов не встретится переход по соответствующему символу в родительском состоянии. Так, в D2FA, состояние, где требуется переход по умолчанию приводит к задержке равной количеству требуемых переходов. Для такого же состояния в δ -КА задержка будет соответствовать количеству тактов, необходимому для перехода в таблице переходов в родительское состояние, обеспечивающее данный переход по символу. Так, на рисунке 4.b для автомата D2FA переход по символу “a” из состояния 5 требует задержки в один такт, обеспечивающей переход из состояния 5 в состояние 1, и далее в состояние 2. Для автомата δ -КА на рисунке 4.c для перехода по символу “a” требуется задержка в 2 такта — в таблице состояний из состояния 5 требуется переход к записи с родительским состоянием 3, далее в состояние 1, где и хранится переход по символу “a”.

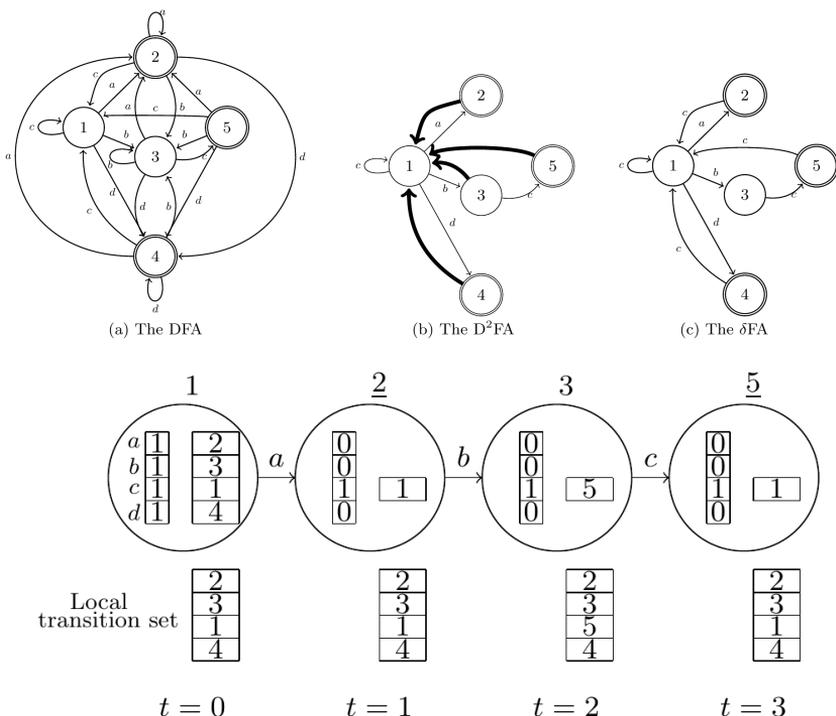


Рис. 3. Примеры ДКА (верх, а), ДКА с задержками (верх, б) и δ -КА (верх, с), принимающих язык $L = \{a^+ \cup b^+c \cup c^*d^+\}$, и таблица переходов для δ -КА (низ рисунка). Источник: [25].

2.1.5. Алгоритм пространственной и временной амортизации ДКА [4]

Авторы Микела Бечи и Патрик Кроули придумали алгоритм (в англоязычной литературе названный *Amortized time/bandwidth overhead DFAs*, а сокращенно — А-ДКА), который использует избыточность переходов распознающего регулярный язык конечного автомата. Избыточность была замечена авторами эмпирически и заключалась в том, что большая часть состояний ДКА, построенного на наборах данных сигнатурных баз систем обнаружения вторжений, имела сходные наборы исходящих переходов [4]. Ранее это было отмечено и в работе Кумара и соавторов [1]. В своей работе авторы А-ДКА усовершенствовали решение ДКА с задержками (D2FA), предложенное Кумаром и соавторами в 2006 году.

При разработке алгоритма Микела Бечи и Патрик Кроули используют глубину состояния. Теоретически алгоритм построения А-ДКА основан на леммах:

Лемма 2. *Если ни один из переходов по умолчанию в D2FA не ведет из состояния с глубиной d_i в состояние с глубиной d_j , где $d_j \leq d_i$, то любая строка длиной N потребует обработки не более $2N$ обходов состояний.*

Лемма 3. *Если все переходы по умолчанию в D2FA приводят из состояния глубины d_i к состоянию глубины d_j , где $d_j \leq d_{i-k}$, $k \in \mathbb{N}$, то для любой строки длины N потребуется не более $N(k+1)/k$ переходов для обработки.*

Таким образом, проблема построения А-ДКА в соответствии с леммами может быть сформулирована как проблема поиска максимального остовного дерева на ориентированном графе (в оригинале — sparse reduction graph) (См Algorithm 2). Максимальное остовное дерево приводит к самому высокому уровню сжатия НДКА с задержками — А-ДКА. Согласно леммам авторов корнем этого дерева всегда выбирается начальное состояние соответствующего графу ДКА. Так как при построении направление ребер обращается к корню, то в таком графе отсутствуют ориентированные циклы, и поиск максимального остовного дерева может быть осуществлен за $O(n^2)$, где n — число состояний начального ДКА [4].

Алгоритм построения А-ДКА предполагает обход ДКА в ширину, что позволяет вычислять и переходы по умолчанию, и глубину состояний за один проход. Фактически, алгоритм сводится к тому, чтобы каждое состояние переходило по переходу по умолчанию в состояние, имеющее наибольшее общее количество исходящих переходов и обладающее меньшей глубиной.

Алгоритм А-ДКА Микелы Бечи и Патрика Кроули в худшем случае использует $2N$ переходов и $N(k+1)/k$ переходов для обработки строки длиной N , где k — наперед заданное значение, $k \in \mathbb{N}$. Показано, что в общем случае А-ДКА дает улучшение сжатия по сравнению с НДКА с задержками (D2FA). В то время как НДКА с задержками (D2FA) балансирует объем и пропускную способность памяти, алгоритм А-ДКА направлен на достижение постоянной границы на уровне $2N$ в худшем случае. Фактически, $2N$ переходов сопоставимо с реализацией НДКА с задержками (D2FA) с минимальным диаметром, равным двум, что практически мало применимо. Что касается асимптотики временной сложности

Algorithm 2 Алгоритм построения А-ДКА.

1: Вход: 1) Граф $G = \{e, v, d\}$, v — вершины которого соответствуют ДКА, распознающему заданный язык, с присвоенным каждой i -ой вершине значения глубины i -ого состояния — $d(v_i)$, $e = \emptyset$ 2) Значение k , $k \in \mathbb{N}$. Состояния хранятся в виде отсортированного по возрастанию глубины состояний списка — S_d . Выход: Ориентированный граф $G = \{e, v\}$, $m(v_i, v_j)$ — число общих переходов v_i и v_j (начальное значение соответствует 0).

2: **for** v_i из S_d **do**

3: **for** v_j из S_d при условии $d(v_j) < d(v_i)$ **do**

4: **if** $d(v_i) - d(v_j) \geq k$ **then**

5: Посчитать m_{ij}

6: **if** $m(v_i, v_j) = \max(m(v_i, v_j) | v_j \in \cup_{j=1}^n v_j)$ **then**

7: Построить переход по умолчанию e_{ij}

8: **end if**

9: **end if**

10: **end for**

11: **end for**

сти, алгоритмы построения D2FA и А-ДКА имеют временные границы $O(n^2 \log_2 n)$ и $O(n^2)$ соответственно [4].

2.2. Методы сжатия с использованием группировки регулярных выражений

Данные методы объединяет стратегия не подстроиться под начальную структуру автомата, распознающего язык, а выделить в языке независимые (или условно независимые) подмножества. Данное выделение подмножеств позволяет построить ДКА для каждого подмножества и объединить полученные ДКА в единый НДКА, сокращая при этом требования к памяти.

2.2.1. Алгоритм сжатия ДКА, строящий мультиавтомат на основе группировки взаимодействующих регулярных выражений [2]

Построение мультиавтомата является принципиально другим подходом в отношении сжатия распознающего язык конечного автомата, по сравнению с алгоритмами, которые были рассмотрены выше.

Алгоритм построения мультиавтомата базируется на том, что регулярные выражения имеют общие части, и работает через выборочное группирование регулярных выражений на основе их “взаимодействия”. Регулярные выражения взаимодействуют, если ДКА, принимающий язык, являющийся объединением этих выражений, имеет больше состояний, чем сумма состояний ДКА, принимающих языки, описанные данными регулярными выражениями в отдельности [2].

Данный алгоритм ищет регулярные выражения, которые взаимодействуют, и объединяет в группы не взаимодействующие регулярные выражения. Количество групп задается до начала построения и зависит от заданного объема памяти, используемой автоматом. Поиск же наилучшей по памяти группировки регулярных выражений — NP-трудная задача. Таким образом, на основе m выделенных групп регулярных выражений строится m соответствующих ДКА. Соответствующие ДКА объединяются в один НДКА параллельно. Таким образом, мультиавтомат (multistride DFA) — это объединение нескольких ДКА, каждый из которых построен для распознавания слов, описываемых одной группой регулярных выражений, тогда как язык принимаемый мультиавтоматом, описывается суммой групп этих регулярных выражений. Группировка регулярных выражений позволяет использовать многопоточность для параллельной проверки принадлежности слова регулярному языку, используя в каждом потоке проверку по каждой выделенной группе взаимодействующих регулярных выражений. На практике, в системах обнаружения вторжений и системах анализа сетевого трафика авторами данного алгоритма достигнуто снижение необходимой памяти для реализации мультиавтомата на порядок при сравнении с классическим ДКА. Так, верхняя граница числа состояний такого мультиавтомата — $O(m2^l)$, где m — число регулярных выражений R_i , из языка $\cup_{i=1}^m R_i$, где $l = \max(|R_i|)$ [2].

2.2.2. Алгоритм сжатия ДКА, строящий мультиавтомат на основе группировки взаимодействующих регулярных выражений с оценкой взаимодействия [3]

Рохрер (Rohrer) и соавторы дополнили предыдущий жадный алгоритм оценкой степени взаимодействия регулярных выражений. Так, для каждого регулярных выражений R_i и R_j рассчитывали значение $I_{ij} = S_{ij} - S_i - S_j$, где S_i — мощность алфавита состояний ДКА, соответствующе-



Рис. 4. Строеие ДКА-мультиавтомата, состоящего из двух ДКА ($DFA1$ — соответствует регулярному выражению R_1 , $DFA2$ — соответствует регулярному выражению R_2) и принимающего язык $L = R_1 \cup R_2$. Источник: [2].

го R_i, S_j — мощность алфавита состояний ДКА, соответствующего R_j , S_{ij} — мощность алфавита состояний ДКА, соответствующего $R_i \cup R_j$.

Авторами показано, что для k групп n регулярных выражений мощность состояний мультиавтомата S_{nR_i} , распознающего $\cup_i^n R_i$ будет оцениваться как:

$$O\left(\sum_{i=1}^n S_i + \sum_{l=1}^k \sum_{i=2, i \in R_l}^n \sum_{j=1, j \in R_l}^{i-1} I_{ij}\right)$$

Задача поиска оптимального разбиения сводится к задаче минимизации S_{nR_i} , с учетом фиксированной $\sum_{i=1}^n S_i$. Таким образом, задача остается NP -трудной. Учитывая, что число возможных разбиений на группы — $O(k^n)$, время поиска разбиения достаточно большое для практического применения [3].

2.2.3. Алгоритм сжатия ДКА на основе группировки взаимодействующих регулярных выражений с оценкой вклада в экспоненту [26]

Данный жадный алгоритм является продолжением предыдущего и также базируется на взаимодействии регулярных выражений. Авторы сводят проблема группировки регулярных выражений к NP -трудной задаче максимального разбиения графа на k частей и предлагают эвристический алгоритм, называемый одношаговым жадным алгоритмом сжатия

ДКА, для решения этой NP -трудной задачи. Идея алгоритма и его название возникли из наблюдения того, что не все регулярные выражения вносят одинаковый вклад в экспоненту. Как мы уже упоминали, особое внимание в проблеме экспоненциального взрыва при построении конечного автомата, соответствующего множеству регулярных выражений, уделяется использованию в регулярных выражениях сочетаний “.+” и “.*” и “считающих” выражений, типа “ $\{n\}$ ” и “ $[a_i]\{n\}$ ”. Авторы предложили использовать средний вклад выражения в проблему экспоненциального взрыва и сформулировали следующую EP_i метрику для регулярных выражений:

$$EP_i = \frac{1}{n} \sum_{j=1}^n \frac{I_{ij}}{S_j}, i \neq j,$$

где используются обозначения предыдущего раздела. Чем больше значение EP_i , тем больше вклад данного регулярного выражения в проблему экспоненциального взрыва [26].

Авторы представили инициальный алгоритм распределения регулярных выражений по группам на основе полученных для каждого регулярного выражения значения EP . Алгоритм заключается в нисходящей сортировке регулярных выражений в отношении значений EP и последующем добавлении регулярного выражения с минимальным значением EP в группу с суммарным для группы значением EP (см. Algorithm 3). Авторами показано, что данный инициальный алгоритм быстрее для дальнейшей сортировки по группам, чем случайный выбор группы для регулярного выражения. Далее для распределения по группам авторы представили одношаговый жадный алгоритм, который перемещает регулярное выражение в ту группу, которая дает наибольший суммарный выигрыш для задачи минимизации числа состояний автомата, распознающего язык, представленный данными регулярными выражениями из $\cup_{i=1}^n R_i$ [26].

3. Заключение

В направлении решения задачи экспоненциального взрыва без добавления специальных структурных элементов особо успешными были некоторые идеи и алгоритмы, ставшие прародителями целых направлений в сигнатурном анализе. Во-первых, к таким находкам относится алгоритм

Algorithm 3 Жадный алгоритм сжатия ДКА на основе группировки взаимодействующих регулярных выражений

```
1: Начальное состояние формируется инициальным алгоритмом с оценкой вклада регулярного выражения в экспоненту. Считается начальное значение  $C_0$ , равное мощности алфавита состояний начального ДКА, полученного после работы инициального алгоритма.  
2: for  $R_i$  из  $\cup_{i=1}^n R_i$  do  
3:   for группа  $G_j$  из  $\cup_{j=1}^k G_j$  do  
4:     Рассчитать  $C_{ij}$  для  $R_i \in G_j$   
5:     Сохранить  $C_{ij}$   
6:   end for  
7: end for  
8: for  $R_i$  из  $\cup_{i=1}^n R_i$  do  
9:   if  $C_{ij} = \max \cup_{i=1}^n \cup_{j=1}^k C_{ij}$  then  
10:    if  $R_i$  не в  $G_j$  then  
11:      Переместить  $R_i$  в  $G_j$   
12:      Перейти на шаг 2  
13:    else  
14:      Остановка  
15:    end if  
16:  end if  
17: end for
```

Ахо-Корасик с переходами в случае ошибки для поиска подстрок, появление которого породило целую плеяду алгоритмов. Во-вторых, идея построения мультиавтомата на основе группировки взаимодействующих регулярных выражений, которая инициировала направление параллельных вычислений для реализации поиска по сигнатурам. Однако несмотря на приличное число идей оптимизации, алгоритмы группировки взаимодействующих регулярных выражений остаются NP -полной задачей и не гарантируют решение проблемы экспоненциального взрыва в общем случае. Алгоритмы, использующие переходы в случае ошибки, также не решают проблему экспоненциального взрыва в общем случае, тем не менее позволяя значительно сократить потребности в памяти для реализации поиска по сигнатурам на практике.

Список литературы

- [1] Kumar S., Chandrasekaran B., Yu F., Crowley P. Turner J., “Algorithms to Accelerate Multiple Regular Expressions Matching for Deep Packet Inspection”, *ACM SIGCOMM Computer Communication Review*, **36**:4 (2006), 155–164.
- [2] Yu F., Chen Zh., Diao Y., Lakshman T.V., Katz R.H., “Fast and memory-efficient regular expression matching for deep packet inspection”, *Proceedings of the 2006 ACM/IEEE symposium on Architecture for networking and communications systems*. — ACM., 2006, 93-102.
- [3] Rohrer J., Atasu K., van Lunteren J., Hagleitner C., “Memory-efficient distribution of regular expressions for fast deep packet inspection.”, *Proceedings of the 7th IEEE/ACM international conference on Hardware/software codesign and system synthesis*. — ACM., 2009, 147-154.
- [4] Becchi M., Crowley P., “A-DFA: A Time- and Space-Efficient DFA Compression Algorithm for Fast Regular Expression Evaluation”, *Transactions on Architecture and Code Optimization*. — ACM., **10**:1 (2013), 4.1-4.4.
- [5] Liu C. and Wu J., “Fast Deep Packet Inspection with a Dual Finite Automata.”, *IEEE Transactions on Computers.*, **62**:2 (2013), 310-321.
- [6] Bernadotte A., Kumar R., Winblad B., Pavlov P.F., “In silico identification and biochemical characterization of the human dicarboxylate clamp TPR protein interaction network”, *FEBS Open Bio*, **8** (2018), 1830–1843.
- [7] Бернадотт А., Галатенко А.В., “Аппаратная конструкция для решения проблемы экспоненциального взрыва для одного класса регулярных языков”, *Интеллектуальные системы. Теория и приложения*, **23**:4 (2019), 121–146.
- [8] Хопкрофт Дж., Мотвани Р., Ульман Дж., *Введение в теорию автоматов, языков и вычислений*, Вильямс, Москва, 2017.
- [9] Rabin M.O., Scott D., “Finite automata and their decision problems”, *IBM J. Research and Development*, **3**:2 (1959), 115–125.
- [10] Кудрявцев В. Б., Алешин С. И., Подколзин А. С., *Введение в теорию автоматов*, Наука, Москва, 1985.
- [11] Кудрявцев В. Б., Гасанов Э. Э., Подколзин А. С., *Основы теории интеллектуальных систем*, Макс Пресс, Москва, 2016.
- [12] Лупанов О.Б., “О сравнении двух типов конечных источников”, *Проблемы кибернетики*, 1963, № 9, 321–326.
- [13] Документация для Perl-совместимых регулярных выражений, <http://perldoc.perl.org/perlre.html>.
- [14] Александров Д. Е., “Об оценках автоматной сложности распознавания класса регулярных языков”, *Интеллектуальные системы. Теория и приложения*, **18**:4 (2014), 121–146.
- [15] Александров Д. Е., “Об уменьшении автоматной сложности за счет расширения регулярных языков”, *Программная инженерия*, 2014, № 11, 26–34.
- [16] Александров Д. Е., “Об оценках мощности некоторых классов регулярных языков”, *Дискретная математика*, **27**:2 (2015), 3–21.

- [17] Smith R., Estan C., Jha S., “XFA: Faster Signature Matching with Extended Automata”, IEEE Symposium on Security and Privacy, 2008, 187-201.
- [18] Александров Д. Е., “Эффективные методы реализации проверки содержания сетевых пакетов регулярными выражениями”, *Интеллектуальные системы. Теория и приложения*, **18:1** (2014), 37–60.
- [19] Александров Д. Е., *Сложность распознавания принадлежности слова регулярному языку в системах обнаружения вторжений*, Диссертация на соискание ученой степени кандидата физико-математических наук, Москва, 2015.
- [20] Aho A.V., Corasick M.J., “Efficient string matching: an aid to bibliographic search”, *Communication of the ACM*, **18:6** (1975), 333-340.
- [21] De La Briandais R., “File Searching Using Variable Length Keys”, *Proceedings of Western Joint Computer Conference.* — ACM., 1959, 295-298.
- [22] Kumar S., Chandrasekaran B., Turner J., Varghese G., “Curing regular expressions matching algorithms from insomnia, amnesia, and acalculia”, ANCS '07 Proceedings of the 3rd ACM/IEEE Symposium on Architecture for networking and communications systems , 2008, 155–164.
- [23] Kumar S., *A Thesis on Acceleration of Network Processing Algorithms*, Doctor of Science Thesis, Washington University, Saint Louis, Missouri, 2008.
- [24] Kruskal J. B., “On the shortest spanning subtree of a graph and the traveling salesman problem”, *Proceedings of the American Mathematical Society*, **7:1** (1956), 48-50.
- [25] Ficara, D., Giordano, S., Procissi, G., “An improved DFA for fast regular expression matching”, *ACM SIGCOMM Computer Communication Review*, **38:5** (2008), 29–40.
- [26] Xu C., Su J., Chen S., “Exploring efficient grouping algorithms in regular expression matching”, *PLoS ONE*, **13:10** (2018), e0206068.

Finite automaton modification by using compression algorithms Bernadotte A.

Аннотация

The question of whether a word belongs to a given regular language finds its application in areas where it is relevant to search for certain patterns in data of different nature. Growth of the deterministic finite automaton (DFA) states number which is exponential in number of regular expressions of the given language is still a real problem. In this article, we take a look at modifying a finite automaton by using compression algorithms. These approaches modify the finite automaton without changing the language and without adding additional structural elements.

Keywords: DFA, NDF, regular language, exponential blowup, compression algorithm.

О синтезе колонии жуков с линейным ростом

Воротников А.С.¹

Рассматривается динамическая система, описывающая поведение популяции жуков. Жуки живут на поле, которое в начальный момент представляет собой целочисленную решётку, к каждой клетке которой в начальный момент находится одинаковое количество еды для жуков. Жуки в соответствии с некоторым алгоритмом перемещаются по полю, едят расположенную в нём еду и размножаются, причём на все действия расходуется энергия. Система моделируется однородными структурами. В работе показано, что для любой прямой из некоторого класса существует колония, чей линеаризованный график численности бесконечное число раз пересекает выбранную прямую.

Ключевые слова: автоматное моделирование биологической системы, скорость роста динамических систем, клеточный автомат.

1. Введение

Рассмотрим следующую динамическую систему: дано бесконечное поле с ненулевым однородно расположенным запасом еды. На этом поле появляется жук, который перемещается по полю, ест имеющуюся еду и размножается. Поле моделируется целочисленной решёткой на плоскости, в которой каждому узлу в начальный момент сопоставлено некоторое одинаковое количество еды. Эту целочисленную решётку в дальнейшем будем называть картой. Жук действует по алгоритму, который упрощённо представляется схемой: искать еду, есть, пока не насытишься (если еды не хватило, снова искать), размножатся. На все действия расходуется энергия, которая получается жуком из еды, расположенной

¹*Воротников Алексей Сергеевич* — студент каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: vorotnikov.lexa@yandex.ru.

Vorotnikov Alexey Sergeevich — student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

на поле. Если запас энергии жука падает ниже ноля, то жук умирает — исчезает с поля. Под размножением подразумевается деление жука на двух жуков, обладающих половиной запаса энергии родителя.

Хочется выяснить поведение функции численности популяции жуков во времени.

Каждого жука можно интерпретировать как рой или колонию. К биологическим видам, образующим подобные структуры относятся: вольвокс[1], хоанофлагеллаты[2], хроококковые. Моделирование биологических систем активно развивается со времён публикаций работ Лотки[3] и Вольтерры [4]. Чаще всего применяется подход моделирования дифференциальными уравнениями.

В данной работе предлагается использовать подход автоматного моделирования. Такой подход применялся ранее [5]. Рассматриваемый объект родственен клеточным автоматам [6, 7, 8]. В данный момент ведутся активные исследования в области автоматов и однородных структур [9, 10, 11]

В данной работе построен конкретный класс колоний, для которого показывается, что для любой растущей прямой, чей тангенс угла наклона меньше $\frac{40}{33}$ существует такая колония из этого класса, что её линеаризованный график численности популяции бесконечное число раз пересекает выбранную прямую.

В дальнейшем планируется построить колонии, покрывающие прямые вплоть до вертикальных.

Для любой колонии существует тривиальная квадратичная верхняя оценка численности популяции, но вопрос существования колоний с более чем линейной скоростью роста остаётся открытым.

Автор выражает глубокую благодарность научному руководителю д.ф.-м.н., профессору Э.Э. Гасанову за постановку задачи и помощь в работе.

2. Основные понятия и формулировка результатов

Пусть $S \in \mathbb{Q}_+ = \{x \in \mathbb{Q} | x \geq 0\}$.

Картой будем называть множество $\Sigma = \mathbb{Z}^2 \times \{0, S\} \times (\mathbb{N} \cup \{0\})$.

Проще говоря, карта — это разлинованная на клетки плоскость. Первая компонента — это координаты клетки. Вторая компонента — это количество еды в клетке, т.е. в каждой клетке либо нет еды, либо есть

еда, соответствующая запасу энергии S . Последняя компонента — это количество жуков, находящихся в клетке.

Сначала опишем жука неформально.

Будем считать, что имеются 2 параметра, которые свойственны любому жуку:

- $V \in \mathbb{Q}_+$ — расход энергии на переход в соседнюю клетку;
- $D \in \mathbb{Q}_+$ — порог энергии, при превышении которого жук обязан разделиться.

Будем считать, что состояние каждого жука характеризуется четвёркой (x, y, E, d) , где x, y — абсцисса и ордината текущей клетки, в которой находится жук, $E \in \mathbb{Q}$ — текущий запас энергии, $d \in \{n, e, s, w\}$ — текущее направление жука. Здесь n, e, s, w задают направления на север, восток, юг, запад соответственно.

Каждый такт жук обязан совершить одно из следующих действий:

- умереть, т.е. исчезнуть с карты;
- съесть поровну с другими жуками, находящимися вместе с ним на одной клетке, всё содержимое клетки, это действие назовём *едой*;
- передвинуться в соседнюю по стороне клетку, это действие назовём *движением*;
- разделиться на двух одинаковых по энергии жуков, это действие назовём *делением*.

Каждый такт жуки функционируют по следующему правилу: сначала все выбирают действие, которое совершат, затем совершают это действие в порядке, указанном выше (сначала все, кто решили «умереть» умирают, затем все, кто решил «есть», едят и т.д.), если необходимо, изменяют состояние карты во время своего действия.

Пусть t — текущий момент времени, $(x(t), y(t), E(t), d(t))$ — текущее состояние некоторого жука. Выбор действия осуществляется по следующим правилам.

- 1) Если $E(t) < 0$, то жук выбирает действие «умереть».
- 2) Если $0 \leq E(t) < D$ и в клетке, в которой находится жук, есть еда, то есть вторая компонента ненулевая, жук выбирает действие «еда».

- 3) Если $0 \leq E(t) < D$ и клетка, в которой находится жук, пустая, т.е. в ней нет еды, то жук проверяет четыре соседние по стороне клетки в таком порядке: передняя, левая, задняя, правая. Направление считается относительно текущего направления жука $d(t)$. Если среди этих клеток есть клетка с едой, то жук поворачивается к ней (т.е. выбирает новое направление движения d') и переходит в состояние «движение». Если все соседние клетки пустые, то жук переходит в состояние «движение» с направлением движения $d' = d(t)$.
- 4) Если $E(t) \geq D$, то жук переходит в состояние «деление».

После выбора действие совершается следующим образом.

- 1) Если выбрано действие «умереть», то сначала третья компонента клетки уменьшается на единицу, а затем жук убирается с карты.
- 2) Если выбрано действие «еда», и в клетке $(x(t), y(t))$ вместе с данным жуком находятся всего m жуков, то состояние жука изменяется на $(x(t+1), y(t+1), E(t+1), d(t+1))$, где $x(t+1) = x(t)$, $y(t+1) = y(t)$, $E(t+1) = E(t) - 1 + S/m$, где m — значение третьей компоненты клетки, $d(t+1) = d(t)$. После действия вторая компонента клетки обнуляется, то есть еда в клетке исчезает.
- 3) Если выбрано действие «движение» и d' — направление движения, то $d(t+1) = d'$, $E(t+1) = E(t) - 1 - V$ а координата жука изменяется по правилу: если $d' = n$, то $x(t+1) = x(t)$, $y(t+1) = y(t) + 1$; если $d' = s$, то $x(t+1) = x(t)$, $y(t+1) = y(t) - 1$; если $d' = e$, то $x(t+1) = x(t) + 1$, $y(t+1) = y(t)$; если $d' = w$, то $x(t+1) = x(t) - 1$, $y(t+1) = y(t)$.
- 4) Если выбрано действие «деление», состояние жука изменяется на $(x(t+1), y(t+1), E(t+1), d(t+1))$, где $x(t+1) = x(t)$, $y(t+1) = y(t)$, $E(t+1) = (E(t))/2 - 1$, $d(t+1) = d(t)$. При этом на карте появляется один новый жук, состояние которого будет $(x'(t+1), y'(t+1), E'(t+1), d'(t+1))$, где $x'(t+1) = x(t)$, $y'(t+1) = y(t)$, $E'(t+1) = (E(t))/2 - 1$ а $d'(t+1)$ вычисляется по следующему правилу: если $d(t) = n$, то $d'(t+1) = w$; если $d(t) = s$, то $d'(t+1) = e$; если $d(t) = e$, то $d'(t+1) = n$; если $d(t) = w$, то $d'(t+1) = s$, т.е. потомок «смотрит» влево по отношению к родителю. При этом новый жук начинает жить со следующего такта. Третья компонента клетки увеличивается на единицу.

В разделе 3 будет показана корректность определения.

Потомками жука будем называть всех жуков, появившихся на карте в результате деления данного жука.

Однородной картой будем называть карту, у которой все вторые компоненты одинаковые ненулевые.

Колонией назовём совокупность: карта с запасом энергии S и один жук с параметрами (E, V, D) . Будем обозначать такую колонию $K = (S, E, V, D)$. В дальнейшем, если не указано иное, рассматриваются колонии на однородных картах. Тогда без ограничения общности состояние жука перед первым тактом характеризуется четвёркой $(0, 1, E, n)$, то есть находится в клетке с координатами $(0, 1)$, обладает запасом энергии E и смотрит вверх. Все случаи сводятся к такому поворотами на $\frac{\pi}{2}$ и целочисленными сдвигами.

В дальнейшем будем рассматривать колонии вида $K = (S, 0, V, D)$ и обозначать такие колонии $K = (S, V, D)$. Таким образом нивелируется влияние начального состояния на последующую жизнь колонии.

Назовём *численностью популяции* в момент t — число жуков $M(t)$ на карте после t тактов.

Назовём *функцией численности* $N(t): \mathbb{R}_+ \rightarrow \mathbb{R}_+$, такую, что $N(t) = M(t)$ для любого целого t . В остальных точках $N(t)$ линейно интерполирует $M(t)$, то есть функция численности является линейризованной численностью популяции.

Теорема 1. *Для произвольной прямой из класса $\mathcal{A} = \{y = ax + b \mid 0 < a < \frac{40}{33}, b \in \mathbb{R}\}$ существует такая колония, что график её функции численности бесконечное число раз пересечёт выбранную прямую.*

3. Вспомогательные утверждения

Если хоть один жук в клетке выбрал действие «еда», то все жуки в этой клетке выбрали или действие «умереть» или действие «еда».

Доказательство. Отметим, что в клетке есть еда, иначе ни один жук не выбрал бы действие «еда».

Отметим так же, что если жук выбрал действие «умереть», то он исчез с карты до выполнения других действий остальными жуками.

- Пусть какой-то жук выбрал действие «движение». Это могло произойти, только если он не нашёл еды в клетке, выше мы отметили, что еда в клетке есть, получено противоречие.

- Пусть какой-то жук выбрал действие «деление», тогда он должен был на предыдущем такте выбрать действия «еда» или «деление»: действие «движение» приводит к тому, что на предыдущем такте у жука было больше энергии, чем на текущем, значит он должен был разделиться на предыдущем такте, что приводит к противоречию. Если было выбрано действие «еда», то энергии в клетке уже нет, получено противоречие с предположением. Если было выбрано действие «деление», то рассмотрим самое первое деление этого жука на этой клетке, тогда мы приходим к рассмотренному выше случаю, когда перед действием «деление» жук выбирает действие «еда», что снова приводит к противоречию.

□

Рассмотрим колонию $K = (6k + 5, 6k - 3, 12k - 7)$, где k — чётный натуральный параметр не меньше 16. Для доказательства нам потребуется несколько лемм.

Лемма 1. *Произвольный жук колонии K с нулевым запасом энергии в момент времени t_0 делится после прохождения ровно через k клеток, если после съедения каждой клетки жуку досталось ровно $6k + 5$ единиц энергии. Тогда после деления и перехода в соседнюю клетку, то есть в момент времени $t_0 + 2k + 1$, энергия жука равна нулю.*

Доказательство. Для доказательства первой половины утверждения достаточно показать, что

$$E(t_0 + 2k - 3) < D \leq E(t_0 + 2k - 1). \quad (1)$$

Это означает, что после прохождения $k - 1$ клетки у жука будет недостаточно энергии для деления, а после прохождения k клеток энергии уже хватит.

$$\begin{aligned} E(t_0 + 2k - 3) &= (k - 1)(S - V - 2) + V + 1 = \\ &= (k - 1)(6k + 5 - 6k + 3 - 2) + 6k - 3 + 1 = 12k - 8, \\ E(t_0 + 2k - 1) &= k(S - V - 2) + V + 1 = 12k - 2, \\ D &= 12k - 7. \end{aligned}$$

Слагаемое $V + 1$ нужно, так как жук совершает на одно движение меньше, чем съедает клеток. Тем самым неравенство (1) доказано.

До деления запас энергии жука равен

$$E(t_0 + 2k - 1) = 12k - 2,$$

после деления он составит

$$E(t_0 + 2k) = \frac{k(S - V - 2) + V + 1}{2} - 1 = \frac{12k - 2}{2} - 1 = 6k - 2.$$

После перехода в соседнюю клетку останется

$$\begin{aligned} E(t_0 + 2k + 1) &= \frac{k(S - V - 2) + V + 1}{2} - 1 - V - 1 = \\ &= 6k - 2 - 6k + 3 - 1 = 0. \end{aligned}$$

□

Лемма 2. *Если жук обязан пройти через клетку с нулевым запасом энергии, то он умрёт.*

Доказательство. Жук не может выбрать действие «движение», обладая энергией больше D . Чтобы перейти через пустую клетку жук должен затратить $2V + 2$ энергии. Поскольку

$$D = 12k - 7 < 12k - 4 = 2(6k - 3 + 1) = 2(V + 1),$$

то жук умрёт.

□

Лемма 3. *Если существует жук с нулевым запасом энергии, который съедает $\frac{k}{2} < p \leq k - 1$ клеток, получая с каждой клетки $6k + 5$ единиц энергии, затем съедает клетку так, что получает $3k + 2.5$ единицы энергии (делит клетку с другим жуком), и после этого он съест ещё $k - 1 - p + \frac{k}{2}$ клеток, получая с каждой $6k + 5$ единиц энергии, то спустя $k + \frac{k}{2}$ клеток после начала движения жук разделится, перейдёт в соседнюю клетку и умрёт.*

Доказательство. Без ограничения общности начальный момент времени примем равным нулю. В нулевой момент времени жук находится на клетке с ненулевым запасом энергии, иначе он обязан сделать шаг и умереть. Тогда после съедения p клеток энергия жука составит $E(2p - 1) = p(S - V - 2) + V + 1 = 6p + 6k - 2$, что меньше D — это очевидно из доказательства Леммы 1. В момент времени $2p + 1$ жук съест клетку

Рис. 1. Схема устройства карты Σ . Здесь и далее чёрные клетки — клетки с нулевым запасом энергии, белые с $S \neq 0$.

поровну с другим жуком, $E(2p+1) = p(S-V-2) + \frac{S}{2} - 1 = 6p+3k-1.5$, затем перейдёт в соседнюю клетку: $E(2p+2) = p(S-V-2) + \frac{S}{2} - V - 2 = 6p-3k-0.5 > 0$ по условию леммы. Затем жук пройдёт оставшиеся $k-1-p+\frac{k}{2}$ клеток: $E(2(k-1+\frac{k}{2})-1) = (k-2+\frac{k}{2})(S-V-2) + \frac{S}{2} - 1 = 6k-12+3k+3k+1.5 = 12k-10.5 < D$, но $E(2(k-1+\frac{k}{2})+1) = (k-1+\frac{k}{2})(S-V-2) + \frac{S}{2} - 1 = 6k-6+3k+3k+1.5 = 12k-4.5 > D$. После деления у жука останется $E(2(k-1+\frac{k}{2})+2) = 6k-2.25$ единиц энергии. После перехода в соседнюю клетку количество энергии составит $E(2(k-1+\frac{k}{2})+3) = -0.75$ значит, жук умрёт. \square

Лемма 4. Если карта Σ строится по следующему правилу (см. Рисунок ??):

- в клетке $(0, 0)$ нулевой запас энергии;
- вне квадрата со стороной $k-1$ и нижним левым углом в $(0, 0)$ у всех клеток нулевой запас энергии;
- внутри квадрата со стороной $k-1$ и нижним левым углом в $(0, 0)$ у всех клеток, кроме клетки $(0, 0)$ запас энергии составляет $6k+5$;

то функция численности от времени колонии $K = (6k+5, 12k-2, 6k-3, 12k-7)$, состоящей из карты Σ и жука с состоянием перед первым тактом $(0, 0, 12k-2, e)$, выглядит следующим образом:

$$g(t) = \begin{cases} 1, & \text{при } t = 1; \\ 2, & \text{при } t \in [1, 2k+1]; \\ 4, & \text{при } t \in [2k+2, 4k-4]; \\ 2, & \text{при } t \in [4k-3, 4k]; \\ 1, & \text{при } t \in [4k+1, 5k+2]; \\ 2, & \text{при } t \in [5k+3, 5k+4]; \\ 0, & \text{иначе,} \end{cases}$$

где k — чётный натуральный параметр не меньше 16.

Рис. 2. Траектория движения первого жука проведена жирной сплошной линией. Траектория потомка от его первого деления — штрихпунктирной линией, от второго — тонкой сплошной линией. Траектория движения “внука” первого жука (потомка потомка от его первого деления) обозначена тонкой штрихпунктирной линией. Стрелки условно указывают на клетку, в которой каждый конкретный жук умрёт.

Доказательство. В первый такт жук разделится: $E(1) = \frac{12k-2}{2} - 1 = 6k - 2$, сам жук и его потомок перейдут в соседние клетки $E(2) = 6k - 2 - V - 1 = 6k - 2 - 6k + 3 - 1 = 0$. Затем по Лемме 1 пойдут вдоль смежных сторон квадрата до противоположных углов, повернут, пройдут ещё две клетки, а затем делятся в момент времени $1 + 2k + 1 = 2k + 2$. После этого они проходят $k - 5$ клеток до последнего нетронутого угла квадрата, куда приходят одновременно в момент времени $2k + 2 + 2(k - 5) + 1 = 4k - 7$. Жуки поровну съедают квадрат $E(4k - 6) = (k - 5)(S - V - 2) + \frac{S}{2} - 1 = 6k - 30 + 3k + 2.5 - 1 = 9k - 28.5$. В это момент все четыре клетки, смежные с той, на которой они находятся, уже пустые, а значит жуки перемещаются на пустую клетку $E(4k - 5) = 9k - 28.5 - 6k + 3 - 1 = 3k - 26.5$. По условию леммы $3k > 26.5$, значит жуки сделают ещё один шаг, но затем умрут: $E(4k - 4) = 3k - 26.5 - 6k + 3 - 1 = -3k - 24.5 < 0$. Теперь обратим внимание на потомков. Они сделают $k - 4$ шага, после чего оба окажутся на одной клетке в момент времени $4k - 7$. По Лемме 3 при $p = k - 5$ эта ситуация гарантирует смерть обоих жуков не более чем через $\frac{k}{2} + 4$ клетки. После прохождения клетки, на которой жуки “столкнулись”, их энергия составит $E(4k - 6) = 9k - 28.5$, потомок жука, первым появившегося в квадрате, съест одну клетку и окажется перед съеденной клеткой. Он повернёт налево, там находится одна единственная клетка. Жук съест эту клетку и вынужден будет сделать шаг на пустую клетку, а значит умрёт в момент времени $4k + 1$. Второй из рассматриваемых потомков в момент времени $4k - 5$ окажется в начале полосы из клеток с запасом энергии S длиной $k - 4$ клетки. Пройдя ровно $\frac{k}{2} + 4$ клетки он разделится, он и его потомок перейдут в соседние клетки в момент времени $4k - 5 + 2(\frac{k}{2} + 4) - 1 + 1 + 1 = 5k + 4$ и там умрут $E(5k + 5) = -0.75$. Требование $k \geq 16$ гарантирует, что у жука будет необходимое количество клеток: $k - 4 \geq \frac{k}{2} + 4$. \square

4. Доказательство Теоремы 1

Докажем следующее утверждение: существует момент времени t_0 такой, что для любого $t > t_0$ график функции численности колонии, колеблется со всё возрастающей амплитудой, но постоянным периодом в конусе, образованном прямыми $y = \frac{40(x-(k-1))}{2k+1} - 196$ и $y = \frac{20(x-(2k-1))}{2k+1}x - 78$, причём как верхняя, так нижняя граница конуса достигаются бесконечное число раз.

Рассмотрим поведение колонии K . Первый жук, пройдя k клеток за $2k - 1$ тактов времени, делится. Далее первый жук повторяет те же действия до бесконечности. Для всех его потомков верно то же самое. Но только самый первый потомок первого потомка — жук, появившийся в результате деления на $4k + 1$ -ом такте — будет жить так же; остальные, выбрав направление s , через $k - 1$ клетку встретят перед собой проеденную полоску и повернут в направлении e . Таким образом в ограниченные проеденными линиями квадраты, состоящие из ещё не тронутых клеток, попадут жуки. Жизнь жуков в таких квадратах рассмотрена в Лемме 4, обратим внимание на поведение остальных жуков. А именно, рассмотрим первого потомка первого потомка первого потомка — того самого жука, что появится на карте в момент времени $6k + 2$ и разделится первый раз в клетке $(0, 0)$. На Рисунке 4 рассматриваемый жук обозначен A . Жук B появляется в момент времени $6k + 2$ на клетке $(-k, 0)$. Потомок от деления жука A в клетке $(0, 0)$ в момент времени $8k + 3$, обозначенный на Рисунке 4 C , выберет направление s , и, пройдя через $k - 1$ клетку, попадёт на одну клетку с потомком жука B . Потомок жука B появился в момент времени $8k + 3$ на клетке $(-k, -k)$ — на Рисунке 4 он обозначен D .

На Рисунке 4 появляется ещё один новый жук E . Он появился на карте вследствие деления жука B в момент времени $10k + 4$ на клетке $(-k, -2k)$. По Лемме 3 при $p = k - 1$ жуки C и D после "столкновения" в клетке $(0, -k)$ в момент времени $10k + 3$ пройдут ещё $\frac{k}{2}$ клеток, разделятся и умрут. Когда потомок от деления жука E в момент времени $12k + 5$ на клетке $(0, -2k)$ дойдёт до съеденной жуком C клетки $(0, -\frac{3}{2}k)$, он повернёт налево. Дальнейшее его поведение нас не интересует, так как его жизнь и жизнь его потомков в квадрате ограничена по времени константой $3(k - 1)^2$, где 3 — верхняя оценка на количество тактов, необходимое для перемещение жука в соседнюю клетку (достигается при делении жука, иначе нужно 2 такта), а количество клеток равно площади квадрата. То же самое произойдёт и с потомком второго потомка жука E — жуком,

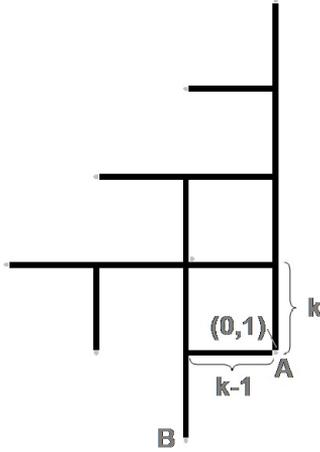


Рис. 3. Рисунок соответствует моменту времени $8k + 1$.

появившемся в клетке $(k, -k)$ в момент времени $16k + 7$ — с той лишь разницей, что он повернёт вниз после обнаружения клетки, съеденной жуком D . Теперь обратим внимание на то, что дальше все потомки жука E будут поворачивать налево, в квадрат, ограниченный предыдущим потомком жука E , потомком предыдущего потомка жука E и полоской, съеденной жуком A . Одновременно будет запускаться жизнь в двух квадратах, такая же, как во все других квадратах на карте.

Поведение жуков B и A аналогично поведению первого жука.

Теперь всё готово для явного описания функции численности колонии K . Рассмотрим жуков вне квадратов и тех, которые только вошли, но ещё не разделились. После такта с действием «деление» жука рассматриваем в рамках рассмотрения квадратов. В третьей четверти (если разбить плоскость на условные четверти: рисунок (4)) жуки живут аналогично жизни во второй с задержкой на $2k + 1$ такт. В первой четверти задержка составляет $3(2k + 1)$, а в четвёртой жуки живут так же как во второй с задержкой $4(2k + 1)$; ещё существуют четыре жука, порождённые поведением жука E с рисунка (4). Итого

$$2 \left\lfloor \frac{t}{2k+1} \right\rfloor - 2 + 2 \left(\left\lfloor \frac{t}{2k+1} \right\rfloor - 1 \right) - 2 + 2 \left(\left\lfloor \frac{t}{2k+1} \right\rfloor - 4 \right) - 2 + \\ + 2 \left(\left\lfloor \frac{t}{2k+1} \right\rfloor - 3 \right) - 2 + 4 = 8 \left(\left\lfloor \frac{t}{2k+1} \right\rfloor - 2, 5 \right). \quad (2)$$

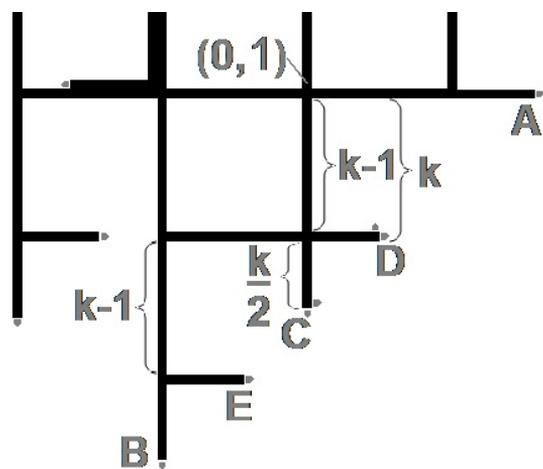


Рис. 4. Рисунок соответствует моменту времени $11k + 7$.

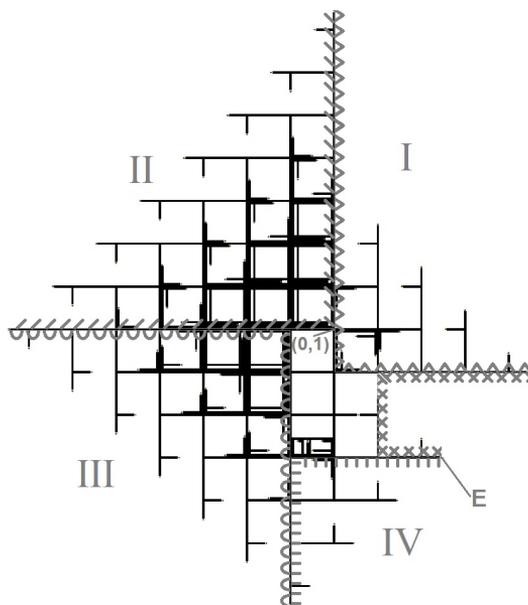


Рис. 5. Схема разбиения плоскости на четверти.

Теперь рассмотрим несколько квадратов, точнее тот случай, который реализуется в этой системе — есть n квадратов, в которых жизнь только началась, но есть ещё $n - 1$, $n - 2$ и, возможно, меньшее количество квадратов, в которых жизнь ещё не закончилась. Численность популяции жуков в квадратах во второй четверти выглядит так:

$$N_{II}(t) = \sum_{m=1}^{\infty} mg(t - (2k + 1)(m + 2) - 2k + 1),$$

где $g(t)$ — численность популяции жуков в одном квадрате. Первый квадрат появляется только через $6k + 3 + 2k - 1 = 8k + 2$ такта. С учётом озвученных выше задержек можем так же выписать аналогичные функции для всех четвертей:

$$N_{III}(t) = \sum_{m=1}^{\infty} mg(t - (2k + 1)(m + 3) - 2k + 1),$$

$$N_{IV}(t) = \sum_{m=1}^{\infty} mg(t - (2k + 1)(m + 6) - 2k + 1),$$

$$N_I(t) = \sum_{m=1}^{\infty} mg(t - (2k + 1)(m + 5) - 2k + 1).$$

Сейчас это суммы функций с пересекающимися носителями, то есть множествами, на которых функция не равна нулю; преобразуем их. Из длины носителя $5k + 4$ и величины сдвига $2k + 1$ ясно, что пересекаются носители только трёх соседних функций.

Становится понятно, что можно рассматривать интервал $(4k + 2, 6k + 3]$, до и после него всё повторяется аналогично. Запишем новую функцию $f_n(t)$, с носителем $(4k + 2, 6k + 3]$, предполагая, что в четверти $n - 2$ квадрата соответствуют чёрному графику, $n - 1$ квадрат соответствует графику, изображённым редким пунктиром, и n квадратов соответствует графику, изображённому более частым пунктиром, на рисунке (4). Для простоты сдвинем аргумент на $4k + 2$.

$$f_n(t) = \begin{cases} n - 2 + 4(n - 1) + 2n & = 7n - 6, \text{ при } t \in [1, k]; \\ 2(n - 2) + 4(n - 1) + 2n & = 8n - 8, \text{ при } t \in [k + 1, k + 2]; \\ 4(n - 1) + 2n & = 6n - 4, \text{ при } t \in [k + 3, 2k - 5]; \\ 2(n - 1) + 2n & = 4n - 2, \text{ при } t \in [2k - 4, 2k - 1]; \\ (n - 1) + 2n & = 3n - 1, \text{ при } t \in [2k, 2k + 1]. \end{cases}$$

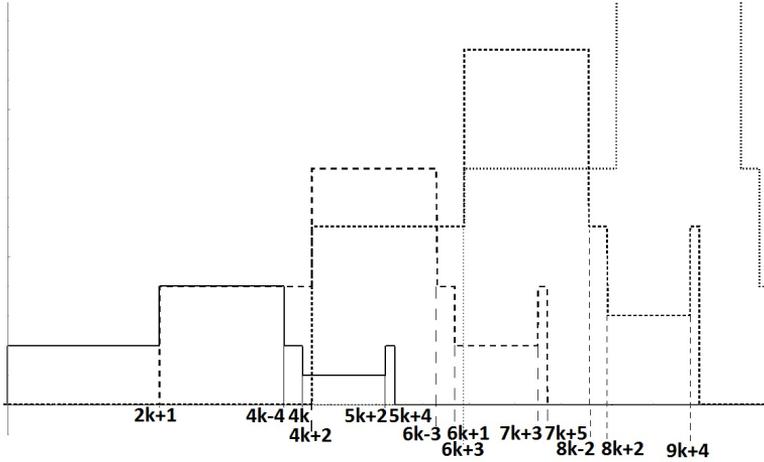


Рис. 6. Графики $g(t)$, $2g(t - (2k + 1))$, $3g(t - 2(2k + 1))$, $4g(t - 3(2k + 1))$.

Тогда, учтя, что для корректного применения функции $f_n(t)$ надо чтобы в четверти был запущен третий “слой” квадратов, перепишем функцию для второй четверти.

$$N'_{II}(t) = \sum_{m=3}^{\infty} f_m(t - (2k + 1)(m + 2) - 2k + 1),$$

$$N'_{II}(t) = N_{II}(t), \quad t > 5(2k + 1) + 2k - 1.$$

Аналогично переписываются функции для оставшихся четвертей.

Вернёмся к двум квадратам, которые появились из-за жука E . Выпишем для них аналогичную функцию.

$$f(t) = \begin{cases} 2 + 8 + 4 = 14, & \text{при } t \in [1, k]; \\ 4 + 8 + 4 = 16, & \text{при } t \in [k + 1, k + 2]; \\ 8 + 4 = 12, & \text{при } t \in [k + 3, 2k - 5]; \\ 4 + 4 = 8, & \text{при } t \in [2k - 4, 2k - 1]; \\ 2 + 4 = 6, & \text{при } t \in [2k, 2k + 1]. \end{cases}$$

Теперь соберём всю информацию про квадраты в одну сумму

$$\begin{aligned}
N'(t) &= \sum_{m=3}^{\infty} \left(f_{m+4}(t - (2k + 1)((m + 4) + 2) - 2k + 1) + \right. \\
&\quad + f_{m+3}(t - (2k + 1)((m + 3) + 3) - 2k + 1) + \\
&\quad + f_{m+1}(t - (2k + 1)((m + 1) + 5) - 2k + 1) + \\
&\quad + f_m(t - (2k + 1)(m + 6) - 2k + 1) + \\
&\quad \left. + f(t - (2k + 1)(m + 6) - 2k + 1) \right) = \\
&= \sum_{m=3}^{\infty} \left(f_{\lfloor \frac{t}{2k+1} \rfloor - 3}(t - (2k + 1)(m + 6) - 2k + 1) + \right. \\
&\quad + f_{\lfloor \frac{t}{2k+1} \rfloor - 4}(t - (2k + 1)(m + 6) - 2k + 1) + \\
&\quad + f_{\lfloor \frac{t}{2k+1} \rfloor - 6}(t - (2k + 1)(m + 6) - 2k + 1) + \\
&\quad + f_{\lfloor \frac{t}{2k+1} \rfloor - 7}(t - (2k + 1)(m + 6) - 2k + 1) + \\
&\quad \left. f(t - (2k + 1)(m + 7)) \right) = \\
&= \sum_{m=3}^{\infty} F'_{\lfloor \frac{t}{2k+1} \rfloor - 7}(t - (2k + 1)(m + 7)),
\end{aligned}$$

где

$$F'_n(t) = \begin{cases} 7(4n + 8) - 4 * 6 + 14 = 28n + 46, & \text{при } t \in [1, k]; \\ 8(4n + 8) - 4 * 8 + 16 = 32n + 48, & \text{при } t \in [k + 1, k + 2]; \\ 6(4n + 8) - 4 * 4 + 12 = 24n + 44, & \text{при } t \in [k + 3, 2k - 5]; \\ 4(4n + 8) - 4 * 2 + 8 = 16n + 32, & \text{при } t \in [2k - 4, 2k - 1]; \\ 3(4n + 8) - 4 * 1 + 6 = 12n + 26, & \text{при } t \in [2k, 2k + 1]. \end{cases}$$

Выпишем явный вид для $N(t)$ при $t > \max(20k + 10, 8(2k + 1) + 3(k - 1)^2)$. Первая величина в максимуме гарантирует, что жизнь началась уже во всех четвертях, а вторая, что в двух проблемных квадратах, порождён-

ных жизнью жука E , не останется ни одного жука.

$$\begin{aligned}
 N(t) &= \sum_{m=3}^{\infty} F'_{\lfloor \frac{t}{2k+1} \rfloor - 7} (t - (2k+1)(m+6) - 2k+1) + \\
 &\quad + 8 \left(\left\lfloor \frac{t}{2k+1} \right\rfloor - 2, 5 \right) = \\
 &= \sum_{m=3}^{\infty} F'_{\lfloor \frac{t}{2k+1} \rfloor - 7} (t - (2k+1)(m+6) - 2k+1) + \\
 &\quad + 8 \left(\left\lfloor \frac{t}{2k+1} \right\rfloor - 7 \right) + 8 * 4, 5 = \\
 &= \sum_{m=3}^{\infty} F_{\lfloor \frac{t}{2k+1} \rfloor - 7} (t - (2k+1)(m+7)),
 \end{aligned}$$

где

$$F_n(t) = \begin{cases} 28n + 46 + 8n + 36 = 32n + 82, & \text{при } t \in [1, k-2]; \\ 32n + 48 + 8n + 36 = 40n + 84, & \text{при } t \in [k-1, k]; \\ 24n + 44 + 8n + 36 = 32n + 80, & \text{при } t \in [k+1, 2k-7]; \\ 16n + 32 + 8n + 36 = 24n + 68, & \text{при } t \in [2k-6, 2k-3]; \\ 12n + 26 + 8n + 36 = 20n + 62, & \text{при } t \in [2k-2, 2k-1]; \\ 28(n+1) + 46 + 8n + 36 = 32n + 110, & \text{при } t \in [2k, 2k+1]. \end{cases}$$

Таким образом функция численности колеблется со всё возрастающей амплитудой, но постоянным периодом в конусе, образованном прямыми $y = \frac{40(x-(k-1))}{2k+1} - 196$ и $y = \frac{20(x-(2k-1))}{2k+1} - 78$, причём бесконечное число раз достигает как верхней, так и нижней прямой, составляющих конус. Так же понятно, что сколь угодно повышая k мы сможем добиться сколь угодно низкого наклона нашего конуса над горизонтальной осью. Обозначенное выше ограничение, заставляющее нас работать с чётными k , не повлияет на покрытие: найдём k , при которых нижняя граница конуса для k лежит ниже верхней границы для $k+2$.

$$\frac{20}{2k+1} < \frac{40}{2(k+2)+1};$$

$$40k + 100 < 80k + 40;$$

$$40k > 60,$$

что точно верно при $k \geq 16$.

Таким образом для произвольной прямой из класса $\mathcal{A} = \{y = ax + b | 0 < a < \frac{40}{33}, b \in \mathbb{R}\}$ построена такая колония K , что график её функции численности бесконечное число раз пересечёт выбранную прямую. Если прямая пересекала ось Oy выше конуса, ограничивающего график функции численности, то она неизбежно единственный раз пересечёт верхнюю границу, но никогда не пересечёт нижнюю, тогда как сам график бесконечное число раз достигает обеих границ. Аналогична ситуация, когда прямая начинается ниже конуса.

Теорема 1 доказана.

Список литературы

- [1] Хржановский В. Г., *Ботаника*, Высшая школа, Москва, 1975.
- [2] *Зоология беспозвоночных*. Т. 1: *От простейших до моллюсков и артропод*, ред. В. Вестхайде и Р. Ригера., Т-во научных изданий КМК, 2008.
- [3] Lotka, A. J., “Analytical note on certain rhythmic relations in organic systems”, *Proc. Nat. Acad.*, **6** (1920), 410-415.
- [4] Volterra, V., “Fluctuations in the abundance of a species considered mathematically”, *Nature*, **118** (1926), 558-560.
- [5] Н. Б. Лупанова, “Об автоматном моделировании некоторых биологических систем”, *Докл. АН СССР*, **301**:5 (1988), 1066–1069; *Dokl. Math.*, **33**:8 (1988), 559–561.
- [6] Кудрявцев В. Б., Алёшин С. В., Подколзин А. С., *Введение в теорию автоматов: Монография*, 2, Издательство Московского университета, Москва, 2019, ISBN: 978-5-19-011370-9.
- [7] Кудрявцев В. Б., Гасанов Э. Э., Подколзин А. С., *Теория интеллектуальных систем: в 4 кн. Книга четвертая. Теория автоматов.*, Издательские решения, Москва, 2018, ISBN: 978-5-4493-5160-9.
- [8] Подколзин А. С., “О поведении однородных структур”, *Проблемы кибернетики*, 1974, № 31, 133–166.
- [9] Калачев Г. В., Титова Е. Е., “О мере множества законов движения точки, реализуемых клеточными автоматами”, *Интеллектуальные системы. Теория и приложения.*, **22**:3 (2018), 105-125.
- [10] Гасанов Э. Э., “Клеточные автоматы с локаторами”, *Интеллектуальные системы. Теория и приложения.*, **24**:2 (2020), 4-8.
- [11] Ведерников И. К., “Класс автоматов, достаточный для оптимального прогнозирования общерегулярных сверхсобытий”, *Интеллектуальные системы. Теория и приложения.*, **24**:1 (2020).

On the synthesis of a colony of beetles with linear growth
Vorotnikov A.S.

The dynamic system of field and bugs - colony - them is considered. Bugs live on the field that is modeled by integer lattice in each sell of which there are the same count of the food for bugs at the first moment. Bugs must move around field or eat food or divide or dye according to the some algorithm moreover all activities spending energy. This system is modeled by cellular automaton. The colony whose linearized population size infinite number of times crossing line from some class built in this work.

Keywords: automaton modelling of biological system, growth rate of homogeneous structures, cellular automaton.

Часть 3.
Математические модели

Расшифровка булевых функций фиксированного веса

Быстрыгова А.В.¹

В работе исследуется сложность расшифровки класса булевых функций фиксированного веса при помощи запросов на значение, запросов на сравнение, запросов на ограниченную и расширенную эквивалентность. Причем, при расшифровке разрешено использовать только один из упомянутых типов запросов. Для всех типов запросов кроме запросов на сравнение получены точные оценки сложности расшифровки. Для запросов на сравнение приводится верхняя оценка, а также демонстрируется ее совпадение с нижней оценкой для функций веса 1, 2, 3.

Ключевые слова: булевы функции фиксированного веса, запросы на значение, запросы на сравнение, запросы на ограниченную эквивалентность, запросы на расширенную эквивалентность, точная расшифровка.

1. Введение

Должное внимание исследователей с середины прошлого века и по сей день приковано к задачам расшифровки функций фиксированными типами запросов. Под расшифровкой функции из заданного класса понимают игру между двумя игроками: учителем и учеником, — в которой учитель тайно выбирает одну функцию из класса, известного ученику, и затем отвечает на разные его вопросы в отношении выбранной функции. В свою очередь ученик, зная только класс функций, по ответам на свои вопросы должен восстановить функцию, загаданную учителем.

В теории расшифровки функций (computational learning theory) привычной практикой стало рассмотрение задачи расшифровки одного и

¹Быстрыгова Анастасия Викторовна — аспирант каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: anastasiya.bistrigova@yandex.com.

Bistrigova Anastasiya Viktorovna — graduate student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

того же класса функций разными типами запросов или даже разными комбинациями нескольких типов запросов. Пожалуй, одной из первых работ, в которой встречаются запросы на значение, ограниченную эквивалентность и расширенную эквивалентность, является работа [1]. В своей работе Дана Ангуин приводит алгоритм расшифровки произвольного класса функций запросами на ограниченную эквивалентность, который по сути является переборным. В частности, с помощью него показано, что для расшифровки булевых функций веса 1 необходимо задать в худшем случае $2^n - 1$ запросов, причем похожий алгоритм можно адаптировать и под запросы на значение и получить такое же число запросов. Более того, автор приводит так называемый “алгоритм голосования большинством”, позволяющий для произвольного класса функций F понять, какая функция выбрана учителем, за не более $\log_2 |F|$ запросов на расширенную эквивалентность. Автор приводит алгоритм расшифровки запросами на эквивалентность КНФ функции арности n , где каждая дизъюнкция состоит из не более k литерал. Также она показывает, что для расшифровки ДНФ функции арности n , в которой каждая конъюнкция состоит из одного литерала, в худшем случае требуется $2^n - 1$ запросов на эквивалентность. Помимо этого, она рассматривает задачу расшифровки монотонных ДНФ при помощи одновременного использования запросов на ограниченную эквивалентность и запросов на значение и показывает, что для расшифровки функции арности n с m элементарными конъюнкциями понадобится $m + 1$ запросов на ограниченную эквивалентность и mn запросов на значение. В другой своей работе [2] Ангуин вводит обозначения для запросов на значение MQ , на ограниченную EQ и расширенную эквивалентность XEQ , которыми мы и будем далее пользоваться. Помимо этого, с помощью неравенств она демонстрирует соотношение между тем, насколько много запросов потребуется если использовать только один из упомянутых типов запросов или комбинации этих запросов независимо от класса функций.

Касаемо запросов на ограниченную и расширенную эквивалентность, многие работы направлены на обоснование существования полиномиальных алгоритмов расшифровки фиксированных классов функций с помощью этих типов запросов. Одной из таких работ стала работа [3], посвященная рассмотрению функций арности n , равных нулю в случае кратности p суммы значений переменных, и равных единице в противном случае. Авторам [3] удалось получить полиномиальный алгоритм расшифровки класса конъюнкций этих функций по модулю произвольного простого p . Количество запросов на ограниченную эквивалентность,

необходимых данному алгоритму, равно $n^{p-1} + 1$. Авторы [4] продемонстрировали алгоритм, позволяющий расшифровать функцию из класса $\{\sum_{i=1}^n w_i x_i \geq N\}$ за $O(n^2 \log_2 n)$ запросов на ограниченную эквивалентность.

В работе [5] рассматривается расшифровка неповторных функций в базисе всевозможных монотонных пороговых функций запросами на тождественность, которые по сути представляют собой комбинацию запросов на эквивалентность и запросов на значение. Запрос на тождественность — это подфункция. Если подфункция — это набор, тогда ответ на запрос есть значение функции в запрашиваемой точке, иначе, ответ равен 1, если подфункция тождественно равна константе, и 0, если это не так. Авторами показано, что для расшифровки неповторной функции в указанном базисе требуется в худшем случае экспоненциально относительно числа переменных числа запросов тождественности, более того, для произвольного конечного подмножества этого базиса допускается решение полиномиальным числом запросов.

В данной работе помимо представленных Даной Англуин запросов на значение, запросов на ограниченную и расширенную эквивалентность рассматриваются также запросы на сравнение. Задача расшифровки запросами на сравнение стала исследоваться сравнительно недавно. Работа [6] стала первой, в которой изучается сложность расшифровки функций запросами на сравнение. В работе [7] автор рассматривал задачу расшифровки полиномиальных функций ранжирования. Похожий тип запросов применялся в задаче, рассматриваемой в работе [8]. Обе работы [9] и [10] посвящены получению оценок сложности расшифровки для всех замкнутых классов Поста. Различие в том, что в первой рассматривалась задача для запросов на значение, в то время как во второй — для запросов на сравнение. В работе [10] продемонстрировано, что для расшифровки замкнутых классов Поста запросы на сравнение не так уж и уступают запросам на значение, то есть для расшифрования функции из фиксированного замкнутого класса Поста необходимо примерно столько же запросов на сравнение, сколько необходимо запросов на значение. Более того, для двух классов $\{x\}, \{x, \bar{x}\}$ запросами на сравнение можно восстановить загаданную функцию быстрее, чем запросами на значение.

Данная работа посвящена исследованию сложности расшифровки функций фиксированного веса для каждого из четырех упомянутых выше запросов в отдельности: на значение, на сравнение, на ограниченную и расширенную эквивалентность. Для похожего класса — класса функций с ограниченным весом уже проведено исследование [11] функ-

ции Шеннона мощности плоских схем, реализующих такие функции. Но с точки зрения вопросов сложности расшифровки данный класс ранее никем не исследовался, если не брать в расчет результат Англуин для функций веса 1, приведенный выше.

2. Основные понятия и формулировка результатов

Под $P(n)$ арности n . Под $F(n, k)$, где $0 < k < 2^n$, будем понимать множество булевых функций арности n веса k , то есть функций, которые принимают значение 1 ровно на k наборах из 2^n двоичных наборов.

Через $f \equiv g$ при $f, g \in P(n)$ будем обозначать ситуацию, в которой для любого $a \in \{0, 1\}^n$ верно равенство $f(a) = g(a)$.

Пусть загадана функция $f \in F(n, k)$. Тогда определим рассматриваемые типы запросов и ответы на них следующим образом.

Запросом на значение a к функции f называется набор a , а ответом на него является значение функции f на наборе из запроса $f(a)$.

Запросом на сравнение (a, b) будем называть упорядоченную пару наборов a, b , а под ответом на этот запрос понимать знак разности $f(a) - f(b)$.

Под *запросом на ограниченную эквивалентность g к функции f* принято считать функцию $g \in F(n, k)$, а под ответом на указанный запрос — слово *YES*, если $f \equiv g$, и любой набор b такой, что $f(b) \neq g(b)$, в противном случае.

Под *запросом на расширенную эквивалентность g* понимают функцию $g \in P(n)$, под ответом на этот запрос считают слово *YES*, если $f \equiv g$, и какой-то b такой, что $f(b) \neq g(b)$, в противном случае.

Будем говорить, что *последовательность запросов расшифровывает загаданную функцию $f \in F(n, k)$* , если последовательность конечна, состоит из запросов одного типа и функция f однозначно восстанавливается по ответам на запросы этой последовательности.

Алгоритмом расшифровки для запросов на значение $A_{n,k}^{MQ}$ будем называть процесс задания последовательности запросов на значение таким образом, что каждый элемент последовательности выбирается определенным образом в зависимости от ответов учителя на запросы — предыдущие члены последовательности, причем сформированная таким образом последовательность расшифровывают загаданную функцию $f \in F(n, k)$. Аналогично вводится определение алгоритма расшифровки для

запросов на ограниченную эквивалентность $A_{n,k}^{EQ}$ и расширенную эквивалентность $A_{n,k}^{XEQ}$, а также алгоритма расшифровки для запросов на сравнение $A_{n,k}^{CQ}$.

Через $\mathcal{A}_{n,k}^{MQ}$ будем обозначать множество всех алгоритмов $A_{n,k}^{MQ}$ расшифровки для запросов на значение. Похожим образом вводится определение множества всех алгоритмов расшифровки для запросов на ограниченную эквивалентность $\mathcal{A}_{n,k}^{EQ}$ и расширенную эквивалентность $\mathcal{A}_{n,k}^{XEQ}$, а также множества всех алгоритмов расшифровки для запросов на сравнение $\mathcal{A}_{n,k}^{CQ}$.

Через $q(A, f)$ обозначим минимальное количество первых запросов в последовательности запросов алгоритма A , которые расшифровывают функцию $f \in F(n, k)$. Тогда под *сложностью расшифровки запросами* будем понимать число запросов, которое придется задать наилучшему алгоритму расшифровки для расшифровки самой плохой функции. Иными словами, сложность расшифровки запросами типа $T \in \{MQ, CQ, EQ, XEQ\}$ задается следующим образом

$$\varphi_T(n, k) = \min_{A \in \mathcal{A}_{n,k}^T} \max_{f \in F(n, k)} q(A, f).$$

Будем считать, что ученику известны оба значения n и k .

Если a — вещественное число, тогда под $\lceil a \rceil$ будем понимать наименьшее целое, не меньшее a , под $\lfloor a \rfloor$ — наибольшее целое, не большее a , под $a \bmod b$ — остаток от деления a на b , под $|a|$ — модуль числа a .

Если A — множество, то под $|A|$ будем понимать мощность множества A .

Теорема 1. *Сложность расшифровки класса $F(n, k)$ запросами на значение равна $\varphi_{MQ}(n, k) = 2^n - 1$.*

Теорема 2. *Сложность расшифровки класса $F(n, k)$ запросами на ограниченную эквивалентность равна $\varphi_{EQ}(n, k) = 2^n - 1$.*

Теорема 3. *Сложность расшифровки класса $F(n, k)$ запросами на расширенную эквивалентность равна $\varphi_{XEQ}(n, k) = \min(k, 2^n - k)$.*

Теорема 4. *Для запросов на сравнение справедлива следующая оценка $\varphi_{CQ}(n, 1) = 2^{n-1}$.*

Теорема 5. *Если $n \geq 2$, то для запросов на сравнение справедливо равенство $\varphi_{CQ}(n, 2) = \lceil 2^{n+1}/3 \rceil$.*

Теорема 6. Если $n > 6$, то для запросов на сравнение справедливо следующее равенство $\varphi_{CQ}(n, 3) = 2^n - \lfloor 3/2 \cdot \lceil 2^n/5 \rceil - \lfloor (2^n \bmod 5)/2 \rfloor$.

Теорема 7. Пусть $k \leq 2^{n-1}$ и для целых положительных $x_m, x_{m+1}, \dots, x_{k-1}, x_k$, где $m = \lfloor (k+1)/2 \rfloor$, верно равенство

$$2^n = m \cdot x_m + (m+1) \cdot x_{m+1} + \dots + (k-1) \cdot x_{k-1} + k \cdot x_k.$$

Тогда справедлива следующая верхняя оценка

$$\begin{aligned} \varphi_{CQ}(n, k) \leq 2^n - (x_m + x_{m+1} + \dots + x_{k-1} + x_k) \\ + \lfloor \max(x_m, x_{m+1}, \dots, x_{k-1}, x_k)/2 \rfloor. \end{aligned}$$

В частности, при подстановке определенных значений x_m, x_{m+1}, \dots, x_k в теорему 7, получаем следующую теорему.

Теорема 8. Пусть $k \leq 2^{n-1}$, $m = \lfloor (k+1)/2 \rfloor$, $s = m + (m+1) + \dots + (k-1) + k$, верно равенство $2^n = s \cdot q + r$, $r \in [0, s)$, $q \geq m$, q, r — целые положительные числа. Тогда справедлива следующая верхняя оценка

$$\begin{aligned} \varphi_{CQ}(n, k) \leq 2^n - (k-m+1)q - c + \lfloor 0.5 \cdot \max(q-r + (m+1)c, q+r - mc) \rfloor \\ \leq 2^n - k/2 \cdot \lceil 2^n/s \rceil + \lfloor 0.5 \cdot (\lceil 2^n/s \rceil + \lfloor (k+1)/2 \rfloor) \rfloor, \end{aligned}$$

где c вычисляется следующим образом

- $c = \lfloor 2r/(2m+1) \rfloor$ при $2r \bmod (2m+1) \leq m$,
- $c = \lfloor 2r/(2m-1) \rfloor + 1$ при $2r \bmod (2m+1) > m$.

3. Запросы на значение, ограниченную и расширенную эквивалентность

Приведем доказательство теоремы 1.

Доказательство. Верхняя оценка. Заметим, что в работе [1], было доказано неравенство $\varphi_{MQ}(n, 1) \leq 2^n - 1$. Докажем данную верхнюю оценку для произвольного k . Запросим значение на любых $2^n - 1$ наборах. Если среди ответов на эти запросы встретилось ровно k единиц, значит значение функции на неопрошенном наборе равно 0, иначе 1. Соответственно, функция f восстановлена.

Нижняя оценка. Если ученик повторит запрос, просто ответим на него также, как отвечали прежде. Поэтому можно считать, что ученик не повторяет запросы. На первые $2^n - 2 - (k - 1)$ запросов ученика будем отвечать 0, на следующие $k - 1$ запросов ответим 1. Для ученика останутся неопределенными 2 набора и не найдена одна единица загаданной функции. Поэтому он будет вынужден задать еще один запрос. \square

Приведем доказательство теоремы 2.

Доказательство. Верхняя оценка. Заметим, что в работе [1], было доказано неравенство $\varphi_{EQ}(n, 1) \leq 2^n - 1$. Докажем данную верхнюю оценку для произвольного k . Положим A — множество всех 2^n двоичных n -местных наборов, про которые мы пока не знаем, чему равно значение загаданной функции на них. A_0, A_1 — множество наборов, на которых значение загаданной функции равно 0 и 1 соответственно. Изначально, оба множества A_0, A_1 пусты. Каждый запрос формируем следующим образом. Выбираем любое подмножество B мощности $(k - |A_1|)$ из A и учителю передаем функцию g , которая равна единице на наборах из $B \cup A_1$ и равна нулю на остальных наборах (то есть наборах из $A_0 \cup (A \setminus B)$). Если учитель возвращает в ответ запрос x , то:

- 1) если $x \in B$, тогда удаляем x из A и добавляем в A_0 ,
- 2) если $x \notin B$, тогда удаляем x из A и добавляем в A_1 .

Как только $|A| + |A_1| = k$, тогда делаем вывод, что загаданная функция равна единице на всех наборах в A и значит f восстановлена. Если $|A_1| = k$, тогда загаданная функция также восстановлена.

Достаточно задать $2^n - 1$ запросов. Так как за один запрос раскрывается значение ровно на одном наборе. Если $|A_1| < k$ после опроса $2^n - 1$ запросов, то значение функции на оставшемся наборе равно единице, иначе 0.

Нижняя оценка. Если в ответ на свой запрос ученик получил значение функции на каком-то наборе, а позже отправил запрос-функцию, которая отличается в соответствующем наборе от верного значения, тогда учитель в качестве ответа вновь отправляет ему этот набор.

Поэтому можно считать, что ученик не совершает такие “бесполезные” запросы. На каждый из первых $2^n - 2 - (k - 1)$ запросов будем возвращать набор, на котором функция ученика равна 1. Следовательно, за каждый такой запрос, мы раскроем информацию об одном нуле. На каждый из $k - 1$ последующих запросов будем возвращать набор, на

котором функция ученика равна 0. Таким образом, за каждый такой запрос мы раскроем информацию об одной единице. Ученику неизвестно значение на ровно двух наборах и неизвестна одна единица. Поэтому он вынужден сделать еще один запрос. \square

Далее следует доказательство теоремы 3.

Доказательство. Верхняя оценка. Пусть $k \leq 2^n - k$. Первым запросом отправим константу нуль. Так как $k > 0$, мы получим набор, на котором значение загаданной функции равно 1, тем самым мы раскроем информацию об одной единице. Далее отправим функцию, которая равна нулю всюду за исключением раскрытой единицы, и в ответ получим информацию о второй единице. Действуя дальше аналогично, за ровно k запросов мы восстановим загаданную функцию.

Если $k > 2^n - k$. Действуем аналогично с заменой 0 на 1 в предыдущей части доказательства.

Нижняя оценка. Пусть $k \leq 2^n - k$. Положим A — множество всех 2^n двоичных n -местных наборов, значение загаданной функции на которой пока неизвестно.

На каждый очередной запрос g ученика отвечаем следующим образом:

- 1) если в A лежит набор x , на котором g обращается в 1, тогда возвращаем ученику x и удаляем x из A ,
- 2) иначе, если в A лежит набор x , на котором g обращается в 0, тогда возвращаем ученику x и удаляем x из A .

Если не сработал ни один из пунктов, значит множество A пустое, а значит вся функция f восстановлена.

Если учитель воспользовался первым пунктом своей стратегии, то он раскрыл ученику информацию об одном нуле, которых очень много.

Если учитель воспользовался вторым пунктом своей стратегии, то он раскрыл ученику информацию об одной единице, которых не так то и много.

Поэтому если ученик вынуждает учителя постоянно пользоваться пунктом 2, тогда он восстановит функцию за k запросов. \square

4. Расшифровка функций веса 1 запросами на сравнение

Рассматривая запросы на сравнение, будем говорить, что набор a был опрошен, если он был одним из наборов какого-то заданного запроса на сравнение. Также будем говорить, запрос (a, b) покрывает набор x , если набор x совпадает с a или b . Будем говорить, что множество запросов покрывает набор, если хотя бы один запрос из этого множества покрывает этот набор.

Замечание 1. Если известен ответ на запрос (a, b) , а также значение функции на одном из наборов запроса ($f(a)$ или $f(b)$), тогда однозначно восстанавливается значение функции на втором наборе запроса ($f(b)$ или $f(a)$ соответственно).

Лемма 1. Справедлива следующая верхняя оценка

$$\varphi_{CQ}(n, k) \leq k \lfloor 2^n / (k + 1) \rfloor + \max(0, (2^n \bmod (k + 1)) - 1).$$

Доказательство. Пусть $2^n = (k + 1)q + r$, $0 \leq r < (k + 1)$, $0 < q$, q, r — целые числа.

Упорядочим произвольным образом все 2^n двоичных наборов и обозначим их через $x_0, x_1, \dots, x_{2^n-2}, x_{2^n-1}$. Зададим q групп по k запросов. i -я ($0 \leq i < q$) группа состоит из запросов вида $(x_{i \cdot (k+1)+j}, x_{i \cdot (k+1)+j+1})$, где j меняется от 0 до $k - 1$ включительно.

Рассмотрим ответы для i -й группы запросов ($0 \leq i < q$). Возможны два случая.

- 1) Ответы все запросы равны 0, тогда значение загаданной функции на всех наборах из запросов одно и то же. В силу того, что таких наборов $k + 1$, а единиц функции ровно k , значит единицами функциями эти наборы не могут быть и $f(x_{i \cdot (k+1)+j}) = 0$ для всех $j \in \{0, 1, \dots, k\}$.
- 2) Существует j_0 — наименьшее целое из интервала $[0, k - 1]$ такое, что ответ на запрос отличен от нуля, иными словами, $f(x_{i \cdot (k+1)+j_0}) \neq f(x_{i \cdot (k+1)+j_0+1})$.
 - а) Если ответ на запрос $(x_{i \cdot (k+1)+j_0}, x_{i \cdot (k+1)+j_0+1})$ равен 1, то в силу того, что ответы на все запросы $(x_{i \cdot (k+1)+j}, x_{i \cdot (k+1)+j+1})$, $j < j_0$, равны 0, а $f(x_{i \cdot (k+1)+j_0}) = 1$, следует $f(x_{i \cdot (k+1)+j}) = 1$ для

всех $j \in \{0, 1, \dots, j_0\}$. Более того, из ответов на оставшиеся запросы этой группы можно последовательно восстановить значение на всех наборах этой группы, то есть сначала восстановить значение на наборе $x_{i \cdot (k+1) + j_0 + 1}$, затем на наборе $x_{i \cdot (k+1) + j_0 + 2}$ и так далее. Это возможно в силу замечания 1.

- б) Если ответ на запрос $(x_{i \cdot (k+1) + j_0}, x_{i \cdot (k+1) + j_0 + 1})$ равен -1 , то в силу того, что ответы на все запросы $(x_{i \cdot (k+1) + j}, x_{i \cdot (k+1) + j + 1})$, $j < j_0$, равны 0 , а $f(x_{i \cdot (k+1) + j_0 + 1}) = 1$, следует $f(x_{i \cdot (k+1) + j}) = 0$ для всех $j \in \{0, 1, \dots, j_0\}$. Аналогично предыдущему пункту восстановим значение на всех остальных наборах этой группы.

Итого, будет задано $kq = k \cdot \lfloor 2^n / (k + 1) \rfloor$ запросов и восстановлено значение на всех $q \cdot (k + 1)$ наборах, останется неизвестным значение на r наборах. Обозначим через x количество единиц, которое будет найдено за эти kq запросов. Тогда если $x = k$, то все единицы уже найдены. Если $x + r = k$, то оставшиеся r наборов также являются единицами и значит вновь функция полностью расшифрована. Иначе, необходимо найти $k - x$ единиц, где $0 < k - x < r$. Зададим $r - 1$ запросов, где в каждом запросе первая компонента — это один из непокрытых r наборов, а вторая — любой из покрытых ранее. Тогда по ответам на эти запросы мы однозначно восстановим значение функции на всех наборах, кроме оставшегося непокрытого одного. Если после этих запросов найдены все единицы искомой функции, значит непокрытый набор является нулем функции, иначе, этот набор и есть оставшаяся ненайденная единица.

В общей сложности для расшифровки функции будет потрачено следующее количество запросов.

- 1) Если $r = 0$, то $k \cdot 2^n / (k + 1)$ запросов.
- 2) Если $r \geq 1$, то $k \cdot \lfloor 2^n / (k + 1) \rfloor + (r - 1)$ запросов.

□

Приведем доказательство теоремы 4.

Доказательство. Покажем, что $\varphi_{CQ}(n, 1) \geq 2^{n-1}$. Пусть ученик задаст $2^{n-1} - 1$ запросов, на каждый ответим 0 . В итоге, суммарно будет опрошено не более $(2^{n-1} - 1) \cdot 2 = 2^n - 2$ наборов. Про каждый из них ученик поймет, что значение функции на нем равно 0 . Остается как минимум два набора, на которых неизвестно значение функции, следовательно нужно задать хотя бы еще один запрос.

Учитывая верхнюю оценку, полученную в лемме 1, получаем оценку $\varphi_{CQ}(n, 1) = 2^{n-1}$. \square

5. Расшифровка функций веса 2 запросами на сравнение

Прежде всего, рассмотрим следующую задачу. Имеется n одноэлементных множеств. За одну операцию разрешается объединить в одно множество любые два имеющихся на данный момент множества. Цель — за наименьшее число операций объединения получить множества мощности не меньшей заданного числа.

Заметим, что эту задачу можно переформулировать следующим образом. Изначально имеется n -вершинный граф без ребер. За одну операцию можно провести неориентированное ребро между двумя вершинами. Необходимо, используя минимальное число ребер, объединить исходные n вершин в компоненты связности, в которых вершин не меньше заданного числа.

Лемма 2. Пусть n кратно q , тогда для того, чтобы после операций объединения остались только множества мощности строго равной q , достаточно в точности $n \cdot (q - 1)/q$ операций, более того, эта оценка не понижается.

Доказательство. Поскольку все n элементов распадутся на множества мощности равной q , тогда получится в точности n/q множеств. Чтобы объединить q элементов в одно множество необходимо в точности $q - 1$ операций объединения, соответствующее количеству ребер в дереве из q вершин. \square

Лемма 3. Пусть n кратно q , тогда минимальное количество операций объединения, которое необходимо выполнить для того, чтобы после операций объединения остались только множества мощности не меньше q , равно $n \cdot (q - 1)/q$.

Доказательство. Согласно лемме 2, если объединять множества так, чтобы получились только множества мощности ровно q , необходимо не менее $n \cdot (q - 1)/q$ операций.

Рассмотрим общий случай итоговой системы множеств после применения операций объединения. Пусть n_1 кратно q , n_2 кратно q_2 , ..., n_r кратно q_r , при этом $q < q_2 < \dots < q_r$, $n_1 + n_2 + \dots + n_r = n$. Тогда для

того, чтобы n_1 изначальных одноэлементных множеств объединились в q -элементные, n_2 — в q_2 -элементные, \dots , n_r — в q_r -элементные множества, потребуется, согласно лемме 2, не менее $n_1(q-1)/q + n_2(q_2-1)/q_2 + \dots + n_r(q_r-1)/q_r$ операций объединения.

Но учитывая, неравенство $\frac{t_1-1}{t_1} < \frac{t_2-1}{t_2}$ при $0 < t_1 < t_2$, получаем цепочку неравенств

$$\begin{aligned} & n \frac{q-1}{q} - (n_1 \frac{q-1}{q} + n_2 \frac{q_2-1}{q_2} + \dots + n_r \frac{q_r-1}{q_r}) < \\ < n \frac{q-1}{q} - \frac{q-1}{q} (n_1 + n_2 + \dots + n_r) = \frac{q-1}{q} \cdot (n - n) = 0. \end{aligned}$$

Следовательно, $n \frac{q-1}{q} < (n_1 \frac{q-1}{q} + n_2 \frac{q_2-1}{q_2} + \dots + n_r \frac{q_r-1}{q_r})$.

Иными словами, для объединения изначальных n одноэлементных множеств во множества мощности не меньше q оптимальнее всего объединить их во множества мощности равной q . \square

Лемма 4. В графе с V вершинами, E ребрами и K компонентами связности справедливо неравенство $E + K \geq V$.

Доказательство. Доказательство по индукции по количеству ребер в графе.

База индукции. $E = 0$, тогда $K = V$. Очевидно неравенство $K \geq V$. Переход индукции. Пусть $E > 0$ и справедливо неравенство $E + K \geq V$. Добавим одно ребро, тогда E увеличится на 1, а K либо уменьшится на 1, если новое ребро связало вершины из разных компонент связности, либо не изменится, если новое ребро связало две вершины из одной компоненты связности. Следовательно, неравенство $E + K \geq V$ сохранится. \square

Лемма 5. Пусть $n = xq + r$, где q, x, r — целые положительные числа, $1 \leq r < q$, тогда для того, чтобы после операций объединения остались только множества мощности не меньше q , необходимо сделать не менее $x \cdot (q-1) + r$ операций объединения.

Доказательство. Докажем от противного. Пусть существуют n, q, x, r — целые положительные, такие что $n = xq + r, 0 < r < q$, и существует порядок объединения исходных n множеств во множества мощности не менее q , в котором используется $x \cdot (q-1) + (r-1)$ операций объединения.

Согласно лемме 4, исходные n вершин после добавления $x \cdot (q-1) + (r-1)$ ребер распадутся на не менее $n - (x \cdot (q-1) + r - 1) = x + 1$

компонент связности. При этом, в текущей лемме выше утверждается, что все получившиеся компоненты связности имеют мощность не менее q . Соответственно, суммарно во всех получившихся компонентах связности не менее $q(x + 1) = qx + q$ вершин. Получили противоречие с исходным количеством вершин $n = qx + r < qx + q$. \square

Приведем доказательство теоремы 5.

Доказательство. Докажем нижнюю оценку $\varphi_C(n, 2) \geq 2\lceil 2^n/3 \rceil + (2^n \bmod 3) - 1$. Возможны два случая: $2^n = 3x + 1$ и $2^n = 3x + 2$, где x — целое.

Рассмотрим первый случай: $2^n = 3x + 1$. Необходимо доказать, что ученик задаст хотя бы $2x$ запросов. Пусть он задаст $2x - 1$ запросов, на каждый запрос ответим 0, тогда покажем, что либо ему недостаточно информации, полученной по этим запросам, чтобы восстановить загаданную функцию, либо по заданным запросам можно восстановить порядок объединения множеств во множества мощности не менее 3 за меньшее число запросов, чем утверждается в нижней оценке в лемме 3 или лемме 5. Итак, ученик задал $2x - 1$ запросов. Рассмотрим все возможные случаи.

1) $2x - 1$ запросов покрывают не более $3x$ наборов. Обозначим через a_0 один из непокрытых наборов. Согласно лемме 3, $3x$ наборов после не менее $2x$ запросов могут объединиться в x множеств мощности 3. Но раз задано на один запрос меньше, то в лучшем случае сформировано уже $x - 1$ множеств мощности 3, а для образования еще одного такого не хватает одного запроса. Следовательно, среди всех наборов, за исключением набора a_0 , имеется еще

- а) либо три непокрытых набора a_1, a_2, a_3 ,
- б) либо один непокрытый a_1 и одно множество, состоящее из наборов a_2, a_3 ,
- в) либо два двухэлементных множества.

В первом случае 2 единицы функции могут находиться в любых двух наборах из a_0, a_1, a_2, a_3 . Во втором случае 2 единицы либо a_2, a_3 , либо a_0, a_1 . В третьем случае, обе единицы могут находиться в любом из этих двух двухэлементных множеств.

В любом случае, ученик вынужден задать еще хотя бы один запрос, чтобы избавиться от возникшей неоднозначности.

- 2) $2x - 1$ запросов покрывают все $2^n = 3x + 1$ наборов. В силу того, что отсутствуют непокрытые наборы, ответы на все запросы равны 0, значит все наборы разбились на множества мощности хотя бы 2. Если имеются как минимум два множества мощности ровно 2, то учитель может спрятать обе единицы в одно из множеств, поэтому ученик вынужден задать еще хотя бы один запрос. Следовательно, либо множеств мощности 2 ровно одно или нуль.

Случай отсутствия множеств мощности 2 невозможен, так как в этом случае все $3x + 1$ одноэлементных множеств за $2x - 1$ операций объединились во множества мощности не менее 3, что противоречит лемме 5.

Следовательно, после $2x - 1$ операций объединения исходные $3x + 1$ одноэлементных множества объединились во множества, среди которых ровно одно двухэлементное, а остальные мощности не менее трех. Но и этот случай невозможен, так как противоречит лемме 5, потому что $(3x + 1) - 2 = 3(x - 1) + 2$ одноэлементных множеств объединились во множества мощности не менее 3 за $(2x - 1) - 1$ операций вместо $2x$, как утверждается в лемме 5.

Следовательно, в случае $2^n = 3x + 1$ ученик вынужден задать как минимум $2x$ запросов.

Рассмотрим случай $2^n = 3x + 2$. Необходимо доказать, что ученик вынужден задать хотя бы $2x + 1$ запросов.

Пусть ученик задал $2x$ запросов. На первые $2x - 1$ запросов отвечаем 0.

Если $2x$ запросов покрывают ровно $3x$ наборов и к запросу с номером $2x$ в точности $3x - 3$ наборов объединились во множества мощности 3, а запрос с номером $2x$, если в ответ на него придет 0, объединит еще три элемента в одно трехэлементное множество, то есть одна компонента этого запроса — набор одноэлементного множества, вторая — набор двухэлементного множества, тогда ответим

- -1, если первая компонента — набор одноэлементного множества,
- 1, если вторая компонента — набор одноэлементного множества.

Иначе, на запрос с номером $2x$ ответим 0.

Рассмотрим все возможные случаи.

- 1) $2x$ запросов покрыли не более $3x - 1$ наборов. Тогда хотя бы 3 набора не покрыты. Ответы на все запросы были равны 0. Учитель

может спрятать обе единицы в двух из трех непокрытых наборах. Поэтому ученик вынужден задать хотя бы еще один запрос.

2) $2x$ запросов покрыли ровно $3x$ наборов. Тогда возможна одна из следующих двух ситуаций.

а) Ответ на запрос с номером $2x$ был отличен от 0. Следовательно, учитель раскрыл информацию о ровно одной единице и двух нулях функции. Осталось найти еще одну единицу. В силу того, что ответы на первые $2x - 1$ запросов были равны 0, то все остальные наборы из рассматриваемых $3x$ объединились во множества мощности не меньшей 2. В этих множествах вторая единица не может находиться, значит она находится среди непокрытых двух наборов. Следовательно, ученик вынужден задать еще один запрос для однозначного восстановления функции.

б) Ответ на запрос с номером $2x$ был равен 0. Следовательно, рассматриваемые $3x$ наборов не объединились в x множеств мощности 3. Возможна одна из следующих ситуаций.

– Среди множеств, в которые объединились $3x$ наборов, есть множество мощности 2. Тогда учитель может спрятать обе единицы либо в два непокрытых набора, либо в этом двухэлементном множестве. Ученик вынужден задать еще один запрос для восстановления функции.

– Среди множеств, в которые объединились $3x$ наборов, нет множеств мощности 2. Но не все множества мощности ровно 3. Следовательно, $3x$ наборов объединились в p множеств мощности не менее 3, причем существует хотя бы одно множество мощности строго больше 3. На образование p множеств мощности ровно 3 необходимо не менее $2p$ запросов, и суммарно останется $3x - 3p > 0$ наборов раскидать по этим множествам, на добавление каждого такого набора в какое-то из p множеств тратится не менее 1 запроса. Исходя из этого, получаем цепочку неравенств $2x \geq 2p + (3x - 3p)$, $p \geq x$, $3p \geq 3x$, что противоречит неравенству $3x - 3p > 0$.

3) $2x$ запросов покрыли хотя бы $3x + 1$ наборов. Тогда возможна одна из следующих трех ситуаций.

- а) Не менее $3x + 1$ наборов объединились во множества мощности не меньшей 3. Этот случай невозможен, так как противоречит лемме 5.
- б) Среди множеств, в которые объединились не менее $3x + 1$ наборов, имеются хотя бы два множества мощности 2. Тогда обе единицы можно спрятать в любое из них и ученик вынужден использовать еще один дополнительный запрос для понимания, какое из двухэлементных множеств содержит обе единицы.
- в) Среди множеств, в которые объединились не менее $3x + 1$ наборов, имеется ровно одно множество мощности 2. Для рассмотрения этой ситуации отдельно рассмотрим случай, когда $2x$ запросов покрыли $3x + 1$ наборов и когда $2x$ запросов покрыли $3x + 2$ набора.
- $2x$ запросов покрыли $3x + 1$ наборов. Соответственно, $3x - 1 = 3(x - 1) + 2$ наборов за $2x - 1 = 2(x - 1) + 1$ запросов объединились во множества мощности не менее 3, что противоречит лемме 5. Значит, этот случай невозможен.
 - $2x$ запросов покрыли $3x + 2$ набора. Получается, что $3x$ наборов за $2x - 1$ запросов объединились во множества мощности не менее 3, что противоречит лемме 3. Следовательно, и этот случай невозможен.

Исходя из рассмотренных случаев становится ясно, что в случае $2^n = 3x + 2$ ученик вынужден задать как минимум $2x + 1$ запросов.

Учитывая нижнюю оценку, полученную в доказательстве данной теоремы, и верхнюю оценку, полученную в лемме 1, имеем следующее равенство $\varphi_{CQ}(n, 2) = 2\lceil 2^n/3 \rceil + (2^n \bmod 3) - 1$.

Покажем, что последняя величина равна $\lceil 2^{n+1}/3 \rceil$. Для этого рассмотрим два случая.

- 1) Если $2^n = 3x + 1$, тогда $\varphi_{CQ}(n, 2) = 2x$, но с другой стороны $\lceil 2^{n+1}/3 \rceil = \lceil 2(3x + 1)/3 \rceil = \lceil 2x + 2/3 \rceil = 2x$.
- 2) Если $2^n = 3x + 2$, тогда $\varphi_{CQ}(n, 2) = 2x + 1$, но с другой стороны $\lceil 2^{n+1}/3 \rceil = \lceil 2(3x + 2)/3 \rceil = \lceil 2x + 4/3 \rceil = 2x + 1$.

□

6. Верхние оценки сложности расшифровки запросами на сравнение

Далее рассмотрим доказательство теоремы 7, представляющую верхнюю оценку сложности расшифровки функций любой арности n и произвольного веса $k \leq 2^{n-1}$.

Доказательство. Напомним, что через m мы обозначали $\lfloor (k+1)/2 \rfloor$. Произвольным образом разобьем 2^n наборов на x_m множеств размера m , x_{m+1} множеств размера $m+1$, ..., x_{k-1} множеств размера $k-1$ и x_k множеств размера k . Заметим, что $k-m < m$. Обозначим все множества следующим образом $A_{i,j}$, где i указывает размер множества, то есть число от m до k , а j — номер множества размера i , то есть число от 1 до x_i . Элементы множества $A_{i,j}$ будем обозначать $\{a_{i,j}^h | h = 1, i\}$.

Для каждого множества $A_{i,j}$, $i = \overline{m, k}$, $j = \overline{1, x_i}$ зададим следующие запросы: $(a_{i,j}^1, a_{i,j}^2), (a_{i,j}^2, a_{i,j}^3), \dots, (a_{i,j}^{i-2}, a_{i,j}^{i-1}), (a_{i,j}^{i-1}, a_{i,j}^i)$. Иными словами, мы как будто сцепляем изолированные i вершин ребрами в одну цепочку. Рассмотрим ответы на эти $i-1$ запросов. Возможны два случая.

- 1) Ответы все запросы равны 0, тогда значение загаданной функции на всех наборах множества $A_{i,j}$ одно и то же.
- 2) Существует q — наименьшее целое из интервала $[1, i-1]$ такое, что ответ на q -й запрос отличен от нуля, иными словами, $f(a_{i,j}^q) \neq f(a_{i,j}^{q+1})$.
 - а) Если ответ на запрос $(a_{i,j}^q, a_{i,j}^{q+1})$ равен 1, то в силу того, что ответы на все запросы $(a_{i,j}^p, a_{i,j}^{p+1}), p < q$ равны 0, а $f(a_{i,j}^q) = 1$, следует $f(a_{i,j}^p) = 1$ для всех $p \in \{1, 2, \dots, q-1, q\}$. Более того, из ответов на оставшиеся запросы этой группы можно последовательно восстановить значение на всех наборах этой группы, то есть сначала восстановить значение на наборе $a_{i,j}^{q+1}$, затем на наборе $a_{i,j}^{q+2}$ и так далее. Это осуществимо благодаря замечанию 1.
 - б) Если ответ на запрос $(a_{i,j}^q, a_{i,j}^{q+1})$ равен -1 , то в силу того, что ответы на все запросы $(a_{i,j}^p, a_{i,j}^{p+1}), p < q$ равны 0, а $f(a_{i,j}^{q+1}) = 1$, следует $f(a_{i,j}^p) = 0$ для всех $p \in \{1, 2, \dots, q-1, q\}$. Аналогично предыдущему пункту восстановим значение на всех остальных наборах этой группы.

Обратим внимание на то, что в результате этих запросов про каждое множество $A_{i,j}$ мы знаем либо значение функции на всех наборах этого множества, либо то, что на всех наборах множества функция принимает одинаковое значение.

Подытоживая, получаем, что было задано $(m-1) \cdot x_m + m \cdot x_{m+1} + \dots + (k-2) \cdot x_{k-1} + (k-1) \cdot x_k$ запросов, все 2^n наборов покрыты. Причем про какие-то наборы уже известно, чему равно значение функции в них, а оставшиеся наборы распались на классы эквивалентности мощности не меньшей m . Обозначим через t количество единиц функции, найденных в результате этих запросов. Если $t = k$, то искомая функция расшифрована. Иначе, осталось найти $k - t > 0$ единиц и точно известно, что все они лежат ровно в одном из образовавшихся множеств — классов эквивалентности. Поскольку в случае, если бы все ненайденные единицы находились хотя бы в двух множествах $A_{i_1, j_1}, A_{i_2, j_2}$, тогда получилось бы, что $k - t \geq i_1 + i_2 > k$.

Следовательно, если после получения ответов на упомянутые выше запросы, не были найдены все единицы, то остается найти $t \in [m, k]$ единиц, которые находятся ровно в одном из множеств $A_{t,1}, A_{t,2}, \dots, A_{t,x_t-1}, A_{t,x_t}$. Эта задача эквивалентна задаче поиска одной единицы среди наборов $a_{t,1}^1, a_{t,2}^1, \dots, a_{t,x_t-1}^1, a_{t,x_t}^1$, то есть среди первых элементов этих множеств. Причем, про какие-то из этих множеств мы уже узнали значение функции на них. Не нарушая общности, будем считать, что про последние $(x_t - p) \in [0, x_t - 1]$ множеств $A_{t,p+1}, A_{t,p+2}, \dots, A_{t,x_t}$ мы знаем значение функции на каждом его элементе. Соответственно, необходимо найти единицу среди наборов $a_{t,1}^1, a_{t,2}^1, \dots, a_{t,p}^1$.

Для этого понадобится $\lceil p/2 \rceil$ запросов вида $(a_{t,2k+1}^1, a_{t,2k+2}^1)$, где $k = \overline{0, \lceil p/2 \rceil - 1}$. Если $\lceil p/2 \rceil$ — нечетно, то одна из единиц будет точно набор $a_{t,p}$, если на все последние запросы будет получен ответ 0.

Итого, в худшем случае будет задано $(m-1) \cdot x_m + m \cdot x_{m+1} + \dots + (k-2) \cdot x_{k-1} + (k-1) \cdot x_k + \lceil x_t/2 \rceil$ запросов. Из равенства $2^n = m \cdot x_m + (m+1) \cdot x_{m+1} + \dots + (k-1) \cdot x_{k-1} + k \cdot x_k$ получаем равенство

$$\begin{aligned} (m-1) \cdot x_m + m \cdot x_{m+1} + \dots + (k-2) \cdot x_{k-1} + (k-1) \cdot x_k &= \\ &= 2^n - (x_m + x_{m+1} + \dots + x_k). \end{aligned}$$

Это и приводит нас к оценке из теоремы. □

Перейдем к доказательству теоремы 8.

Доказательство. Положим $x_m = q - r + (m + 1) \cdot c$, $x_{m+1} = q + r - m \cdot c$, а $x_{m+2} = x_{m+3} = \dots = x_k = q$. Обозначим через $p = 2r \bmod (2m + 1)$. Определим c следующим образом.

- при $p \leq m$ положим равным $c = [2r/(2m + 1)]$,
- при $p > m$ положим равным $c = [2r/(2m + 1)] + 1$.

Заметим, что в случае $p \leq m$ выполнено $x_{m+1} - x_m = 2r - c(2m + 1) = p \leq m$. В случае $p > m$ верна цепочка равенств $x_m - x_{m+1} = c(2m + 1) - 2r = 2m + 1 - p \leq m$. Таким образом, мы присвоили x_i значения, отличающиеся по модулю не более чем на m .

Подставляя полученные значения x_i в верхнюю оценку теоремы 7, получаем первое неравенство теоремы.

Заметим, что $k - m + 1 \geq k/2$, $q = [2^n/s]$. Учитывая неравенство, $|x_m - x_{m+1}| \leq m$, получаем $\max(x_m, x_{m+1}) \leq q + m = [2^n/s] + (k + 1)/2$. Все это позволяет нам получить второе неравенство из утверждения теоремы. \square

7. Расшифровка функций веса 3 запросами на сравнение

Лемма 6. Пусть $n \geq 3$ и для целых положительных x_2, x_3 верно равенство $2^n = 2x_2 + 3x_3$, причем $x_2 \geq 4$. Тогда справедливо неравенство

$$2[(2^n - 3x_3)/3] + ((2^n - 3x_3) \bmod 3) - 1 \leq x_2 + [(x_2 - 1)/2].$$

Доказательство. Пусть $2^n - 3x_3 = 3q + r$, где q, r — целые неотрицательные числа, $r \in [1, 2]$. Соответственно, имеем цепочку равенств $2x_2 = 2^n - 3x_3 = 3q + r$. Перепишем искомое неравенство при помощи введенных обозначений $2[2x_2/3] + (2x_2 \bmod 3) - 1 \leq x_2 + [(x_2 - 1)/2]$ или $2q + r - 1 \leq x_2 + [(x_2 - 1)/2]$. Рассмотрим два случая в зависимости от делимости x_2 на 2.

- Пусть $x_2 = 2s$, $s \geq 0$. Левая часть искомого неравенства имеет вид $(2q + r) - 1 = (3q + r) - q - 1 = 4s - (4s - r)/3 - 1 = 8s/3 + r/3 - 1$. Правая часть искомого неравенства имеет вид $2s + (s - 1) = 3s - 1$. Рассмотрим разность $8s/3 + r/3 - 1 - 3s + 1 = -s/3 + r/3$. Приходим к выводу, что искомое неравенство справедливо при $s \geq 2, x_2 \geq 4, (2^n - 3x_3) \geq 8, n \geq 3$.

- Пусть $x_2 = 2s + 1, s \geq 0$. Левая часть искомого неравенства имеет вид $(2q + r) - 1 = (3q + r) - q - 1 = 4s + 2 - (4s + 2 - r)/3 - 1 = 8s/3 + (1 + r)/3$. Правая часть искомого неравенства имеет вид $2s + 1 + s = 3s + 1$. Рассмотрим разность $8s/3 + (1 + r)/3 - 3s - 1 = -s/3 - 2/3 + r/3$. Приходим к выводу, что искомое неравенство справедливо при $s \geq 0$, что у нас изначально и выполняется.

□

Лемма 7. Пусть $n \geq 3$ и для целых положительных x_2, x_3 верно равенство $2^n = 2x_2 + 3x_3$, причем $x_2 < 4 \leq x_3$. Тогда имеет неравенство

$$2\lceil 2^n/3 \rceil + (2^n \bmod 3) - 1 \leq 2^n - (x_2 + x_3) + \lceil x_3/2 \rceil.$$

Доказательство. Пусть $2^n = 3q + r, q \geq 0, r \in [1, 2]$. Рассмотрим 4 случая в зависимости от значений, которые принимает x_2 .

- 1) $x_2 = 0$. Поскольку $x_3 = (2^n - 2x_2)/3$, то такой случай невозможен в силу не равенства $2^n \bmod 3$ нулю.
- 2) $x_2 = 1$. Поскольку $x_3 = (2^n - 2x_2)/3$, тогда $r = 2$, а $x_3 = q$. Искомое неравенство принимает вид $2q + 1 \leq 3q + 2 - (1 + q) + \lceil q/2 \rceil = 2q + 1 + \lceil q/2 \rceil$, которое верно при любом $q \geq 0$.
- 3) $x_2 = 2$. Поскольку $x_3 = (2^n - 2x_2)/3$, тогда $r = 1$, а $x_3 = q - 1$. Искомое неравенство принимает вид $2q \leq 3q + 1 - (2 + q - 1) + \lceil (q - 1)/2 \rceil = 2q + \lceil (q - 1)/2 \rceil$, которое верно при любом $q - 1 \geq 0$, что и имеем, поскольку $x_3 = q - 1 \geq 4$.
- 4) $x_2 = 3$. Поскольку $x_3 = (2^n - 2x_2)/3$, то такой случай невозможен в силу не равенства $2^n \bmod 3$ нулю.

□

Пусть A — алгоритм расшифровки $F(n, k)$, покрывающий все 2^n наборы. Тогда через $C(A)$ будем обозначать такое число q , что первые $q - 1$ запросов алгоритма A покрывают не все наборы, а q запросов покрывают все 2^n наборов. Представим, что каждый набор — это одноэлементное множество. Будем говорить, что запрос (a, b) объединяет два множества, которым принадлежат наборы a и b . Тогда под $N(A, x, y)$ будем понимать количество множеств мощности y , которые образовались после отправки первых x запросов алгоритма A .

Заметим, что для $k = 3$ можно немного уточнить приведенную в теореме 7 верхнюю оценку $2^n - (x_2 + x_3) + [\max(x_2, x_3)/2]$. Поэтому перейдем к доказательству следующей леммы.

Лемма 8. Пусть $n \geq 4$ и для целых положительных x_2, x_3 верно равенство $2^n = 2x_2 + 3x_3$. Тогда справедлива следующая верхняя оценка

$$\varphi_{CQ}(n, 3) \leq 2^n - (x_2 + x_3) + [\max(x_2 - 1, x_3)/2].$$

Доказательство. Рассмотрим алгоритм A расшифровки функции, приведенный в теореме 7. Соответственно, зададим $(2^n - x_2 - x_3)$ запросов, образовав $N(A, 2^n - (x_2 + x_3), 3) = x_3$ множеств мощности 3 и $N(A, 2^n - (x_2 + x_3), 2) = x_2$ множеств мощности 2 соответственно. Причем, сначала будем задавать запросы, относящиеся ко множествам мощности 3, а лишь затем ко множествам мощности 2.

При этом заметим что, если ответ на каждый из первых $(2^n - x_2 - x_3)$ запросов был 0, тогда все единицы находятся в одном из x_3 множеств мощности 3 и для нахождения нужного множества мощности 3 надо дополнительно задать $[x_3/2]$ запросов, то есть суммарно будет задано $2^n - (x_2 + x_3) + [x_3/2]$ запросов. Иначе, если хоть раз был получен ответ отличный от 0, тогда стало известно значение на всех элементах множества 2 или множества 3.

- Если первый ответ отличный от 0 был получен на запросах, относящихся к формированию множеств мощности 2, тогда все множества мощности 3 уже сформированы, в них точно не находятся единицы, а значит единицы будут находиться в одном из $x_2 - 1$ множеств мощности 2. Соответственно, суммарно придется потратить $2^n - (x_2 + x_3) + [(x_2 - 1)/2]$ запросов.
- Если первый ответ отличный от 0 был получен на запросах, относящихся к формированию множеств мощности 3, тогда возможно три ситуации.

- 1) Запрос объединял два одноэлементных множества. Соответственно, найдена только одна единица, и сформировано множество мощности 2. Будем считать, что мы отошли на один запрос от нашего плана сначала только сформировать множества мощности 3, а затем множества мощности 2. Иными словами, мы с опережением сформировали одно множество мощности 2 и про него уже все знаем. Также нам известно, что

останется найти еще две единицы. Поэтому если мы запросим оставшиеся запросы, чтобы сформировать x_2 множеств мощности 2 и x_3 множеств мощности 3, и всегда в ответ получим 0, то две единицы будут содержаться в одном из $x_2 - 1$ множеств мощности 2. В этом случае, суммарно будет задано $2^n - (x_2 + x_3) + [(x_2 - 1)/2]$ запроса. Если в процессе формирования этих множеств мы еще раз получим ответ отличный от 0, то мы за $2^n - (x_2 + x_3)$ запросов найдем все единицы, дополнительные запросы не понадобятся.

- 2) Запрос объединял одно- и двухэлементное множество. По ответу на запрос мы однозначно поймем, какое из этих множеств состоит из единиц функции. Если они в двухэлементном множестве, тогда останется найти оставшуюся единицу и она раскроется в течение $2^n - (x_2 + x_3)$ запросов при формировании x_2 множеств мощности 2 и x_3 множеств мощности 3, в силу этого дополнительные запросы не пригодятся и суммарно будет задано всего лишь $2^n - (x_2 + x_3)$ запросов.

Если единица в одноэлементном множестве, тогда очевидно, что осталось найти две единицы и поэтому в принципе можно перестать формировать трехэлементные множества, а продолжить искать две единицы среди оставшихся непокрытых t наборов, где $2^n - 3x_3 \leq t \leq 2^n - 3$, алгоритмом, описанном в доказательстве леммы 1. Но для удобства вычисления числа запросов, доопросим все запросы, необходимые для формирования x_3 множеств мощности 3, и затем вместо формирования x_2 множеств мощности 2, воспользуемся алгоритмом, описанном при доказательстве верхней оценки леммы 1, то есть будем формировать множества мощности 3. Суммарно будет задано $2x_3$ запросов для формирования x_3 множеств мощности 3, а затем по алгоритму леммы 1 $2[(2^n - 3x_3)/3] + ((2^n - 3x_3) \bmod 3) - 1 = 2[2^n/3] - 2x_3 + (2^n \bmod 3) - 1$, что приводит нас к оценке $2x_3 + 2[2^n/3] - 2x_3 + (2^n \bmod 3) - 1 = 2[2^n/3] + (2^n \bmod 3) - 1$. Соответственно, в случае, когда запрос с ответом отличным от 0 объединял одно- и двухэлементное множество, ученик будет вынужден задать не более $\max(2[2^n/3] + (2^n \bmod 3) - 1, 2^n - (x_2 + x_3) + [(x_2 - 1)/2])$. Определим условия на x_2, x_3 , при которых этот максимум был равен $2^n - (x_2 + x_3) + [(x_2 - 1)/2]$. Тогда в целом при таком алгоритме расшифровки с учетом оценки $2^n - (x_2 + x_3) + [x_3/2]$ случая, описанного выше, понадобится

ся не более $2^n - (x_2 + x_3) + [\max(x_2 - 1, x_3)]/2$ запросов, что и докажет оценку данной леммы для ситуации, когда первый ответ отличный от 0 приходит во время первых $2x_3$ запросов. Перейдем доказательству неравенства $2[2^n/3] + (2^n \bmod 3) - 1 \leq 2^n - (x_2 + x_3) + [(x_2 - 1)/2] = x_2 + 2x_3 + [(x_2 - 1)/2]$. Если из обеих частей отбросим общие $2x_3$ запросы, останется лишь доказать неравенство $2[(2^n - 3x_3)/3] + ((2^n - 3x_3) \bmod 3) - 1 \leq x_2 + [(x_2 - 1)/2]$. В силу леммы 6, это неравенство имеет место при $x_2 \geq 4$.

Следовательно можно заключить, что необходимо задать в худшем случае $\max(2[2^n/3] + (2^n \bmod 3) - 1, 2^n - (x_2 + x_3) + [\max(x_2 - 1, x_3)]/2)$. При этом, при $x_2 \geq 4$ число запросов равно $2^n - (x_2 + x_3) + [\max(x_2 - 1, x_3)]/2$, как и утверждается в данной лемме.

При $x_2 < 4$ число запросов равно $\max(2[2^n/3] + (2^n \bmod 3) - 1, 2^n - (x_2 + x_3) + [x_3/2])$. Поскольку $n \geq 4$, то $(2^n - 2x_2) = 3x_3 \geq 10, x_3 \geq 4$, то есть $\max(x_3, x_2 - 1) = x_3$. Осталось показать, что $2[2^n/3] + (2^n \bmod 3) - 1 \leq 2^n - (x_2 + x_3) + [x_3/2]$, а это было сделано в лемме 7.

Следовательно, оценка данной леммы доказана. □

Будем говорить, что x_2, x_3, \dots, x_{2^n} удовлетворяют условию $\mu(n)$, если x_2, x_3, \dots, x_{2^n} — целые неотрицательные числа, а также выполнено

$$2x_2 + 3x_3 + 4x_4 + \dots + 2^n x_{2^n} = 2^n, x_2 + x_3 > 1.$$

Рассмотрим следующую задачу. Пусть n — целое число, не меньшее 4. При условии, что x_2, x_3, \dots, x_{2^n} удовлетворяют условию $\mu(n)$, требуется максимизировать функцию

$$M(x_2, x_3, \dots, x_{2^n}) = \begin{cases} x_2 + x_3 + x_4 + \dots + x_{2^n} - [x_3/2], & \text{если } x_2 \leq x_3, \\ x_2 + x_3 + x_4 + \dots + x_{2^n} - [(x_2 - 1)/2], & \text{если } x_2 > x_3. \end{cases}$$

Эту функцию можно переписать в следующем виде

$$M(x_2, x_3, \dots, x_{2^n}) = x_2 + x_3 + x_4 + \dots + x_{2^n} - [\max(x_2 - 1, x_3)]/2.$$

Лемма 9. Пусть a_2, a_3, \dots, a_{2^n} удовлетворяют условию $\mu(n)$, помимо этого хотя бы для одного $t \in [4, 2^n]$ верно $a_t > 0$. Тогда существуют b_2, b_3, \dots, b_{2^n} , для которых справедливо неравенство $M(a_2, a_3, \dots, a_{2^n}) < M(b_2, b_3, \dots, b_{2^n})$. При этом b_2, b_3, \dots, b_{2^n} удовлетворяют условию $\mu(n)$

и отличаются от a_2, a_3, \dots, a_{2^n} хотя бы в одном члене, то есть $a_j \neq b_j$ для некоторого $j \in [2, 2^n]$.

Доказательство. Рассмотрим два случая в зависимости от четности t .

- 1) Пусть $t = 2 \cdot p$, p — целое. Тогда положим $b_2 = a_2 + p$, $b_3 = a_3$, $b_4 = a_4, \dots, b_{t-1} = a_{t-1}$, $b_t = a_t - 1$, $b_{t+1} = a_{t+1}, \dots, b_{2^n} = a_{2^n}$. Заметим, что b_2, b_3, \dots, b_{2^n} удовлетворяют $\mu(n)$ условию. Рассмотрим значение $M(b_2, b_3, \dots, b_{2^n}) = b_2 + b_3 - [0.5 \cdot \max(b_2 - 1, b_3)] + (b_4 + b_5 + \dots + b_{2^n}) = M(a_2, a_3, \dots, a_{2^n}) + p - 1 + [0.5 \cdot \max(a_2 + p - 1, a_3)] - [0.5 \cdot \max(a_2 - 1, a_3)]$. Осталось показать, что $p - 1 + [0.5 \cdot \max(a_2 + p - 1, a_3)] - [0.5 \cdot \max(a_2 - 1, a_3)] > 0$.

Заметим, что $p \geq 2$, так как $t > 3$.

Если $a_2 + p - 1 \leq a_3$, то $[0.5 \cdot \max(a_2 + p - 1, a_3)] - [0.5 \cdot \max(a_2 - 1, a_3)] = 0$.

Если $a_2 - 1 > a_3$, то $[0.5 \cdot \max(a_2 + p - 1, a_3)] - [0.5 \cdot \max(a_2 - 1, a_3)] = [0.5 \cdot (a_2 + p - 1)] - [0.5 \cdot (a_2 - 1)] > 0$.

Если $a_2 - 1 \leq a_3$, $a_2 + p - 1 \geq a_3$, то $[0.5 \cdot \max(a_2 + p - 1, a_3)] - [0.5 \cdot \max(a_2 - 1, a_3)] = [0.5 \cdot (a_2 + p - 1)] - [0.5 \cdot a_3] \geq 0$.

Приходим к выводу, что $p - 1 + [0.5 \cdot \max(a_2 + p - 1, a_3)] - [0.5 \cdot \max(a_2 - 1, a_3)] > 0$.

- 2) Пусть $t = 2 \cdot p + 1$, p — целое. Заметим, что $t > 3$, поэтому $p \geq 2$. Тогда положим $b_2 = a_2 + (p - 1)$, $b_3 = a_3 + 1$, $b_4 = a_4, \dots, b_{t-1} = a_{t-1}$, $b_t = a_t - 1$, $b_{t+1} = a_{t+1}, \dots, b_{2^n} = a_{2^n}$. Заметим, что b_2, b_3, \dots, b_{2^n} удовлетворяют $\mu(n)$ условию.

Рассмотрим значение $M(b_2, b_3, \dots, b_{2^n}) = b_2 + b_3 - [0.5 \cdot \max(b_2 - 1, b_3)] + (b_4 + b_5 + \dots + b_{2^n}) = M(a_2, a_3, \dots, a_{2^n}) + p - 1 + 1 - 1 + [0.5 \cdot \max(a_2 + p - 2, a_3 + 1)] - [0.5 \cdot \max(a_2 - 1, a_3)]$. Осталось показать, что $p - 1 + [0.5 \cdot \max(a_2 + p - 2, a_3 + 1)] - [0.5 \cdot \max(a_2 - 1, a_3)] > 0$.

Если $a_2 + p - 2 \leq a_3 + 1$, следовательно, $a_2 - 1 \leq a_3$, то $[0.5 \cdot \max(a_2 + p - 2, a_3 + 1)] - [0.5 \cdot \max(a_2 - 1, a_3)] = [0.5 \cdot (a_3 + 1)] - [0.5 \cdot a_3] \geq 0$.

Если $a_2 - 1 > a_3$, то $[0.5 \cdot \max(a_2 + p - 2, a_3 + 1)] - [0.5 \cdot \max(a_2 - 1, a_3)] = [0.5 \cdot (a_2 + p - 2)] - [0.5 \cdot (a_2 - 1)] \geq 0$.

Если $a_2 - 1 \leq a_3$, $a_2 + p - 2 \geq a_3 + 1$, то $[0.5 \cdot \max(a_2 + p - 2, a_3 + 1)] - [0.5 \cdot \max(a_2 - 1, a_3)] = [0.5 \cdot (a_2 + p - 2)] - [0.5 \cdot a_3] \geq 0$.

Приходим к выводу, что $p - 1 + [0.5 \cdot \max(a_2 + p - 2, a_3 + 1)] - [0.5 \cdot \max(a_2 - 1, a_3)] > 0$.

□

Из леммы 9 вытекает следующее следствие.

Следствие 1. Пусть a_2, a_3, \dots, a_{2^n} удовлетворяют условию $\mu(n)$, причем $a_4 + a_5 + \dots + a_{2^n} > 0$. Тогда $M(a_2, a_3, a_4, \dots, a_{2^n}) < M(a_2, a_3, 0, \dots, 0)$.

Лемма 10. Пусть $n \geq 4$, a_2, a_3, \dots, a_{2^n} удовлетворяют условию $\mu(n)$ и выполнено $a_4 = a_5 = \dots = a_{2^n} = 0, a_2 < a_3$. Тогда существуют b_2, b_3, \dots, b_{2^n} , для которых справедливо неравенство $M(a_2, a_3, \dots, a_{2^n}) \leq M(b_2, b_3, \dots, b_{2^n})$. При этом b_2, b_3, \dots, b_{2^n} удовлетворяют условию $\mu(n)$ и верно неравенство $b_3 < a_3$.

Доказательство. Поскольку $a_3 + a_2 > 1, a_2 < a_3$, то либо $a_2 = 0, a_3 \geq 2$, либо $a_2 \geq 1, a_3 \geq 2$. Если $a_3 = 2$, а $a_2 \leq 1$, $3a_3 + 2a_2 \leq 8 < 2^n$ при $n \geq 4$. Следовательно, если $a_2 < a_3$, то $a_3 > 2$. Положим равными $b_2 = a_2 + 3, b_3 = a_3 - 2, b_4 = b_5 = \dots = b_{2^n} = 0$. Очевидно, что $b_2, b_3, b_4, \dots, b_{2^n}$ удовлетворяют условию $\mu(n)$. Рассмотрим значение выражения $M(b_2, b_3, b_4, \dots, b_{2^n}) = b_2 + b_3 - [0.5 \cdot \max(b_2 - 1, b_3)] + (b_4 + \dots + b_{2^n}) = M(a_2, a_3, a_4, \dots, a_{2^n}) + 1 + [0.5 \cdot \max(a_2 + 3 - 1, a_3 - 2)] - [0.5 \cdot a_3]$. Осталось показать, что $1 + [0.5 \cdot \max(a_2 + 2, a_3 - 2)] - [0.5 \cdot a_3] \geq 0$.

Если $a_2 + 2 \leq a_3 - 2$, то $1 + [0.5 \cdot \max(a_2 + 2, a_3 - 2)] - [0.5 \cdot a_3] = 1 + [0.5 \cdot (a_3 - 2)] - [0.5 \cdot a_3] = 1 + [0.5 \cdot a_3] - 1 - [0.5 \cdot a_3] = 0$.

Если $a_2 + 2 > a_3$, то $1 + [0.5 \cdot \max(a_2 + 2, a_3 - 2)] - [0.5 \cdot a_3] = 1 + [0.5 \cdot (a_2 + 2)] - [0.5 \cdot a_3] > 0$.

Если $a_3 - 2 \leq a_2 + 2 < a_3$, то $1 + [0.5 \cdot \max(a_2 + 2, a_3 - 2)] - [0.5 \cdot a_3] = 1 + [0.5 \cdot (a_2 + 2)] - [0.5 \cdot a_3] \geq 1 + [0.5 \cdot a_3] - 1 - [0.5 \cdot a_3] = 0$. □

Из леммы 10 и следствия 1 вытекает следующее следствие.

Следствие 2. Чтобы при $n \geq 4$ найти максимальное значение функции $M(b_2, b_3, \dots, b_{2^n})$, достаточно перебрать удовлетворяющие условию $\mu(n)$ b_2, b_3, \dots, b_{2^n} , для которых выполнены ограничения $b_2 \geq b_3, b_4 = b_5 = \dots = b_{2^n} = 0$.

Лемма 11. Если $n \geq 4$, то максимальное значение функции $M(x_2, x_3, \dots, x_{2^n})$ равно $\lfloor 3/2 \cdot \lfloor 2^n/5 \rfloor + \lfloor (2^n \bmod 5)/2 \rfloor$ и достигается при следующих x_2, x_3, \dots, x_{2^n} :

- $x_2 = \lfloor 2^n/5 \rfloor + 2, x_3 = \lfloor 2^n/5 \rfloor - 1, x_4 = \dots = x_{2^n} = 0$, если $2^n \bmod 5 = 1$,
- $x_2 = \lfloor 2^n/5 \rfloor + 1, x_3 = \lfloor 2^n/5 \rfloor, x_4 = \dots = x_{2^n} = 0$, если $2^n \bmod 5 = 2$,

- $x_2 = \lfloor 2^n/5 \rfloor + 3, x_3 = \lfloor 2^n/5 \rfloor - 1, x_4 = \dots = x_{2^n} = 0$, если $2^n \bmod 5 = 3$,
- $x_2 = \lfloor 2^n/5 \rfloor + 2, x_3 = \lfloor 2^n/5 \rfloor, x_4 = \dots = x_{2^n} = 0$, если $2^n \bmod 5 = 4$.

Доказательство. Согласно следствию 2, для поиска x_2, x_3, \dots, x_{2^n} достаточно перебрать x_2, x_3, \dots, x_{2^n} , удовлетворяющие условию $\mu(n)$ и следующим ограничениям

- 1) $x_4 = x_5 = \dots = x_{2^n} = 0$;
- 2) $x_2 \geq x_3$.

Отсюда следует, что $2^n = 2x_2 + 3x_3$, следовательно $2^n - 2x_2$ должен быть кратен 3.

- 1) Если $2^n \bmod 3 = 1$, то должно выполняться $x_2 \bmod 3 = 2$.
- 2) Если $2^n \bmod 3 = 2$, то должно выполняться $x_2 \bmod 3 = 1$.

Из равенства $2^n = 2x_2 + 3x_3$ следует то, что $x_2 \neq x_3$, а значит $x_2 > x_3$.

Соответственно, x_2 подберем в соответствии с этими требованиями. Учитывая, что $x_3 = (2^n - 2x_2)/3 < x_2$, получаем $x_2 \geq \lfloor 2^n/5 \rfloor$.

Рассмотрим значение функции $M(x_2, x_3, x_4, \dots, x_{2^n}) = x_2 + x_3 - \lfloor 0.5 \cdot \max(x_2 - 1, x_3) \rfloor = \lfloor (x_2 + 1)/2 \rfloor + \lfloor x_3 \rfloor = \lfloor (x_2 + 1)/2 \rfloor + (2^n - 2x_2)/3$. Рассмотрим два случая в зависимости от четности x_2 .

- 1) Если x_2 — нечетное, тогда $M(x_2, x_3, x_4, \dots, x_{2^n}) = (x_2 + 1)/2 + (2^n - 2x_2)/3 = (2^{n+1} - x_2 + 3)/6$. Учитывая ограничение $x_2 \geq \lfloor 2^n/5 \rfloor$, получаем, что максимум $M(x_2, x_3, x_4, \dots, x_{2^n})$ достигается при x_2 наименьшем целом числе, неменьшим $\lfloor 2^n/5 \rfloor$ и дающий остаток от деления на 3 равный $3 - 2^n \bmod 3$.
- 2) Если x_2 — четное, тогда $M(x_2, x_3, x_4, \dots, x_{2^n}) = x_2/2 + 1 + (2^n - 2x_2)/3 = (2^{n+1} - x_2 + 6)/6$. Учитывая ограничение $x_2 \geq \lfloor 2^n/5 \rfloor$, получаем, что максимум $M(x_2, x_3, x_4, \dots, x_{2^n})$ достигается при x_2 наименьшем целом числе, неменьшим $\lfloor 2^n/5 \rfloor$ и дающий остаток от деления на 3 равный $3 - 2^n \bmod 3$.

Заметим, что если $n \bmod 4 = 0$, то $2^n \bmod 3 = 1, 2^n \bmod 5 = 1$. Если $n \bmod 4 = 1$, то $2^n \bmod 3 = 2, 2^n \bmod 5 = 2$. Если $n \bmod 4 = 2$, то $2^n \bmod 3 = 1, 2^n \bmod 5 = 4$. Если $n \bmod 4 = 3$, то $2^n \bmod 3 = 2, 2^n \bmod 5 = 3$.

Пусть $2^n = 3q + 1 = 5t + r$, где q, t, r — целые неотрицательные числа, $r \in \{1, 4\}$. Соответственно, $\lfloor 2^n/5 \rfloor = t + 1$.

Рассмотрим два случая относительно остатков от деления 2^n на 5.

- $r = 1$. Заметим, что t — нечетное, так как $5t = 2^n - 1$. Теперь рассмотрим возможные остатки от деления t на 3. Если $t \bmod 3 = 1$, то $2^n = 5(3p + 1) + 1 = 3 \cdot 5p + 6 \neq 3q + 1$. Если $t \bmod 3 = 2$, то $2^n = 5(3p + 2) + 1 = 3 \cdot 5p + 11 \neq 3q + 1$. Если $t \bmod 3 = 0$, то $2^n = 5 \cdot 3p + 1 = 3 \cdot 5p + 1 = 3q + 1$. Отсюда следует, что $t \bmod 3 = 0$.

Тогда $x_2 = t + 2, x_3 = (2^n - 2x_2)/3 = (3t + r - 4)/3 = t - 1$. Соответственно, $M(x_2, x_3, x_4, \dots, x_{2^n}) = 2t + 1 - [(t + 2 - 1)/2] = 2t + 1 - (t + 1)/2 = (3t + 1)/2 =]3/2 \cdot [2^n/5][$.

- $r = 4$. Обратим наше внимание, что t — четное, так как $5t = 2^n - 4$. Теперь рассмотрим возможные остатки от деления t на 3. Если $t \bmod 3 = 1$, то $2^n = 5(3p + 1) + 4 = 3 \cdot 5p + 9 \neq 3q + 1$. Если $t \bmod 3 = 2$, то $2^n = 5(3p + 2) + 4 = 3 \cdot 5p + 14 \neq 3q + 1$. Если $t \bmod 3 = 0$, то $2^n = 5 \cdot 3p + 4 = 3q + 1$. Отсюда следует, что $t \bmod 3 = 0$.

Тогда $x_2 = t + 2, x_3 = (2^n - 2x_2)/3 = (3t + r - 4)/3 = t$. Соответственно, $M(x_2, x_3, x_4, \dots, x_{2^n}) = 2t + 2 - [(t + 2 - 1)/2] = 2t + 2 - t/2 = 3t/2 + 2 =]3/2 \cdot [2^n/5][+2$.

Пусть $2^n = 3q + 2 = 5t + r$, где q, t, r — целые неотрицательные числа, $r \in \{2, 3\}$. Соответственно, $]2^n/5[= t + 1$.

Рассмотрим два случая относительно остатков на деление 2^n на 5.

- $r = 2$. Заметим, что t — четное, так как $5t = 2^n - 2$. Теперь рассмотрим возможные остатки от деления t на 3. Если $t \bmod 3 = 1$, то $2^n = 5(3p + 1) + 2 = 3 \cdot 5p + 7 \neq 3q + 2$. Если $t \bmod 3 = 2$, то $2^n = 5(3p + 2) + 2 = 3 \cdot 5p + 12 \neq 3q + 2$. Если $t \bmod 3 = 0$, то $2^n = 5 \cdot 3p + 2 = 3 \cdot 5p + 2 = 3q + 2$. Отсюда следует, что $t \bmod 3 = 0$.

Тогда $x_2 = t + 1, x_3 = (2^n - 2x_2)/3 = (3t + r - 2)/3 = t$. Соответственно, $M(x_2, x_3, x_4, \dots, x_{2^n}) = 2t + 1 - [(t + 1 - 1)/2] = 2t + 1 - t/2 = 3t/2 + 1 =]3/2 \cdot [2^n/5][+1$.

- $r = 3$. Также заметим, что t — нечетное, так как $5t = 2^n - 3$. Теперь рассмотрим возможные остатки от деления t на 3. Если $t \bmod 3 = 1$, то $2^n = 5(3p + 1) + 3 = 3 \cdot 5p + 8 = 3q + 2$. Если $t \bmod 3 = 2$, то $2^n = 5(3p + 2) + 3 = 3 \cdot 5p + 13 \neq 3q + 2$. Если $t \bmod 3 = 0$, то $2^n = 5 \cdot 3p + 3 \neq 3q + 2$. Отсюда следует, что $t \bmod 3 = 1$.

Тогда $x_2 = t + 3, x_3 = (2^n - 2x_2)/3 = (3t + r - 6)/3 = t - 1$. Соответственно, $M(x_2, x_3, x_4, \dots, x_{2^n}) = 2t + 2 - [(t + 3 - 1)/2] = 2t + 2 - (t + 1)/2 = (3t + 3)/2 =]3/2 \cdot [2^n/5][+1$.

□

Лемма 12. Пусть $n \geq 7$, q, r — целые положительные числа из выражения $2^n = 5 \cdot q + r$, $r \in [1, 4]$. Тогда справедливо неравенство

$$2^n - 1.5 \cdot q - [0.5 \cdot r] < 3 \cdot 2^{n-2} - 2.$$

Доказательство. Рассмотрим разность $2^n - 1.5 \cdot q - [0.5 \cdot r] - 3 \cdot 2^{n-2} + 2 = 2^{n-2} - 3/10 \cdot (2^n - r) + 2 - [0.5 \cdot r] = 1/10 \cdot (10 \cdot 2^{n-2} - 12 \cdot 2^{n-2} + 3r + 20 - 10 \cdot [0.5 \cdot r]) = 1/10 \cdot (-2^{n-1} + 3r - 10 \cdot [0.5 \cdot r] + 20)$. Последнее выражение отрицательно при $n \geq 7$ независимо от того, какое значение принимает r . □

Лемма 13. При $n \geq 4$ справедлива следующая верхняя оценка

$$\varphi_C(n, 3) \leq 2^n -]3/2 \cdot [2^n/5][- [(2^n \bmod 5)/2].$$

Доказательство. Подставим в оценку леммы 8 приводимые в лемме 11 значения x_2, x_3 , а x_4, \dots, x_{2^n} положим равными 0. Заметим, что они удовлетворяют условию теоремы 7. Причем, получаемая верхняя оценка равна $2^n - M(x_2, x_3, 0, \dots, 0)$. Согласно лемме 11, при выбранных таким образом x_2, x_3 достигается минимум $2^n -]3/2 \cdot [2^n/5][- [(2^n \bmod 5)/2]$. □

Лемма 14. Если для расшифровки функции $f \in F(n, 3)$ ученик использует алгоритм, покрывающий не все наборы, тогда для однозначного восстановления функции ему потребуется как минимум $3 \cdot 2^{n-2} - 2$ запросов.

Доказательство. Пусть $2^n = 4x + 4$, где x — целое. Пусть алгоритмом расшифровки не покрыто $r > 0$ наборов, при этом задано y запросов и ученик по ответам на свои запросы однозначно может восстановить загаданную функцию. Заметим, что $3x + 1 = 3 \cdot 2^{n-2} - 2$. В роли учителя на все первые $3x + 1$ запросов будем отвечать 0. Ответы на последующие запросы определим позднее при рассмотрении разных случаев.

Обратим внимание, что возможен один из двух случаев: либо все непокрытые наборы нули функции, либо все они являются ее единицами. Поскольку если бы часть непокрытых наборов была нулями, а оставшиеся непокрытые наборы единицами, то невозможно было без дополнительных запросов понять, какие из них и есть единицы. После y запросов $2^n - r$ наборов распались на классы эквивалентности, то есть подразбились на множества наборов, про которые известно, что значение функции на всех элементах одного множества одинаковое. Про все

множества мощности не меньше четырех ученик сразу понял, что в них нет единиц функции. Рассмотрим всевозможные значения r .

- 1) $r \geq 4$. Учитель может спрятать все единицы функции среди этих четырех непокрытых наборов. В силу этого, ученику недостаточно y наборов для однозначного восстановления функции, соответственно такой случай невозможен.
- 2) $r = 3$. Если среди множеств имеется множество мощности 3, тогда ученик не поймет единицы лежат в этом множестве или среди непокрытых наборов. Следовательно, множеств мощности 3 не должно быть. Если среди множеств имеется множество мощности 2, тогда ученик поймет, что две единицы лежат в каком-то из множеств мощности 2, но не поймет какой из непокрытых наборов является третьей единицей. Соответственно, множеств мощности 2 также не должно быть. Приходим к выводу, что раз ученик однозначно восстановил функцию после y запросов, значит все множества имеют мощность не меньше 4, а непокрытые наборы и есть единицы функции. Согласно лемме 5, для объединения $4x + 1$ наборов во множества мощности не меньшей 4, необходимо не менее $3x + 1$ запросов, иными словами, $y \geq 3 \cdot 2^{n-2} - 2$.
- 3) $r = 2$. Если среди множеств имеется множество мощности 2, тогда ученик поймет, что две единицы лежат в каком-то из множеств мощности 2, но не поймет какой из непокрытых наборов является третьей единицей. Соответственно, множеств мощности 2 также не должно быть. Если среди множеств имеются хотя бы два множества мощности 3, тогда ученик не определит, в каком из этих множеств лежат единицы функции. Исходя из этого возможные следующие два случая.
 - Имеется ровно одно множество мощности 3 и несколько множеств мощности не меньшей 4. На образование множества мощности 3 необходимо 2 запроса, а на покрытие оставшихся $4x - 1$ наборов, согласно лемме 5, понадобится $3(x - 1) + 3$ запроса. Отсюда следует, что $y \geq 3x + 2 \geq 3 \cdot 2^{n-2} - 2$.
 - Отсутствуют множества мощности 3 и все наборы разбились на множества мощности не меньшей 4. Согласно лемме 5, для этого необходимо $3x + 2$ запроса. На первые $3x + 1$ запросов учитель отвечает 0. Если же ученик задает $(3x + 2)$ -й запрос

и учитель видит, что если ответ на этот запрос будет равен 0, то покрытыми станут $4x + 2$ набора и они распадутся на множества мощности не меньшей 4, тогда в этом случае учитель отвечает не 0, раскрывая единицы в той компоненте запроса, которая относится ко множеству меньшей мощности. Тем самым, учитель гарантирует, что не обманывал ученика и действительно его ответы соответствуют какой-то функции из $F(n, 3)$.

Рассмотреть ответ на $(3x + 2)$ -й запрос необходимо было лишь для последней цели, а так и этот случай демонстрирует, что ученику потребуется как минимум $3x + 1$ запросов, то есть $y \geq 3 \cdot 2^{n-2} - 2$.

4) $r = 1$. Если имеется и множество мощности 2, и множество мощности 3, тогда ученик не поймет, единицы функции лежат во множестве мощности 3 или во множестве мощности 2 и непокрытом наборе. Если имеется несколько множеств мощности 3, тогда ученик не поймет, в каком из них лежат единицы функции. Если имеются несколько множеств мощности 2, тогда ученик может понять, что одна единица — это непокрытый набор, но не сможет определить, в каком из множеств мощности 2 лежат оставшиеся единицы. Следовательно, возможны следующие три случая.

- Имеется ровно одно множество мощности 2, нет множеств мощности 3 и имеются несколько множеств мощности не меньшей 4. На образование множества мощности 2 необходим 1 запрос, а на покрытие оставшихся $4x + 1$ наборов, согласно лемме 5, понадобится $3x + 1$ запросов. Отсюда следует, что $y \geq 3x + 2 \geq 3 \cdot 2^{n-2} - 2$.
- Имеется ровно одно множество мощности 3, нет множеств мощности 2 и имеются несколько множеств мощности не меньшей 4. На образование множества мощности 3 необходимо 2 запроса, а на покрытие оставшихся $4x$ наборов, согласно лемме 3, понадобится $3x$ запросов. Отсюда следует, что $y \geq 3x + 2 \geq 3 \cdot 2^{n-2} - 2$.
- Отсутствуют множества мощности 2 и 3 и все наборы разбились на множества мощности не меньшей 4. Согласно лемме 5, для этого необходимо $3x + 3$ запроса. На первые $3x + 2$ запросов учитель отвечает 0. Если же ученик задает $(3x + 3)$ -й

запрос и учитель видит, что если ответ на этот запрос будет равен 0, то покрытыми станут $4x + 3$ набора и они распадутся на множества мощности не меньшей 4, тогда в этом случае учитель отвечает не 0, раскрывая единицы в той компоненте запроса, которая относится ко множеству меньшей мощности. Тем самым, учитель гарантирует, что не обманывал ученика и действительно его ответы соответствуют какой-то функции из $F(n, 3)$. Рассмотреть ответ на $(3x + 3)$ -й запрос необходимо было лишь для последней цели, а так и этот случай демонстрирует, что ученику потребуется как минимум $3x + 1$ запросов, то есть $y \geq 3 \cdot 2^{n-2} - 2$.

□

Лемма 15. Пусть $2^n = 5 \cdot q + r$, $r \in [1, 4]$, и для расшифровки функции $f \in F(n, 3)$ ученик использует алгоритм расшифровки A , покрывающий все 2^n наборов. Тогда ученик задаст не менее $3 \cdot 2^{n-2} - 2$ запросов, если $N(A, C(A), 2) + N(A, C(A), 3) \leq 1$.

Доказательство. Пусть $2^n = 4x + 4$. Рассмотрим три случая.

1) $N(A, C(A), 2) + N(A, C(A), 3) = 0$

После $C(A)$ запросов образовались множества мощности строго больше 3. Согласно лемме 3, для этого потребуется не менее $3 \cdot 2^{n-2}$ запросов.

2) $N(A, C(A), 2) = 1, N(A, C(A), 3) = 0$

После $C(A)$ запросов образовались множества мощности строго больше 3 и ровно одно множество мощности 2. Согласно лемме 5, для образования множеств мощности не меньшей 4 из $2^n - 2 = 4x + 2$ потребуется не менее $3x + 2 = 3 \cdot 2^{n-2} - 1$ запросов.

3) $N(A, C(A), 2) = 0, N(A, C(A), 3) = 1$

После $C(A)$ запросов образовались множества мощности строго больше 3 и ровно одно множество мощности 3. Согласно лемме 5, для образования множеств мощности не меньшей 4 из $2^n - 3 = 4x + 1$ потребуется не менее $3x + 1 = 3 \cdot 2^{n-2} - 2$ запросов.

□

Лемма 16. Пусть $n \geq 7$. Тогда верно неравенство

$$\varphi_C(n, 3) \geq 2^n - \lfloor 3/2 \cdot \lfloor 2^n/5 \rfloor - \lfloor (2^n \bmod 5)/2 \rfloor.$$

Доказательство. Из лемм 12, 14 вытекает то, что для расшифровки функции с тремя единицами невыгодно использовать алгоритм расшифровки, непокрывающий все наборы. Поскольку верхняя оценка леммы 13 утверждает, что $\varphi_C(n, 3) \leq 2^n - \lfloor 3/2 \cdot \lfloor 2^n/5 \rfloor - \lfloor (2^n \bmod 5)/2 \rfloor \leq 2^n - 1.5 \cdot \lfloor 2^n/5 \rfloor - \lfloor (2^n \bmod 5)/2 \rfloor$.

Соответственно, чтобы задать как можно меньше запросов, ученик вынужден использовать алгоритм, покрывающий все наборы. Пусть первый запрос, после ответа на который окажутся покрытыми все 2^n наборов, имеет номер w . Пусть после w запросов исходные 2^n одноэлементных множеств — наборов объединятся в x_2 множеств мощности 2, x_3 множеств мощности 3, ..., x_{2^n} множеств мощности 2^n , иными словами верно соотношение $2^n = 2x_2 + 3x_3 + 4x_4 + \dots + 2^n \cdot x_{2^n}$. На первые w запросов ученика (a, b) учитель будет отвечать следующим образом:

- На первые $w - 1$ запросов учитель отвечает 0.
- Ответ на w -й запрос определяется следующим образом.

Если $x_2 \leq x_3$, учитель отвечает 0. Если $x_2 > x_3$, учитель отвечает

- 1) 1, если a — один из непокрытых наборов, покрываемых запросом (a, b) ,
- 2) -1 , в противном случае.

Из леммы 15 следует $x_2 + x_3 > 1$.

Спустя w запросов ученик понимает, что во всех $x_4 + x_5 + \dots + x_{2^n}$ множествах мощности строго больше трех нет единиц.

Если на w -й запрос ученик в ответ получит 0, тогда он поймет, что все единицы находятся в одном из множеств мощности 3. Для определения нужного множества ученику потребуется $\lfloor x_3/2 \rfloor$ запросов.

Если на w -й запрос ученик в ответ получит отличный от 0, тогда он найдет в точности одну единицу. Не нарушая общности будем считать, что на w -й запрос (a, b) ученик получил в ответ 1. Тогда он понимает, что a — единица, а множество, представителем которого является набор b , состоит полностью из нулей. Следовательно, ученику остается найти оставшиеся 2 единицы и они очевидно лежат в каком-то из множеств мощности 2. Для определения нужного множества ученику потребуется

- $\lfloor x_2/2 \rfloor$ запросов, если и b был уже покрыт первыми $w - 1$ запросами, иными словами, суммарно множество с представителем в b и набор a образуют множество мощности хотя бы 3,
- $\lfloor (x_2 - 1)/2 \rfloor$, если и a , и b оба не были покрыты первыми $w - 1$ запросами, а значит суммарно после w запросов образуют множество мощности 2, соответственно, про одно из x_2 множеств ученик полностью знает значение функции на каждом его элементе, поэтому остается искать среди меньшего числа множеств множество с двумя единицами.

Заметим, что $w = x_2 + 2x_3 + 3x_4 + \dots + (i - 1)x_i + \dots + (2^n - 1)x_{2^n} = 2^n - (x_2 + x_3 + \dots + x_{2^n})$, поскольку для образования множества мощности i необходимо $i - 1$ запросов. Соответственно, если $x_2 \leq x_3$ ученик задаст не менее $2^n - (x_2 + x_3 + \dots + x_{2^n}) + \lfloor x_3/2 \rfloor$, а если $x_2 > x_3$, то вынужден будет задать не менее $2^n - (x_2 + x_3 + \dots + x_{2^n}) + \lfloor (x_2 - 1)/2 \rfloor$ запросов.

Цель ученика подобрать такие x_2, x_3, \dots, x_{2^n} , чтобы минимизировать это количество запросов, а значит максимизировать величину $(x_2 + x_3 + \dots + x_{2^n}) - \lfloor x_3/2 \rfloor$ при $x_2 \leq x_3$ и $(x_2 + x_3 + \dots + x_{2^n}) - \lfloor (x_2 - 1)/2 \rfloor$ при $x_2 > x_3$, соответственно максимизировать функцию $M(x_2, x_3, \dots, x_{2^n}) = (x_2 + x_3 + \dots + x_{2^n}) - \lfloor \max(x_2 - 1, x_3) / 2 \rfloor$. Согласно лемме 11, максимальное значение этой функции равно $\lceil 3/2 \cdot \lfloor 2^n/5 \rfloor \rceil + \lfloor (2^n \bmod 5)/2 \rfloor$, а значит ученик вынужден задать не менее $2^n - \lceil 3/2 \cdot \lfloor 2^n/5 \rfloor \rceil - \lfloor (2^n \bmod 5)/2 \rfloor$. □

Доказательство теоремы 6 следует из лемм 13 и 16.

8. Анализ результатов

Исходя из результатов, можно сделать вывод, что помимо классов $\{x\}$, $\{x, \bar{x}\}$, приводимых в [10], еще и класс функций фиксированного веса таков, что сложность его расшифровки запросами на сравнение строго меньше сложности расшифровки запросами на значение.

Также показано, что с точки зрения сложности расшифровки для класса функций фиксированного веса лучшими являются запросы на расширенную эквивалентность. Между тем использовать запросы на значение и запросы на эквивалентность для этого класса нецелесообразно в силу того, что сложность расшифровки этими типами запросов схожа с восстановлением всего вектора значений функций.

9. Благодарность

Автор выражает благодарность своему научному руководителю — д.ф.м.н., профессору Э. Э. Гасанову за постановку задачи и помощь в работе.

Список литературы

- [1] D. Angluin, “Queries and Concept Learning”, *Machine Learning*, **2**:0885-6125 (1988), 319–342.
- [2] D. Angluin, “Queries Revisited”, *Theoretical Computer Science*, **313**:2 (2001), 175–194.
- [3] A. Bertoni, N. Cesa-Bianchi, G. Fiorino, “Efficient learning with equivalence queries of conjunctions of modulo functions”, *Information Processing Letters*, **56**:1 (1995), 15–17.
- [4] W. Maass, G. Turan, “How Fast Can A Threshold Gate Learn?”, *Computational Learning Theory and Natural Learning Systems*, **1** (1990), 381–414.
- [5] Вороненко А. А., Чистиков Д. В., *Проблемы теоретической кибернетики. Материалы XVI Международной конференции*, Издательство Нижегородского университета, 2011.
- [6] Гасанов Э. Э., “Расшифровка линейных функций ранжирования”, *Материалы XI Международного семинара «Дискретная математика и ее приложения» (Москва, 18-23 июня 2012 г.)*, 2012, 332–334.
- [7] Хегай С. И., “Расшифровка полиномиальных функций ранжирования”, *Интеллектуальные системы*, **19**:1 (2015), 213–230.
- [8] А. А. Абдель Маджид, “О сложности восстановления частичного порядка”, *Интеллектуальные системы*, **20**:4 (2016), 5–10.
- [9] Быстрыгова А. В., “Параметро-эффективная расшифровка булевых функций из замкнутых классов Поста”, *Дискретная математика*, **31**:2 (2019), 34–58.
- [10] Быстрыгова А. В., “Запросы на сравнение в задаче параметро-эффективной расшифровки булевых функций”, *Интеллектуальные системы*, **23**:4 (2019), 115–124.
- [11] Калачев Г.В., “Оценки мощности плоских схем, реализующих функции с ограниченным числом единиц”, *Интеллектуальные системы*, **21**:1 (2017), 1–51.

Learning of Boolean fixed-weight functions Bistrigova A.V.

This paper is concerned with the learning complexity of Boolean fixed-weight functions using membership queries, comparison queries, equivalence queries, and extended equivalence queries. Moreover, it is allowed to use only one type of queries during learning. This paper

gives exact values of learning complexity for all the types of queries besides comparison queries. The paper presents the upper bound on learning complexity for comparison queries. In addition, the paper demonstrates that the upper bound is equal to the lower bound for the 1-, 2-, 3-weight functions.

Keywords: Boolean fixed-weight functions, membership queries, comparison queries, equivalence queries, extended equivalence queries, exact learning.

Поиск ближайшего соседа на прямой с помощью клеточного автомата с локаторами

Васильев Д.И.¹

Аннотация

В данной статье 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 статье показано, что эта возможность позволяет уменьшить сложность рассматриваемой задачи с линейной до логарифмической по сравнению с классической моделью клеточного автомата.

Ключевые слова: клеточные автоматы, однородные структуры, поиск ближайшей точки.

1. Введение.

Клеточные автоматы являются дискретными математическими моделями широкого класса реальных систем вместе с протекающими в них процессами.

Понятие клеточного автомата возникло в результате усовершенствования модели Дж. фон Неймана [1, 2, 3], предложенной им для описания процессов самовоспроизведения в биологии и технике. Эта модель развивалась и использовалась в работах А. Беркса [4], Э. Мура [5], В. Б. Кудрявцева, А. С. Подколзина, А. А. Болотова [6] и других исследователей [7, 8, 9].

Клеточный автомат — это математический объект с дискретными пространством и временем. Пространство поделено на клетки, в каждой

¹Васильев Денис Игоревич — старший консультант, KORUS Consulting LLC, e-mail: denis.vasilev.igor@gmail.com.

Denis Igorevich Vasilev — Senior Consultant, KORUS Consulting LLC.

из которых находится элементарный автомат. Такты времени задаются натуральными числами. Состояние автомата каждой пространственной клетки определяется очень простыми правилами взаимодействия. Эти правила предписывают изменения состояния автомата каждой клетки в следующем такте времени в ответ на текущее состояние автоматов соседних клеток.

В работе Гасанова Э.Э. [10] было введено понятие клеточного автомата с локаторами, которое отличается от понятия обычного клеточного автомата тем, что допускает передачу информации не только между соседними ячейками, но и на любое расстояние, посредством передачи сигнала в эфир. В работе рассматривается применение этой модели к одномерной задаче поиска ближайшего соседа: на \mathbb{Z}^1 произвольно отмечены особая точка, называемая "центральной" и две других, которые будем называть "общими"; нужно понять, какая из общих точек находится ближе к центральной. Классическая модель клеточного автомата решает эту задачу за линейное (по минимуму расстояний между центральной и общими точками) время. В данной работе будет показано, что модель автомата с локаторами может решать эту задачу за логарифмическое время.

Автор выражает благодарность профессору Э.Э.Гасанову за постановку задачи и к.ф.-м.н. Г.В.Калачеву за ценные замечания и предложения.

2. Описание задачи и формулировка результатов.

В работе Гасанова Э.Э. [10] введено понятие клеточного автомата с локаторами. Здесь приведем это понятие, сузив его на одномерный случай.

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. Таким образом, в случае \mathbb{R}^1 телесным углом из точки a может быть либо луч, выпущенный из a в сторону $a + \epsilon$ или $a - \epsilon$ ($\epsilon > 0$), либо объединение этих двух лучей. Луч соответствующий направлению $a + \epsilon$ будем обозначать как (1) , $a - \epsilon$ — (-1) , а объединение этих лучей — как Ω (скобки в обозначениях (1) и (-1) акцентируют внимание на том, что соответствующая запись должна восприниматься скорее как вектор, нежели как скаляр).

Клеточным автоматом с локаторами на \mathbb{Z}^1 называется восьмерка $\sigma = (\mathbb{Z}^1, E_n, V, E_q, +, L, \varphi, \psi)$, где \mathbb{Z}^1 — множество целых чисел, $E_n = \{0, 1, \dots, n-1\}$, $V = (\alpha_1, \dots, \alpha_{h-1})$ — подмножество $\{(1); (-1)\}$, $E_q = \{0, 1, \dots, q-1\}$, $+$ — коммутативная полугрупповая операция с нейтральным элементом, заданная на E_q , $L = (\nu_1, \dots, \nu_m)$ — упорядоченный набор попарно различных телесных углов в \mathbb{R}^1 с вершиной в начале координат (в нашем случае — подмножество $\{(1); (-1); \Omega\}$), φ — функция, зависящая от переменных $x_0, x_1, \dots, x_{h-1}, z_1, \dots, z_m$, $\varphi : E_n^h \times E_q^m \rightarrow E_n$, $\varphi(0, \dots, 0) = 0$, ψ — функция, зависящая от переменных $x_0, x_1, \dots, x_{h-1}, z_1, \dots, z_m$, $\psi : E_n^h \times E_q^m \rightarrow E_q$. Элементы множества \mathbb{Z}^1 называются *ячейками* клеточного автомата σ ; элементы множества E_n называются *состояниями ячейки* клеточного автомата σ ; набор V называется *шаблоном соседства* клеточного автомата σ ; элементы множества E_q называются *сигналами вещания*; набор L называется *шаблон локаторов* клеточного автомата σ ; функция φ называется *локальной функцией переходов* автомата σ ; функция ψ называется *функцией вещания* автомата σ ; переменные x_0, x_1, \dots, x_{h-1} принимают значения из E_n , переменные z_1, \dots, z_m принимают значения из E_q . Состояние 0 интерпретируется как *состояние покоя*, а условия $\varphi(0, \dots, 0) = 0$ и $\psi(0, \dots, 0) = 0$ — как условия 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 φ и функции вещания ψ соответственно x_0, x_1, \dots, x_{h-1} и z_1, \dots, z_m . Это соответствие можно сделать более явным, если индексировать переменные функций φ и ψ самими векторами и телесными углами, т.е. считать, что локальная функция переходов φ и функция вещания ψ зависят от переменных $x_0, x_{\alpha_1}, \dots, x_{\alpha_{h-1}}, z_{\nu_1}, \dots, z_{\nu_m}$, здесь индекс первой переменной есть нулевой вектор $0 = (0, \dots, 0) \in \mathbb{Z}^k$. 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 ин-

дексах внешние круглые скобки у векторов. Например, если $k = 1$, $n = 2$, $q = 2$ и $V = \{(-1), (1)\}$, $L = \{\Omega, (-1), (1)\}$, то пример локальной функции переходов может выглядеть так: $\varphi = x_{-1} \& z_{\Omega} \vee x_1 \& z_1$.

Если $\alpha \in \mathbb{Z}^1$, ν — телесный угол с вершиной в начале координат, то через $\nu(\alpha)$ обозначим телесный угол, полученный параллельным переносом угла ν в точку α .

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

Состоянием клеточного автомата с локаторами σ назовем пару (e, f) , где e — произвольная функция, определенная на множестве \mathbb{Z}^1 , принимающая значения из E_q , называемая *состоянием эфира*, f — произвольная функция, определенная на множестве \mathbb{Z}^1 , принимающая значения из E_n и называемая *распределением состояний клеточного автомата с локаторами* σ . Такую функцию можно интерпретировать как некую мозаику, возникающую в 1-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 Σ .

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

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

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

Заметим, что значение локатора не всегда определено. Например, если полугруппа представляет собой множество $\{0; 1\}$ с операцией двоичного сложения, и все ячейки отправляют в эфир 1, то значения локаторов формально будут представлять из себя бесконечную сумму единиц, что некорректно. В дальнейшем мы будем рассматривать лишь те пары автоматов и их начальных состояний, на каждом такте которых значения всех локаторов определены корректно.

На множестве Σ определим *глобальную функцию переходов* Φ клеточного автомата с локаторами σ , полагая $\Phi(e, f) = (e', f')$, где

$(e, f), (e', f') \in \Sigma$ и для любой ячейки $\alpha \in \mathbb{Z}^1$ выполняются тождества

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

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

Содержательная интерпретация отображения Φ 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 ψ и φ одинаково для всех ячеек.

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

Сформулируем задачу поиска ближайшего соседа на прямой. Пусть на пространстве \mathbb{Z}^1 задано начальное состояние I клеточного автомата, удовлетворяющее следующим критериям:

- 1) Любой ячейке присвоено одно из трех состояний $\{q_S; q_{C_0}, *\}$.
- 2) Есть лишь одна ячейка, которой присвоено состояние q_{C_0} .
- 3) Есть лишь конечное и непустое множество ячеек, которым присвоено состояние q_S .

Решением задачи поиска ближайшего соседа, соответствующей начальному состоянию I назовем состояние автомата, удовлетворяющее следующим критериям:

- 1) Ячейке, которой в I было присвоено состояние q_{C_0} присвоено состояние q_{CF} .
- 2) Ближайшей к ячейке в состоянии q_{CF} ячейке из тех, которым в I было присвоено состояние q_S , присвоено состояние q_{LE} , если она находится слева от ячейки в состоянии q_{CF} , и q_{RE} — если справа. Если таких ячеек две, то правой присваивается состояние $*$, а левой — q_{LE} .

- 3) Ячейкам, которые находятся между ячейками в состояниях q_{CF} и q_{LE} присваивается состояние q_{LF} . Ячейкам, которые находятся между ячейками в состояниях q_{CF} и q_{RE} присваивается состояние q_{RF} .
- 4) Остальные ячейки находятся в состоянии $*$.

Определим, что клеточный автомат с локаторами σ решает задачу поиска ближайшего соседа, если его начальное состояние удовлетворяет условиям, описанным выше, и его финальное состояние существует и соответствует решению задачи поиска ближайшего соседа для его начального состояния. Существует клеточный автомат σ с 25 состояниями и с мощностью алфавита вещания 12, который решает задачу поиска ближайшего соседа за время, не превосходящее $\log_2 s + 7$, где s — расстояние от центральной ячейки с начальным состоянием q_{C_0} до её ближайшего соседа с начальным состоянием q_S .

3. Формальное описание автомата

Рассмотрим клеточный автомат $\sigma = (\mathbb{Z}^1, E_n, V, E_q, +, L, \varphi, \psi)$, где $V = \{(1), (-1)\}$, $E_q = \{0, 1\}^2 \times \{0, 1, 2\}$, а $L = (\nu_{-1}, \nu_1)$, где ν_{-1}, ν_1 — вырожденные телесные углы, соответствующие векторам (-1) и (1) .

Полугрупповую операцию на E_q определим следующим образом:
 $(a_1, b_1, c_1) + (a_2, b_2, c_2) = (a_1 + a_2, \max(b_1, b_2), \max(c_1, c_2))$

Пусть множество состояний $E_n = \{q_{\bar{C}C}; q_{\bar{C}C}^<; q_{\bar{C}C}^>; q_S; q_{C_0}; q_L; q_R; q_{C_{W_2}}; q_{C_{W_1}}; q_{C_{L_1}}; q_{C_{R_1}}; q_{LF}; q_{RF}; q_{LE}; q_{RE}; q_{CF}; q_*^L; q_*^R; q_1^L; q_1^R; q_0^L; q_0^R; q_{L_*}; q_{R_*}; *\}$

Здесь мы позволили себе раздать состояниям произвольные имена, а не $0, 1, 2, \dots$ как в определении. Это сделано для упрощения изложения. Состояние покоя при этом соответствует состоянию $*$. Автомат будет сконструирован таким образом, что лишь конечное число ячеек будет находиться в состоянии, отличном от $*$ на каждом из тактов. Учитывая, что операция $+$ обладает свойством $(0, 0, 0) + (0, 0, 0) = (0, 0, 0)$, можно сделать вывод, что значение любого локатора определяется лишь конечным количеством ненулевых слагаемых, то есть корректно определено.

Опишем функции φ и ψ для каждого состояния автомата:

q_k^L и q_k^R , $k \in 0; 1$ — ключевые состояния алгоритма. Автомат сконструирован таким образом, что если изначально есть некоторое количество ячеек в состоянии q_1^L , идущих подряд, то на каждом следующем

такте каждая вторая ячейка с $k = 1$ перейдет в состояние q_0^L , а остальная часть — в состояние q_1^L . Этот факт будет формально доказан в другом разделе статьи, но суть заключается в том, что такое поведение позволяет побитово передавать двоичную запись длины соответствующего отрезка в эфир.

$$\varphi(q_k^L, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = \begin{cases} q_{(k \wedge z_{\nu-1}^1)}^L, & \text{если } z_{\nu_1}^3 = 0 \\ q_{LF}, & \text{если } z_{\nu_1}^3 = 1 \\ * & \text{в остальных случаях} \end{cases}, \quad (4)$$

$$\psi(q_k^L, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = (k \wedge z_{\nu-1}^1, k \wedge z_{\nu-1}^1, 0), \quad (5)$$

$$\varphi(q_k^R, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = \begin{cases} q_{(k \wedge z_{\nu-1}^1)}^R, & \text{если } z_{\nu-1}^3 = 0 \\ q_{RF}, & \text{если } z_{\nu-1}^3 = 2 \\ * & \text{в остальных случаях} \end{cases}, \quad (6)$$

$$\psi^1(q_k^R, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = (k \wedge z_{\nu-1}^1, k \wedge z_{\nu-1}^1, 0). \quad (7)$$

q_{C_0} — это начальное состояние центральной ячейки. Ячейка в этом состоянии посылает сигнал $(0, 0, 1)$, чтобы остальные ячейки смогли определиться: они справа или слева от центра.

$$\varphi(q_{C_0}, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = q_{C_{W_1}}, \quad (8)$$

$$\psi(q_{C_0}, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = (0, 0, 1). \quad (9)$$

q_S это начальное состояние целевых ячеек. Ячейки в этом состоянии ждут сигнала из эфира, чтобы перейти в свою левую или правую версию.

$$\varphi(q_S, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = \begin{cases} q_{L*}, & \text{если } z_{\nu_1}^3 = 1 \\ q_{R*}, & \text{если } z_{\nu-1}^3 = 1 \\ q_S & \text{в остальных случаях} \end{cases}, \quad (10)$$

$$\psi(q_S, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = \begin{cases} (0, 0, 1) & \text{если } z_{\nu_1}^3 = 1 \\ (0, 1, 0) & \text{в остальных случаях} \end{cases}. \quad (11)$$

* это начальное состояние внутренних (не центральной и не целевых) ячеек. Ячейки в этом состоянии ждут сигнала из эфира, чтобы перейти в свою левую или правую версию. По совместительству это состояние покоя.

$$\varphi(*, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = \begin{cases} q_*^L, & \text{если } z_{\nu_1}^3 = 1, z_{\nu-1}^2 = 1 \\ q_*^R, & \text{если } z_{\nu_1}^2 = 1, z_{\nu-1}^3 = 1 \\ * & \text{в остальных случаях} \end{cases}, \quad (12)$$

$$\psi(*, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = (0, 0, 0). \quad (13)$$

Состояния q_*^L и q_*^R обеспечивают две функции. Во первых, они нужны для того, чтобы заполнить эфир необходимыми сигналами перед тем, как соответствующие ячейки перейдут в "рабочие" состояния q_1^L и q_1^R . Во вторых, внутренние ячейки в этом состоянии отслеживают специальный сигнал из эфира, исходя из которого можно понять, находятся ли они между центральной и ближайшей со своей стороны целевой ячейкой, или нет. Соответственно, если они не находятся в такой позиции, они затираются.

$$\varphi(q_*^L, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = \begin{cases} * & \text{если } z_{\nu_1}^3 = 1 \\ q_1^L & \text{в остальных случаях} \end{cases}, \quad (14)$$

$$\psi(q_*^L, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = \begin{cases} (0, 0, 0) & \text{если } z_{\nu_1}^3 = 1 \\ (0, 1, 1) & \text{в остальных случаях} \end{cases}, \quad (15)$$

$$\varphi(q_*^R, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = \begin{cases} * & \text{если } z_{\nu-1}^2 = 1 \\ q_1^R & \text{в остальных случаях} \end{cases}, \quad (16)$$

$$\psi(q_*^R, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = \begin{cases} (0, 0, 0) & \text{если } z_{\nu-1}^2 = 1 \\ (0, 1, 1) & \text{в остальных случаях} \end{cases}. \quad (17)$$

Состояния q_{L*} и q_{R*} — это специальные состояния для целевых ячеек. Ячейка в этом состоянии, получив специальный сигнал из эфира,

может понять, является она ближайшей к центру со своей стороны целевой ячейкой, или нет. В случае отрицательного ответа, соответствующая ячейка затирается.

$$\varphi(q_{L*}, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = \begin{cases} * & \text{если } z_{\nu_1}^3 = 1 \\ q_L & \text{в остальных случаях} \end{cases}, \quad (18)$$

$$\psi(q_*^L, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = (0, 0, 0), \quad (19)$$

$$\varphi(q_{R*}, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = \begin{cases} * & \text{если } z_{\nu-1}^2 = 1 \\ q_R & \text{в остальных случаях} \end{cases}, \quad (20)$$

$$\psi(q_*^R, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = (0, 0, 0). \quad (21)$$

Тройка $q_{CC}^{\bar{}}$, q_{CC}^{\gt} , q_{CC}^{\lt} — это рабочие состояния центральной ячейки. Центральная ячейка в этих состояниях побитово сравнивает длины отрезков. Алгоритм устроен так, что побитовые записи длин подаются на центральную ячейку от младшего разряда к старшему. Центральная ячейка анализируя соответствующий бит принимает текущий статус: q_{CC}^{\gt} — если левый бит текущего разряда больше правого, q_{CC}^{\lt} — если он меньше. При равенстве сигналов статус наследуется с предыдущего такта. Состояние $q_{CC}^{\bar{}}$ имеет место только в начальной итерации, или при совпадении всех битов, поданных в центральную ячейку.

$$\varphi(q_{CC}^X, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = \begin{cases} q_{CC}^X, & \text{если } z_{\nu-1}^2 = 1, z_{\nu_1}^2 = 1, z_{\nu-1}^1 = z_{\nu_1}^1 \\ q_{CC}^{\gt}, & \text{если } z_{\nu-1}^2 = 1, z_{\nu_1}^2 = 1, z_{\nu-1}^1 > z_{\nu_1}^1 \\ q_{CC}^{\lt}, & \text{если } z_{\nu-1}^2 = 1, z_{\nu_1}^2 = 1, z_{\nu-1}^1 < z_{\nu_1}^1 \\ q_{CF} & \text{в остальных случаях} \end{cases}, \quad (22)$$

$$\psi(q_{C1}, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = \begin{cases} (0, 0, 0), & \text{если } z_{\nu-1}^2 = 1, z_{\nu_1}^2 = 1 \\ (0, 0, 1), & \text{если } (z_{\nu-1}^2 = 0 \vee z_{\nu_1}^2 = 0) \wedge \\ & \wedge ((z_{\nu-1}^2 > z_{\nu_1}^2) \vee (z_{\nu-1}^2 = z_{\nu_1}^2 \wedge q_{CC}^X \neq q_{CC}^{\gt})) \\ (0, 0, 2) & \text{в остальных случаях} \end{cases}. \quad (23)$$

q_L и q_R — состояния соответственно левых и правых крайних ячеек. Они просто помечают окончание отрезка и могут затираться, или переходить в свое финальное состояние q_{XE} при получении соответствующего сигнала от центральной ячейки.

$$\varphi(q_L, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = \begin{cases} q_{LE}, & \text{если } z_{\nu_1}^3 = 1 \\ *, & \text{если } z_{\nu_1}^3 = 2 \\ q_L & \text{в остальных случаях} \end{cases}, \quad (24)$$

$$\psi(q_L, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = (0, 0, 0), \quad (25)$$

$$\varphi(q_R, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = \begin{cases} q_{RE}, & \text{если } z_{\nu-1}^3 = 2 \\ *, & \text{если } z_{\nu-1}^3 = 1 \\ q_R & \text{в остальных случаях} \end{cases}, \quad (26)$$

$$\psi(q_R, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = (0, 0, 0). \quad (27)$$

Состояние $q_{C_{W_2}}$ отслеживает, есть ли хотя бы одна ячейка в состоянии q_S с каждой из сторон. Если с одной из сторон такой ячейки не нашлось (то есть не получен соответствующий сигнал), то центральная ячейка переходит в состояние $q_{C_{L_1}}$ или $q_{C_{R_1}}$, чтобы досрочно завершить работу автомата ввиду отсутствия необходимости в сравнении расстояний до левого и правого соседей.

$$\varphi(q_{C_{W_2}}, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = \begin{cases} q_{C_{L_1}}, & \text{если } z_{\nu_1}^2 = 0 \\ q_{C_{R_1}}, & \text{если } z_{\nu-1}^2 = 0 \\ q_{C_{W_1}}, & \text{в остальных случаях} \end{cases}, \quad (28)$$

$$\psi(q_{C_{W_2}}, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = (0, 0, 0). \quad (29)$$

Состояние $q_{C_{W_1}}$ это однотактовый сон. Центральная ячейка в этом состоянии ничего не посылает в эфир и никак не реагирует на сигналы извне.

$$\varphi(q_{C_{W_1}}, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = q_{C_1}, \quad (30)$$

$$\psi(q_{CW_1}, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = (0, 0, 0). \quad (31)$$

Состояния q_{CL_1} или q_{CR_1} посылают в эфир сигнал, означающий досрочное завершение работы автомата.

$$\varphi(q_{CL_1}, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = q_{CF}, \quad (32)$$

$$\psi(q_{CL_1}, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = (0, 0, 1), \quad (33)$$

$$\varphi(q_{CR_1}, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = q_{CF}, \quad (34)$$

$$\psi(q_{CR_1}, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = (0, 0, 2). \quad (35)$$

Состояния, функции выходов и переходов которых перечислены ниже — это состояния при окончании работы автомата. Они не переходят в новые состояния и не посылают ничего в эфир.

$$\varphi(q_{LF}, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = q_{LF}, \quad (36)$$

$$\psi(q_{LF}, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = (0, 0, 0), \quad (37)$$

$$\varphi(q_{RF}, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = q_{RF}, \quad (38)$$

$$\psi(q_{RF}, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = (0, 0, 0), \quad (39)$$

$$\varphi(q_{LE}, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = q_{LE}, \quad (40)$$

$$\psi(q_{LE}, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = (0, 0, 0), \quad (41)$$

$$\varphi(q_{RE}, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = q_{RE}, \quad (42)$$

$$\psi(q_{RE}, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = (0, 0, 0), \quad (43)$$

$$\varphi(q_{CF}, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = q_{CF}, \quad (44)$$

$$\psi(q_{CF}, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = (0, 0, 0). \quad (45)$$

Ячейки, находящиеся в состоянии с символом L в записи будем называть левыми, с символом R — правыми. Пусть $Q_1 = \{q_k^X\}$, где $X \in \{R, L\}, k = 1$, $Q_0 = \{q_k^X\}$, где $X \in \{R, L\}, k = 0$.

4. Поведение автомата.

Опишем поведение автомата, разбив его на несколько этапов. Мы будем сопровождать описание этапов их реализацией на простом примере с начальным состоянием, приведенным ниже:

	t=0															
Q	q_S	*	q_S	*	*	*	*	*	*	q_{C_0}	*	*	*	*	q_S	
L^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L^2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R^2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ^2	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
ψ^3	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

$Q = q_1, q_2, \dots, q_n$ — строка состояний клеточного автомата. Здесь учитывается только та часть ячеек, справа и слева от которых были ячейки, которые не находились в состоянии * изначально, а так же все ячейки, которые изначально находились в состояниях q_S и q_{C_0}

$L^1 = l_1^1, l_2^1, \dots, l_n^1$ — строка значений первой компоненты эфира, соответствующих телесному углу ν_{-1} , в каждой ячейке

$L^2 = l_1^2, l_2^2, \dots, l_n^2$ — строка значений второй компоненты эфира, соответствующих телесному углу ν_{-1} , в каждой ячейке

$L^3 = l_1^3, l_2^3, \dots, l_n^3$ — строка значений третьей компоненты эфира, соответствующих телесному углу ν_{-1} , в каждой ячейке

$R^1 = r_1^1, r_2^1, \dots, r_n^1$ — строка значений первой компоненты эфира, соответствующих телесному углу ν_1 , в каждой ячейке

$R^2 = r_1^2, r_2^2, \dots, r_n^2$ — строка значений второй компоненты эфира, соответствующих телесному углу ν_1 , в каждой ячейке

$R^3 = r_1^3, r_2^3, \dots, r_n^3$ — строка значений третьей компоненты эфира, соответствующих телесному углу ν_1 , в каждой ячейке

$\psi^1 = \psi_1^1, \psi_2^1, \dots, \psi_n^1$ — первая компонента посылаемых каждой ячейкой сигналов в эфир

$\psi^2 = \psi_1^2, \psi_2^2, \dots, \psi_n^2$ — вторая компонента посылаемых каждой ячейкой сигналов в эфир

$\psi^3 = \psi_1^3, \psi_2^3, \dots, \psi_n^3$ — третья компонента посылаемых каждой ячейкой сигналов в эфир

4.1. Этап 1: определение ориентации.

Пусть начальное состояние клеточного автомата удовлетворяет условиям Теоремы 1:

Q	t=0															
	q_S	*	q_S	*	*	*	*	*	*	q_{C_0}	*	*	*	*	q_S	
L^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L^2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R^2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ^2	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
ψ^3	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

На первом такте в эфире нет нетривиальных сигналов. Функции переходов состояний q_S и $*$, при тривиальном сигнале из эфира, принимают значения q_S и $*$ соответственно. Функция выходов этих состояний тождественно равны $*$ соответственно $(0, 1, 0)$ и $(0, 0, 0)$. Функция переходов состояния q_{C_0} тождественно равна $q_{C_{W_2}}$, а функция выходов — $(0, 0, 1)$. Таким образом, на следующем такте состояние поменяет только центральная ячейка. Помимо этого в эфире появятся сигналы от центральной и крайних ячеек:

Q	t=1															
	q_S	*	q_S	*	*	*	*	*	*	$q_{C_{W_2}}$	*	*	*	*	q_S	
L^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L^2	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
L^3	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
R^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R^2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
R^3	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
ψ^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ^2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
ψ^3	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Значения функции переходов состояний q_S и $*$ зависят от того, с какой стороны пришли сигналы $(0, 1, 0)$ и $(0, 0, 1)$. В зависимости от этого ячейки в состоянии q_S могут перейти в состояния q_{L^*} , подав в эфир сигнал $(0, 0, 1)$, или q_{R^*} , подав сигнал $(0, 1, 0)$, а ячейки в состоянии $*$ — в q_*^L или q_*^R (при этом подав в эфир сигнал $(0, 0, 0)$). Состояние $q_{C_{W_2}}$ отслеживает, есть ли хотя бы одна ячейка в состоянии q_S (ищет сигнал $(0, 1, 0)$) с каждой из сторон. Если с одной из сторон такой ячейки не

нашлось (то есть не получен соответствующий сигнал), то центральная ячейка переходит в состояние $q_{C_{L_1}}$ или $q_{C_{R_1}}$, чтобы досрочно завершить работу автомата ввиду отсутствия необходимости в сравнении расстояний до левого и правого соседей. В нашем случае с каждой стороны оказались ячейки в состоянии q_S , поэтому состояние $q_{C_{W_2}}$ по сути представляет собой состояние двухтактного сна:

Q	t=2														
	q_{L_*}	q_*^L	q_{L_*}	q_*^L	q_*^L	q_*^L	q_*^L	q_*^L	q_*^L	$q_{C_{W_1}}$	q_*^R	q_*^R	q_*^R	q_*^R	q_{R_*}
L^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L^2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L^3	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R^2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
R^3	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ^1	0	0	0	1	1	1	1	1	1	0	1	1	1	1	0
ψ^2	0	0	0	1	1	1	1	1	1	0	1	1	1	1	0
ψ^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

На этом этап определения ориентации заканчивается. Если сравнить начальное и конечное состояния автомата на этом этапе, можно сказать, что из всех ячеек, изначально находившихся в состоянии q_S , вне состояния $*$ остались только две ближайшие к центральной. Эти ячейки из состояния q_S перейдут в состояние q_L , если они слева от центра, и в q_R — если справа; ячейки в состоянии $*$ перейдут в состояние q_*^L если они слева от центра и справа от ячейки в состоянии q_L , и в q_*^R — в противоположном случае; центральная ячейка перейдет в состояние $q_{C_{W_1}}$.

4.2. Этап 2: Сравнение длин.

Это сложный, итеративный этап.

Ячейки в состояниях q_L и q_R будем называть крайними, а ячейки, находящиеся между крайней ячейкой и центральной — внутренними. Определим, что этот этап заканчивается в момент, когда центральная ячейка посылает сигналы $\psi \in \{(0, 0, 1); (0, 0, 2)\}$. Из функций выходов и переходов видно, что центральная ячейка на этом этапе всегда находится в состояниях $q_{CC}^{\bar{}}$, $q_{CC}^{\bar{}}$, $q_{CC}^{\bar{}}$, поскольку она находится в состоянии $q_{CC}^{\bar{}}$ изначально, и условия её перехода в состояние вне этой тройки совпадают с условиями подачи этой ячейкой в эфир указанного набора сигналов. Наконец, легко заметить, что дизъюнкция этих условий по сути заключается в том, что все внутренние ячейки хотя бы одной из сторон перешли в состояние $q \in Q_0$.

Рассмотрим функционирование левой части клеточного автомата. Пронумеруем внутренние ячейки автомата слева направо, и будем обозначать за $q^{i,t}$, $i \in \{1, 2, \dots, s_L\}$, $t \geq 0$ состояние i -й ячейки в момент времени t . Будем считать, что $t = 0$ — момент начала этапа 2.1.

$q^{i,t} \in Q_1$ тогда и только тогда, когда $i(\bmod 2^t) = 0$.

Доказательство. Докажем по индукции:

В начальный момент времени $t = 0$ все ячейки находятся в состоянии $q_1^L \in Q_1$, что соответствует тривиальному условию $i(\bmod 2^0) = 0$.

Докажем что лемма выполняется в момент времени $t = 1$. Из описания функции перехода состояний $\{q_1^L; q_1^R; q_0^L; q_0^R\}$ следует, что $q^{i,1} \in Q_1 \leftrightarrow q^{i,0} \in Q_1 \wedge z_{\nu-1}^{i,0} = (1, \alpha, \beta) \leftrightarrow z_{\nu-1}^{i,0} = (1, \alpha, \beta)$ (здесь и далее α и β — произвольные элементы множеств $\{0;1\}$ и $\{0;1;2\}$) соответственно.

Мы знаем, что $z_{\nu-1}^{i,0} = \sum_{j=1}^i \psi(q_*^L, q_{-1}, q_1, z_{\nu-1}, z_{\nu_1}) = \sum_{j=1}^{i-1} (1, 1, 0) = (i(\bmod 2), 1, 0)$, следовательно $q^{i,1} \in Q_1 \leftrightarrow i(\bmod 2) = 0$.

Пусть лемма выполняется для всех $t = 0, 1, 2, \dots, k$, докажем что она выполняется в момент времени $k + 1$. Из описания функции перехода состояний $\{q_1^L; q_1^R; q_0^L; q_0^R\}$ следует, что $q^{i,k+1} \in Q_1 \leftrightarrow q^{i,k} \in Q_1 \wedge z_{\nu-1}^{i,k} = (1, \alpha, \beta)$. По построению функций выходов и переходов состояний $\{q_1^L; q_1^R; q_0^L; q_0^R\}$, $\psi^{i,k-1} = (1, \alpha, \beta) \leftrightarrow q^{i,k} \in Q_1$, следовательно $z_{\nu-1}^{i,k} = \left(\sum_{j(\bmod 2^k)=0, j < i, j > 0} 1, \alpha, \beta \right)$. Таким образом, по предположению индукции:

$$\begin{aligned} q^{i,k+1} \in Q_1 &\leftrightarrow q^{i,k} \in Q_1 \wedge z_{\nu-1}^{i,k} = (1, \alpha, \beta) \leftrightarrow i(\bmod 2^k) = 0 \wedge \\ \sum_{j(\bmod 2^k)=0, j < i, j > 0} 1 &= 1 \leftrightarrow i(\bmod 2^k) = 0 \wedge (i(\bmod 2^{k+1}) > 2^k \vee i(\bmod 2^{k+1}) = 0) \leftrightarrow i(\bmod 2^{k+1}) = 0 \end{aligned} \quad \square$$

Поскольку на этом этапе функции переходов и выходов соответствующих ячеек левой части существенно зависят только от сигналов эфира с левой стороны, а правой части — с правой, левая и правая часть в рамках этого этапа функционируют независимо, и для правой части можно привести аналогичные рассуждения.

Из леммы следует, что в момент времени t в качестве первой компоненты эфира на центральную ячейку приходит величина

$$\sum_{i(\bmod 2^t)=0, i \in [1; s]} 1 = \lfloor \frac{s}{2^t} \rfloor (\bmod 2),$$

что по сути является $(t + 1)$ -м битом двоичной записи числа s . Таким образом, центральная ячейка может побитово сравнивать числа s_L и s_R , читая их двоичную запись справа

налево. Функции выходов и переходов состояний центральной ячейки q_{CC}^X , $X \in \{=, <, >\}$ повторяют алгоритм побитового сравнения чисел: если i -й бит слева больше соответствующего бита справа, то центральная ячейка переходит в состояние $q_{CC}^>$, если меньше — в состояние $q_{CC}^<$, если они равны, то состояние не меняется. Так происходит до тех пор, пока трансляция одного из чисел не закончится ($z_{\nu-1}^2 = 0 \vee z_{\nu_1}^2 = 0$). В этот момент центральная ячейка, в зависимости от своего текущего состояния, отправляет сигнал $(0, 0, 2)$ (левая сторона длиннее) или $(0, 0, 1)$ (правая сторона длиннее).

Реализацию этапа 2 на нашем примере смотрите ниже:

t=3																	
Q	*	*	q_L	q_1^L	$q_{CC}^=$	q_1^R	q_1^R	q_1^R	q_1^R	q_R							
L^1	0	0	0	0	1	0	1	0	1	0	0	0	1	0	1	0	1
L^2	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
L^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R^1	1	1	1	0	1	0	1	0	1	1	0	1	0	1	0	0	0
R^2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
R^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ^1	0	0	0	0	1	0	1	0	1	0	0	1	0	1	0	0	0
ψ^2	0	0	0	0	1	0	1	0	1	0	0	1	0	1	0	0	0
ψ^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

t=4																		
Q	*	*	q_L	q_0^L	q_1^L	q_0^L	q_1^L	q_0^L	q_1^L	q_0^L	q_1^L	$q_{CC}^<$	q_0^R	q_1^R	q_0^R	q_1^R	q_0^R	q_R
L^1	0	0	0	0	0	1	1	0	0	1	1	1	1	1	0	0	1	1
L^2	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
L^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R^1	1	1	1	1	0	0	1	1	0	0	0	0	1	1	0	0	0	0
R^2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
R^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ^1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
ψ^2	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
ψ^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

t=5																		
Q	*	*	q_L	q_0^L	q_0^L	q_0^L	q_1^L	q_0^L	q_0^L	q_0^L	$q_{CC}^>$	q_0^R	q_1^R	q_0^R	q_0^R	q_0^R	q_0^R	q_R
L^1	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0
L^2	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
L^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R^1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0
R^2	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
R^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ^2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Q	t=6																
	*	*	q_L	q_0^L	$q_{CC}^>$	q_0^R	q_0^R	q_0^R	q_0^R	q_0^R	q_R						
L^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L^2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R^2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ^2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ^3	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0

4.3. Этап 3: завершение работы.

Линейный этап. Начинается после того, как в процессе этапа 2 центральная ячейка отправляет в эфир один из сигналов $\{(0, 0, 1), (0, 0, 2)\}$. Пусть левая сторона оказалась длиннее и был отправлен сигнал $(0, 0, 2)$:

Q	t=6																
	*	*	q_L	q_0^L	$q_{CC}^>$	q_0^R	q_0^R	q_0^R	q_0^R	q_0^R	q_R						
L^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L^2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R^2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ^2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ^3	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0

В следующий такт этот сигнал появляется в эфире:

Q	t=7																
	*	*	q_L	q_0^L	q_0^L	q_0^L	q_0^L	q_0^L	q_0^L	q_{CF}	q_0^R	q_0^R	q_0^R	q_0^R	q_0^R	q_R	
L^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L^2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L^3	0	0	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2
R^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R^2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R^3	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0
ψ^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ^2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Левые ячейки, получая такой сигнал на вход, всегда переходят в состояние покоя *, крайняя правая — в состояние q_{RE} , а остальные правые ячейки — в состояние q_{RF} :

Q	t=8										q _{CF}	q _{RF}	q _{RE}				
	*	*	*	*	*	*	*	*	*	*							
L ¹	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
L ²	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
L ³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R ¹	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R ²	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R ³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ψ ¹	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ψ ²	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ψ ³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

5. Иллюстрация поведения автомата

Для наглядности ещё раз приведем пример функционирования автомата, но на этот раз целиком:

Q	t=0															
	q _S	*	q _S	*	*	*	*	*	*	*	q _{C₀}	*	*	*	*	q _S
L ¹	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L ²	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L ³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R ¹	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R ²	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R ³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ ¹	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ ²	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
ψ ³	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Q	t=1															
	q _S	*	q _S	*	*	*	*	*	*	q _{C_{W₂}}	*	*	*	*	q _S	
L ¹	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L ²	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
L ³	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
R ¹	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R ²	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
R ³	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
ψ ¹	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ ²	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
ψ ³	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Q	t=2																
	q _{L*}	q _* ^L	q _{L*}	q _* ^L	q _{C_{W₁}}	q _* ^R	q _{R*}										
L ¹	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
L ²	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
L ³	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
R ¹	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R ²	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	
R ³	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ψ ¹	0	0	0	1	1	1	1	1	1	0	1	1	1	1	1	0	
ψ ²	0	0	0	1	1	1	1	1	1	0	1	1	1	1	1	0	
ψ ³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

		t=3															
Q	*	*	q_L	q_1^L	q_{CC}^-	q_1^R	q_1^R	q_1^R	q_1^R	q_1^R	q_R						
L^1	0	0	0	0	1	0	1	0	1	0	0	0	1	0	1	0	1
L^2	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
L^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R^1	1	1	1	0	1	0	1	0	1	1	0	1	0	1	0	0	0
R^2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
R^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ^1	0	0	0	0	1	0	1	0	1	0	0	1	0	1	0	0	0
ψ^2	0	0	0	0	1	0	1	0	1	0	0	1	0	1	0	0	0
ψ^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

		t=4																	
Q	*	*	q_L	q_0^L	q_1^L	q_0^L	q_1^L	q_0^L	q_1^L	q_0^L	q_1^L	$q_{CC}^<$	q_0^R	q_1^R	q_0^R	q_1^R	q_0^R	q_1^R	q_R
L^1	0	0	0	0	0	1	1	0	0	1	1	1	1	1	0	1	1	1	
L^2	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
L^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R^1	1	1	1	1	0	0	1	1	0	0	0	0	1	1	0	0	0	0	
R^2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	
R^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ψ^1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	
ψ^2	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	
ψ^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

		t=5																
Q	*	*	q_L	q_0^L	q_0^L	q_0^L	q_1^L	q_0^L	q_0^L	q_0^L	$q_{CC}^>$	q_0^R	q_1^R	q_0^R	q_0^R	q_0^R	q_0^R	q_R
L^1	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	
L^2	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	
L^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R^1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	
R^2	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	
R^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ψ^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ψ^2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ψ^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

		t=6															
Q	*	*	q_L	q_0^L	q_0^L	q_0^L	q_0^L	q_0^L	q_0^L	$q_{CC}^>$	q_0^R	q_0^R	q_0^R	q_0^R	q_0^R	q_0^R	q_R
L^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L^2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R^2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ^2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ^3	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0

		t=7														
Q	*	*	q_L	q_0^L	q_0^L	q_0^L	q_0^L	q_0^L	q_0^L	q_{CF}	q_0^R	q_0^R	q_0^R	q_0^R	q_0^R	q_R
L^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L^2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L^3	0	0	0	0	0	0	0	0	0	0	2	2	2	2	2	2
R^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R^2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R^3	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0
ψ^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ^2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

		t=8														
Q	*	*	*	*	*	*	*	*	*	q_{CF}	q_{RF}	q_{RF}	q_{RF}	q_{RF}	q_{RF}	q_{RE}
L^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L^2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R^2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ^1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ^2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ψ^3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6. Время работы автомата

Посчитаем время T , затрачиваемое автоматом σ на поиск кратчайшего отрезка. Из описания поведения автомата видно:

$$T = T_1 + T_2 + T_3. \quad (46)$$

где T_k — длительности соответствующих этапов. Легко заметить, что $T_1 = 3$, $T_3 = 2$. Из Леммы 1 следует, что этап 2 заканчивается в момент $t = \min_{2^r > s} r = \lceil \log_2(s + 0.5) \rceil$, где $s = \min(s_L, s_R)$. Поскольку в Лемме 1 отсчет времени велся от нуля, количество тактов T_2 , затраченных на этап 2 будет оцениваться следующим образом:

$$T_2 = \lceil \log_2(s + 0.5) \rceil + 1 \leq \log_2(s) + 2.$$

Следовательно,

$$T \leq \log_2(s) + 7.$$

Список литературы

- [1] Дж. фон Нейман, *Теория самовоспроизводящихся автоматов*, Мир, Москва, 1971.
- [2] Neumann J., von, *Collected works*, New York, 1961 – 1963.
- [3] Neumann J., von, *Theory of self-reproducing automata*, London, 1966.
- [4] Burks A., *Essays on Cellular Automata*, University of Illinois Press, 1971.
- [5] Мур Э. Ф., “Математические модели самовоспроизведения”, *В кн.: Математические проблемы в биологии*, 1966.
- [6] Кудрявцев В. Б., Подколзин А. С., Болотов А. А., *Основы теории однородных структур*, Наука, Москва, 1990.
- [7] Кудрявцев В. Б., Гасанов Э. Э., Подколзин А. С., *Теория интеллектуальных систем: в 4 кн. Книга четвертая. Теория автоматов.*, Издательские решения, Москва, 2018.
- [8] Титова Е.Е., “Конструирование движущихся изображений клеточными автоматами”, *Интеллектуальные системы*, **18:1** (2014), 153–180.
- [9] Калачев Г.В., Титова Е.Е., “О мере множества законов движения точки, реализуемых клеточными автоматами”, *Интеллектуальные системы. Теория и приложения*, **22:3** (2018), 105–125.
- [10] Гасанов Э.Э., “Клеточные автоматы с локаторами”, *Интеллектуальные системы. Теория и приложения*, **20:2** (2020), 121–133.

The closest neighbour problem solution using the cellular automata with locators model

Vasilev D.I.

The paper considers applying the locator cellular automaton model to the closest neighbour search problem. The locator cellular automaton model assumes the possibility for each cell to translate a signal through any distance using ether. It is proven in this paper that such possibility allows to decrease the problem complexity from linear to logarithmic (against the classic cellular automaton model).

Keywords: cellular automata, homogeneous structures, the closest neighbour search problem.

Решетка 1-следов замкнутых классов кусочно-линейных функций

Кан А.Н.¹

В настоящей работе найдена решетка 1-следов замкнутых классов кусочно-линейных функций. А. Н. Кан

Ключевые слова: кусочно-линейные функции, непрерывные кусочно-линейные функции, финитно-параллельные функции, непрерывные финитно-параллельные функции, непрерывные финитно-линейные функции, кусочно-параллельные функции, финитно-линейные функции, параллельные финитно-линейные функции, решетка 1-следов, функция Хэвисайда, 2-предполный класс.

1. Введение

В настоящей работе найдена решетка 1-следов замкнутых классов кусочно-линейных функций. Ранее, в работе [1] были рассмотрены кусочно-параллельные функции и был найден единственный предполный класс S -финитно-линейных функций. Класс кусочно-параллельных функций является подклассом кусочно-линейных функций, которые и рассматриваются в данной работе. Данный класс связан с нейронными сетями [6]. Для класса кусочно-линейных функций были найдены все предполные классы [3]. Одним из таких классов стал класс согласованных функций, в котором был найден 2-предполный класс и доказано его единственность [5]. Также был исследован класс непрерывных кусочно-линейных функций, для которого также был найден единственный 2-предполный класс [4].

¹Кан Александр Николаевич — аспирант каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: kan.alexander.n@gmail.com.

Kan Alexandr Nikolaevich — graduate student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

2. Основные понятия и определения

Рассмотрим следующие функции действительных аргументов:

1) Функция $\Theta(x)$ (Хэвисайда):

$$\Theta(x) = \begin{cases} 0, & \text{если } x < 0 \\ 1, & \text{иначе} \end{cases} \quad (1)$$

2) Функция $\Theta'(x)$:

$$\Theta'(x) = 1 - \Theta(-x) = \begin{cases} 0, & \text{если } x \leq 0 \\ 1, & \text{иначе} \end{cases} \quad (2)$$

3) Функция $F(x, y)$:

$$F(x, y) = \begin{cases} 0, & \text{если } y \leq 0 \\ x, & \text{иначе} \end{cases} \quad (3)$$

Введем некоторые обозначения:

1) Запись вида $f(\bar{a} \cdot t + \bar{b})$ равняется записи $f(a_1 \cdot t + b_1, \dots, a_n \cdot t + b_n)$, где $\bar{a} = (a_1, \dots, a_n)$ и $\bar{b} = (b_1, \dots, b_n)$.

2) Под операцией " \cdot ", примененной к векторам, подразумевается операция скалярного произведения векторов.

3) Пусть $M \subset PL$, тогда $M^{(1)}$ это множество состоящие, из всех одноместных функций множества M .

4) Пусть $M \subset PL$, тогда $M^{(2)}$ это множество состоящие, из всех двухместных функций множества M .

5) В данной статье рассматривается замыкание по операциям суперпозиции. Замыкание множества $M \subset PL$ по операциям суперпозиции, будем обозначать через $[M]$. [1]

Введем следующие определения:

Определение 1. Функция $f : \mathbb{R}^n \Rightarrow \mathbb{R}$ называется линейной, если найдутся $\bar{a} \in \mathbb{R}^n$ и $c \in \mathbb{R}$, такие что $f(\bar{x}) = \bar{x} \cdot \bar{a} + c$. Множество всех линейных функций обозначим через L .

Пусть l_i - гиперплоскость, задаваемая уравнением $\bar{x} \cdot \bar{a}_i + c_i = 0$, $a_i \in \mathbb{R}^n \setminus \{\bar{0}\}$, $c_i \in \mathbb{R}$, $i = 1, \dots, k$. Для каждой точки $\bar{x} \in \mathbb{R}^n$ рассмотрим вектор

$\sigma(\bar{x}) = (\sigma_1, \dots, \sigma_k)$ с компонентами из множества $\{-1, 0, 1\}$, $\sigma_i = \text{sgn}(\bar{x} \cdot \bar{a}_i + c_i)$, где

$$\text{sgn}(b) = \begin{cases} -1, & \text{если } b < 0 \\ 0, & \text{если } b = 0 \\ 1, & \text{если } b > 0. \end{cases} \quad (4)$$

Определение 2. Две точки $\bar{x}, \bar{y} \in \mathbb{R}^n$ эквивалентны относительно гиперплоскостей l_1, \dots, l_k тогда и только тогда, когда $\sigma(\bar{x}) = \sigma(\bar{y})$, обозначим это через $\bar{x} \sim \bar{y}$.

Легко проверить, что отношение " \sim " является отношением эквивалентности. Таким образом, пространство \mathbb{R}^n разбивается на классы эквивалентности R_1, \dots, R_s .

Определение 3. Сигнатурой класса R называется вектор $\sigma(R) = \sigma(\bar{x})$, где \bar{x} точка класса R .

Пусть R_1, \dots, R_s - все классы эквивалентности на которые гиперплоскости l_1, \dots, l_k разбивают \mathbb{R}^n .

Определение 4. Функция $f : \mathbb{R}^n \Rightarrow \mathbb{R}$ называется кусочно-линейной, если $\forall j \in \{1, \dots, s\}$ найдутся $\bar{b}_j \in \mathbb{R}^n$ и $d_j \in \mathbb{R}$, что для всех $\bar{x} \in R_j$ выполняется $f(\bar{x}) = \bar{x} \cdot \bar{b}_j + d_j$. Линейную функцию $\bar{x} \cdot \bar{b}_j + d_j$, реализуемую на множестве R_j , обозначим $f_{R_j}(\bar{x})$. Множество всех кусочно-линейных функций обозначим через PL .

Определение 5. Функция $f : \mathbb{R} \Rightarrow \mathbb{R}$ называется одноместной непрерывной финитно-линейной, если $f \in CPL^{(1)}$ и $\exists a, b, N \in \mathbb{R}$, что $f(x) = a \cdot x + b, \forall |x| > N$. Класс всех таких функций обозначим через $CFS^{(1)}$.

Определение 6. Функция $f : \mathbb{R} \Rightarrow \mathbb{R}$ называется одноместной непрерывной финитно-параллельной, если $f \in PL^{(1)}$ и $\exists N, a, b_1, b_2 \in \mathbb{R}$ такие, что $\forall x < -N, f(x) = a \cdot x + b_1$ и $\forall x > N, f(x) = a \cdot x + b_2$. Класс всех таких функций обозначим через $CFP^{(1)}$.

Определение 7. Функция $f : \mathbb{R} \Rightarrow \mathbb{R}$ называется линейной с выколотыми точками, если $f \in PL^{(1)}$ и $\exists a_i, b_i \in \mathbb{R}, i = 1, \dots, s$, что $f(x) = a \cdot x + b, \forall x \in \mathbb{R} / \{a_1, \dots, a_s\}$ и $f(a_i) = b_i, i = 1, \dots, s$. Класс всех таких функций обозначим через $L'^{(1)}$.

Определение 8. Функция $f : \mathbb{R} \Rightarrow \mathbb{R}$ называется *одноместной параллельной финитно-линейной*, если $f \in PP^{(1)}$ и $\exists N, a, b \in \mathbb{R}$, что $f(x) = a \cdot x + b$, при $|x| > N$. Класс всех таких функций обозначим через $PFS^{(1)}$.

Определение 9. Функция $f : \mathbb{R} \Rightarrow \mathbb{R}$ называется *одноместной кусочно-параллельной*, если $f \in PL^{(1)}$ и $\exists a_i, x_i^1, x_i^2 \in \mathbb{R} : i = 1, \dots, s$, что $f(x) = a_i + l(x)$, при $x \in (x_i^1, x_i^2)$, где $l(x) \in L^{(1)}$. Класс всех таких функций обозначим через $PP^{(1)}$.

Определение 10. Функция $f : \mathbb{R} \Rightarrow \mathbb{R}$ называется *одноместной финитно-линейной*, если $f \in PL^{(1)}$ и $\exists a, b, N \in \mathbb{R}$ такие, что $f(x) = a \cdot x + b, \forall |x| > N$. Класс всех таких функций обозначим через $FPS^{(1)}$.

Определение 11. Функция $f : \mathbb{R} \Rightarrow \mathbb{R}$ называется *одноместной финитно-параллельной*, если $f \in PL^{(1)}$ и $\exists a, b_1, b_2, N \in \mathbb{R}$ такие, что $f(x) = a \cdot x + b_1$, при $x < -N$, и $f(x) = a \cdot x + b_2$, при $x > N$, Класс всех таких функций обозначим через $FP^{(1)}$.

3. Подрешетка 1-следов замкнутых классов непрерывных кусочно-линейных функций

Пусть $M \subseteq CFS^{(1)}$ тогда верна следующая лемма.

Лемма 1. $CFS^{(1)} \subseteq [M \cup L] \iff M \not\subseteq L^{(1)}$.

Доказательство. Пусть $f_1 \in CFS^{(1)} \setminus L^{(1)}$.

$$f_1(x) = \begin{cases} a_1 \cdot x + b_1, & \text{если } x \leq c_1 \\ a_2 \cdot x + b_2, & \text{если } c_1 \leq x \leq c_2 \\ \cdot \\ \cdot \\ a_{n-1} \cdot x + b_{n-1}, & \text{если } c_{n-2} \leq x \leq c_{n-1} \\ a_1 \cdot x + b_1, & \text{если } x \geq c_{n-1} \end{cases} \quad (5)$$

Так как у нас имеются все линейные функции, то отняв от $f_1(x)$ функцию $a_1 \cdot x + b_1$, получим функцию $f_2(x)$.

$$f_2(x) = \begin{cases} 0, & \text{если } x \leq c_1 \\ a'_2 \cdot x + b'_2, & \text{если } c_1 \leq x \leq c_2 \\ \cdot \\ \cdot \\ a'_{n-1} \cdot x + b'_{n-1}, & \text{если } c_{n-2} \leq x \leq c_{n-1} \\ 0, & \text{если } x \geq c_{n-1} \end{cases} \quad (6)$$

Далее, так как у нас имеются все линейные функции, мы не будем подробно описывать такие линейные преобразования над функцией как отражение, сдвиг на константу, сжатие и растяжение.

Данная функция ограничена, а следовательно существует максимум. Пусть

$$\text{max}_{left} = \{x \mid f_2(x) = \text{max}(f_2) \text{ и } \forall x' < x, f_2(x') < \text{max}(f_2)\} \quad (7)$$

Построим функцию, которая будет иметь единственный максимум в точке ноль.

$$f_3(x) = f_2(x + \text{max}_{left}) + f_2(-x + \text{max}_{left}) \quad (8)$$

Отнимем константу от получившейся функции так, чтобы функция была всюду меньше нуля кроме небольшой окрестности в точке ноль.

$$f_4(x) = f_3(x) - \text{max}(f_3) + c \quad (9)$$

График функции f_4 схематично может быть изображен как показано на Рис. 1.

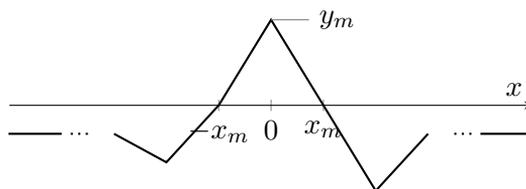


Рис. 1

Изменим функцию f_2 так, чтобы первый изгиб функции начинался в точке ноль и был направлен вверх.

$$f_2'(x) = f_2(x + c_1) \quad (10)$$

Функция f_2' выглядит как показано на Рис. 2.

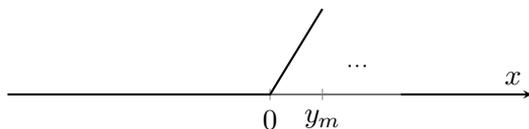


Рис. 2

Далее, подставим функцию f_4 в функцию f_2' . Получим функцию «симметричная горка». Ее график можно увидеть на Рис. 3.

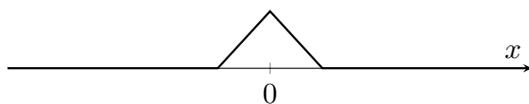


Рис. 3

Так как мы имеем все линейные функции то мы можем получить произвольную функцию вида «симметричная горка».

Из получившихся функций и линейных функций построим произвольную функцию вида «горка». То есть функцию которая может иметь различные углы при правом и левом изгибе. Пусть $g(x)$ произвольная функция вида «горка». Рассмотрим две функции $g_l(x)$ и $g_r(x)$, которые принадлежат семейству «симметричная горка» и имеют угол наклона изгиба как левый изгиб и правый изгиб $g(x)$ соответственно. Пусть $g_l(x)$ имеет горку, которая заканчивается в точке ноль, а $g_r(x)$ имеет горку, начинающуюся в точке ноль. Тогда сложив две эти функции мы получим функцию $g_1(x)$ (Рис. 4)

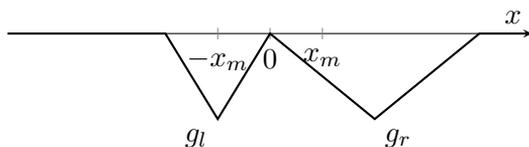


Рис. 4

Функция $g_1(x)$ имеет в себе искомую «горку». Рассмотрим функцию $p(x)$ которая имеет вид как показано на Рис.5.

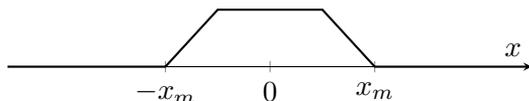


Рис. 5

Функцию $p(x)$ легко получить сложив две соответствующие функции из класса «симметричная горка». Далее сложим функцию $g_1(x)$ с функцией $p(x)$ и получим функцию $g_2(x)$, которая схематично можно изобразить как показано на Рис. 6.

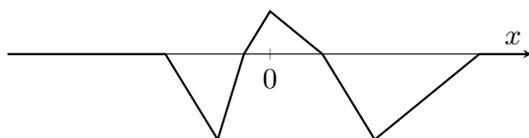


Рис. 6

Мы почти получили искомую функцию, осталось только избавиться от ненужных изгибов. Данную процедуру мы делали ранее. Просто подставим функцию $g_2(x)$ в функцию $d(x)$ из класса симметричная горка.

$$d(x) = \begin{cases} 0, & \text{если } x \leq 0 \\ x, & \text{если } 0 < x \leq \max(g_2(x)) \\ \cdot & \\ \cdot & \end{cases} \quad (11)$$

Функция $d(g_2(x))$ и будет искомой функцией.

Следовательно мы можем получить произвольную функцию из класса «горка». С помощью данных функций попытаемся построить произвольную функцию из класса $CFS^{(1)}$.

Пусть мы хотим получить функцию $g(x) \in CFS^{(1)}$, которая имеет n изгибов, $n > 3$. Если функция имеет 3 изгиба то это функция из класса «горка», а такие функции мы получать умеем. Пусть кол-во изгибов больше трех тогда покажем, что их можно уменьшить на один. Без

ограничения общности можно считать что функция $g(x)$ имеет вид как показано на рис.7.

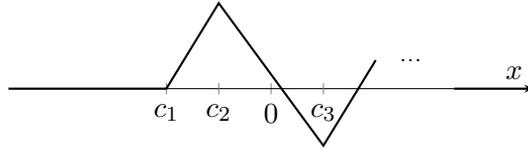


Рис. 7

Прибавим к ней функцию $g_{c1}(x)$ которая принадлежит классу «горка». Функция $g_{c1}(x)$ имеет вид как показано на Рис. 8.

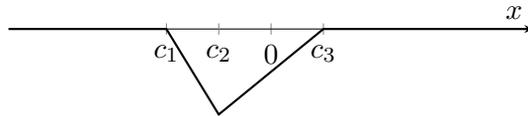


Рис. 8

То есть, функция $g_{c1}(x)$ имеет изгибы в тех же точках что и функция $g(x)$ имеет первые три изгиба. Линейная функция реализуемая на интервале (c_1, c_2) у функции $g_{c1}(x)$ равна соответствующей функции реализуемой на том же интервале у функции $-g(x)$. Сложив функции $g(x)$ и $g_{c1}(x)$ получим, что результирующая функция равна нулю на интервале (c_1, c_2) , а следовательно не имеет изгиба в точке c_1 . Прделав такую операцию несколько раз мы получим функцию вида «горка», которую мы получать умеем. То есть, мы доказали что с помощью функций из класса «горка» мы можем получить произвольную функцию из класса $CFS^{(1)}$. Лемма доказана. \square

Следствие 1. Класс $L^{(1)}$ является единственным максимальным собственным 1-следом в классе $CFS^{(1)}$.

Пусть $M \subseteq CFP^{(1)}$ тогда верна следующая лемма.

Лемма 2. $CFP^{(1)} \subseteq [M \cup L] \iff M \not\subseteq CFS^{(1)}$

Доказательство. Пусть $f(x) \in CFP^{(1)} \setminus CFS^{(1)}$. Без ограничения общности можно считать, что при $x < N$, где $N \in \mathbb{R}$, $f(x) = 0$. Это всегда можно сделать добавив к имеющейся функции некую линейную функцию. Сначала покажем что у нас имеются все функции из класса $CFS^{(1)}$.

Из данной функции легко получить некую функцию из класса $CFS^{(1)}$, которая при это не будет линейной. Например, можно сложить функцию $f(x)$ с функцией $-f(x + c)$, где $c \in \mathbb{R}$. Из предыдущей теоремы следует что из получившейся функции мы можем получить все функции из класса CFS . Далее докажем что $CFP \subseteq [\{f(x)\} \cup CFS]$. Пусть функция $f(x)$ имеет вид как показано на Рис. 9. Рассмотрим функцию $g(x) \in CFS^{(1)}$ (Рис. 10), которая почти совпадает с функцией $f(x)$.

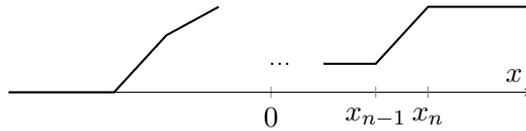


Рис. 9

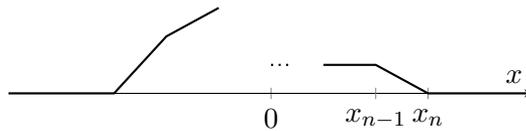


Рис. 10

Данные функции отличаются только на интервалах (x_{n-1}, x_n) и (x_n, ∞) . Если отнять от функции $f(x)$ функцию $g(x)$, то мы получим функцию $s(x) = f(x) - g(x)$, которая имеет вид как показано на Рис. 11.

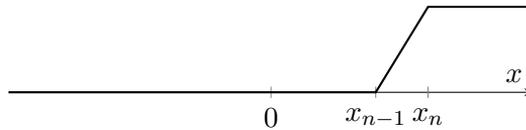


Рис. 11

С помощью линейных функций можно получить произвольную функцию такого вида. Пусть функция $f(x) \in CFP^{(1)}$ произвольная функция которую мы хотим получить. Тогда $f(x) = g(x) + s(x)$, где $g(x) \in CFS^{(1)}$ это почти совпадающая функция, а функция $s(x) = f(x) - g(x)$ которые мы получать умеем. Лемма доказана. \square

Следствие 2. Класс $CFS^{(1)}$ является единственным максимальным собственным 1-следом в классе $CFP^{(1)}$.

Задача 2-полноты класса CPL была рассмотрена в работе [4]. Было показано, что класс $CFP^{(2)}$ единственный 2-предполный класс в $CPL^{(2)}$. Из доказательства вытекает следующее утверждение.

Следствие 3. *класс $CFP^{(1)}$ является единственным максимальным собственным 1-следом в классе $CPL^{(1)}$.*

4. Подрешетка 1-следов замкнутых классов кусочно-линейных функций допускающие разрывы

Пусть $M \subseteq L^{(1)}$ тогда верна следующая лемма.

Лемма 3. $L^{(1)} \subseteq [M \cup L] \iff M \not\subseteq L^{(1)}$

Доказательство. Пусть $f(x) \in L^{(1)} \setminus L^{(1)}$. Без ограничения общности можно считать, что функция $f(x)$ всюду равна нулю за исключением n точек. Пусть функция $f(x)$ имеет единственную точку максимума x_{max} (аналогично теореме полноты в классе CFS), а в точке x_{smax} функция равняется второму по величине значению функции $f(x)$. Пусть функция $f(x)$ выглядит как показано на Рис. 12.

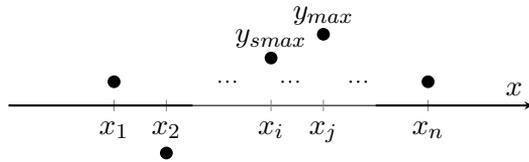


Рис. 12

Простыми линейными преобразованиями получим функцию $f'(x) = N \cdot (f(x + x_j) - y_{max})$, где $N > \frac{|x_1 - x_j|}{|y_{max} - y_{smax}|}$ и функцию $f''(x) = \frac{1}{y_{max}} \cdot f(x + x_j)$. Подставим функцию $f'(x)$ в функцию $f''(x)$. Так как функция $f'(x)$ равна нулю только в точке ноль, а вне нуля строго меньше $-|x_j - x_1|$, то результирующая функция $g(x)$ будет всюду равна нулю, а в нуле равна единице. Из такой функции с одной выколотой точкой легко построить произвольную функцию $g(x) \in L^{(1)}$ с n выколотыми точками.

$$g(x) = \sum_{i=1}^n b_i \cdot p(x + a_i) + (a \cdot x + b) \quad (12)$$

Лемма доказана. □

Следствие 4. Класс $L^{(1)}$ является единственным максимальным собственным 1-следом в классе $L'^{(1)}$.

Пусть $M \subseteq PFS^{(1)}$ тогда верна следующая лемма.

Лемма 4. $PFS^{(1)} \subseteq [M \cup L] \iff M \not\subseteq L'^{(1)}$

Доказательство. Пусть $f(x) \in PFS^{(1)} \setminus L'^{(1)}$. Как и ранее будем считать, что функция $f(x)$ на концах равна нулю и имеет единственный максимум. Рассмотрим два случая:

1) Максимум функции $f(x)$ это точка. Тогда по аналогии с предыдущей теоремой из функции $f(x)$ можно получить все функции из $L'^{(1)}$. К функции $f(x)$ прибавим такую функцию из $L'^{(1)}$, чтобы у результирующей функции $f'(x)$ максимум был интервал. Пусть функция $f'(x)$ выглядит как показано на Рис. 13.

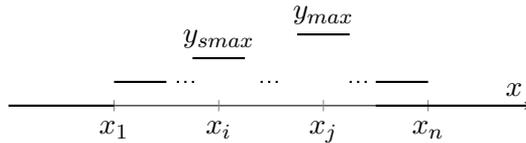


Рис. 13

Рассмотрим две функции. Первая функция $g_1(x) = N \cdot (f'(x + x_j) - y_{max})$, где $N > \frac{|x_1 - x_j|}{|y_{max} - y_{smax}|}$ и функцию $g_2(x) = \frac{1}{y_{max}} \cdot f'(x + x_j)$. Подставим функцию $g_1(x)$ в функцию $g_2(x)$. Так как функция $g_1(x)$ равна нулю в некой окрестности точки ноль, а вне этой окрестности строго меньше нуля и при этом, функция $g_2(x)$ равна нулю при $x < -|x_j - x_1|$, то результирующая функция $l(x)$ будет равна единице в некой окрестности точки ноль а вне окрестности равна нулю (Рис. 14).

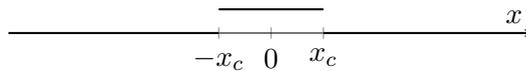


Рис. 14

Пусть $g(x)$ произвольная функция из $PFS^{(1)}$. Тогда $g(x)$ можно представить в следующем виде:

$$g(x) = \sum_{i=1}^m b_i \cdot l(c_i \cdot x + a_i) + \sum_{i=1}^n b'_i \cdot p(x + a'_i) + (a \cdot x + b) \quad (13)$$

2) Максимум функции $f(x)$ это интервал. Построим функцию $l(x)$ так же как и в первой части. Далее рассмотрим функцию:

$$p'(x) = l(x) - l(2 \cdot x - x_c) - l(2 \cdot x + x_c) \quad (14)$$

Функция $p'(x)$ равна единице только в точке ноль, а в остальных случаях функция равна нулю. То есть мы получили функцию с одной выколотой точкой, а следовательно можем получить весь класс $L'^{(1)}$. Как и в первой части доказательства мы получили функции $p(x)$ и $l(x)$ а значит можем получить весь класс PFS . Лемма доказана. \square

Следствие 5. *Класс $L'^{(1)}$ является единственным максимальным собственным 1-следом в классе $PFS^{(1)}$.*

Пусть $M \subseteq PP^{(1)}$ тогда верна следующая лемма.

Лемма 5. $PP'^{(1)} \subseteq [M \cup L] \iff M \not\subseteq PFS^{(1)}$

Доказательство. Данная теорема была доказана в работе [1]. \square

Следствие 6. *Класс $PFS^{(1)}$ является единственным максимальным собственным 1-следом в классе $PP^{(1)}$.*

Пусть функция $f(x) \in FS^{(1)} \setminus PFS^{(1)}$, тогда верна следующая лемма.

Лемма 6. $FS^{(1)} \subseteq [PFS^{(1)} \cup L \cup \{f(x)\}]$.

Доказательство. Пусть имеются все функции из $PFS^{(1)}$ и линейные функции, а также функция $f(x) \in FS^{(1)} \setminus PFS^{(1)}$. Без ограничения общности будем считать, что $f(x) = 0$, при $|x| > N, N \in \mathbb{R}$. Так как $(x) \notin PFS^{(1)}$, то существует интервал (x_i, x_{i+1}) , где функция не равна константе. Пусть функция $f(x)$ выглядит ка показано на Рис. 15.

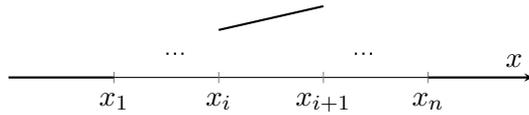


Рис. 15

Построим две функции:

1) Прибавим к функции $f(x)$ функцию $l_1(x) \in PFS^{(1)}$, которая увеличит значение результирующей функции $f'(x)$ на интервале (x_i, x_{i+1}) так чтобы значение функции $f'(x)$ на этом интервале было много больше чем вне этого интервала. Простыми линейными преобразованиями изменим функцию $f'(x)$ так чтобы у результирующей функции $f''(x)$ данный интервал изменился на интервал $(0, 1)$, функция реализующаяся на этом интервале была проводником, а вне интервала $(0, 1)$ функция была много меньше нуля.

2) Прибавим к функции $f(x)$ функцию $l_2(x) \in PFS^{(1)}$, которая опустит у результирующей функции $g'(x)$ значение функции на интервале (x_i, x_{i+1}) к оси Ox . Простыми линейными преобразованиями изменим функцию $g'(x)$ так чтобы у результирующей функции $g''(x)$ данный интервал изменился на интервал $(0, 1)$, функция реализующаяся на этом интервале была проводником, и при $x < N, N \in \mathbb{R}$, функция $g''(x)$ равнялась нулю. Подставим функцию $f''(x)$ в функцию $g''(x)$ получим функцию $h(x)$ (Рис. 16), которая на интервале $(0, 1)$ равна функции "проводник а вне интервала равна нулю.

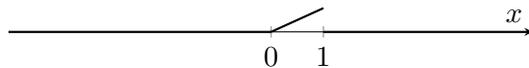


Рис. 16

Пусть $g(x)$ произвольная функция из $FS^{(1)}$. Тогда $g(x)$ можно представить как:

$$g(x) = \sum_{i=1}^m b_i \cdot h(c_i \cdot x + a_i) + r(x) \quad (15)$$

где $r(x) \in PFS^{(1)}$. Лемма доказана. □

Пусть функция $f(x) \in FS^{(1)} \setminus CFS^{(1)}$, тогда верна следующая лемма.

Лемма 7. $FS^{(1)} \subseteq [CFS^{(1)} \cup L \cup \{f(x)\}]$.

Доказательство. Пусть функция $f(x) \in FS^{(1)} \setminus CFS^{(1)}$. Следовательно существует такая точка разрыва a , что функция $f(x)$ в этой точке равна c , а в малой окрестности слева(справа) от этой точки равна некой линейной функции. Линейными преобразованиями сделаем так чтобы функция имела точку разрыва в нуле и равнялась единице в этой точке, а в малой окрестности слева равнялась константе ноль (Рис. 17).

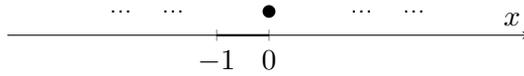


Рис. 17

Рассмотрим функцию $g(x) \in CFS^{(1)}$, которая имеет вид как показано на Рис. 18.

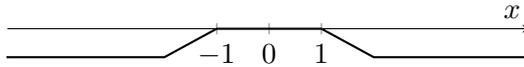


Рис. 18

Функция $g(x)$ на концах равняется константе c , $-1 < c < 0$. Следовательно, подставив функцию $g(x)$ в функцию $f(x)$ мы получим функцию $h(x)$, которая имеет вид как показано на Рис 19.

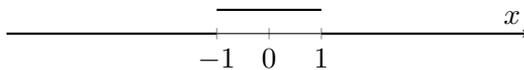


Рис. 19

Из Леммы 4 следует что с помощью функции $h(x)$ и линейных функций мы можем получить все функции из множества $PFS^{(1)}$. Возьмем произвольную функцию $w(x) \in CFS^{(1)} \setminus PFS^{(1)}$. Из леммы 6 следует, что $FS^{(1)} \subseteq [\{w(x)\} \cup PFS^{(1)} \cup L]$. Лемма доказана. \square

Пусть $M \subseteq FS^{(1)}$ тогда верна следующая лемма.

Лемма 8. $FS^{(1)} \subseteq [M \cup L] \iff M \not\subseteq PFS^{(1)} \text{ и } M \not\subseteq CFS^{(1)}$

Доказательство. Пусть $f_c(x) \in FS^{(1)} \setminus CFS^{(1)}$. Из леммы 7 следует что можно получить функцию $f'_c(x)$, которая в нуле равняется единице а в малой окрестности слева от нуля равняется константе ноль (Рис. 17). Далее рассмотрим функцию $g(x) \in M, g(x) \notin L$. Без ограничения общности можно считать что $g(x) = 0$, при $|x| > N, N \in \mathbb{R}$. Следовательно функция $g(x)$ имеет точку максимума или точку где функция стремится к максимуму. Будем считать что такая точка единственная. Рассмотрим случаи:

1) Максимум принимается на интервале(полуинтервале, отрезке) (x_1, x_2) и функция $g(x)$ равняется константе c на этом интервале(полуинтервале, отрезке). С помощью линейных преобразований сделаем так чтобы максимум функции $g(x)$ равнялся нулю и принимался на интервале(полуинтервале, отрезке) $(-1, 1)$, а вне этого интервала удовлетворяла следующим ограничениям $0 > g(x) > -1$. Тогда подставив функцию $g(x)$ в функцию $f'_c(x)$ получим функцию $h(x)$ (Рис. 19). Из леммы 7 следует что с помощью функций $h(x)$ и $f_p(x) \notin PFS$ и линейных функций можно получить все множество $FS^{(1)}$.

2) Максимум принимается(стремится к максимуму) на интервале(полуинтервале, отрезке) (x_1, x_2) и функция $g(x)$ равняется линейной функции $l(x) \in L^{(1)}$ на этом интервале(полуинтервале, отрезке). Рассмотрим функцию $g'(x) = g(x) + g(-x + x_1 + x_2)$. Данная функция является суммой функции $g(x)$ и этой же функции отраженной относительно центра интервала(полуинтервала, отрезка) (x_1, X_2) . Следовательно функция $g'(x)$ на интервале(полуинтервале, отрезке) будет равняться некой константе c . Свели к случаю 1. 3) Максимум принимается(стремится к максимуму) на интервале(полуинтервале, отрезке) (x_1, x_2) и функция $g(x)$ равняется линейной функции $l_1(x) \in L^{(1)}$ на интервале(полуинтервале) $(x_1, (x_1 + x_2)/2)$ и равняется линейной функции $l_2(x) \in L^{(1)}$ на интервале(полуинтервале) $((x_1 + x_2)/2, x_2)$, причем $l_1(x) = l_2(-x + x_1 + x_2)$. Рассмотрим функцию $g'(x) = g(x) + g(-x + x_1 + (x_1 + x_2)/2)$. Данная функция имеет максимум на интервале(полуинтервал, отрезке) $(x_1, (x_1 + x_2)/2)$ и равна некой константе на этом интервале(полуинтервал, отрезке). Свели к случаю 1.

4) Максимум функции $g(x)$ это выколота точка. Аналогично теореме 3, рассмотрим две функции. Первая функция $g'(x)$ имеет максимум в точке ноль и значение максимума равно нулю, при этом вне нуля функция всюду меньше некого значения $N \in \mathbb{R}$. Вторая функция $g''(x)$ равняется единице в точке ноль, а при $|x| > N$ равняется нулю. Подставив функ-

цию $g'(x)$ в функцию $g''(x)$ мы получим функцию с одной выколотой точкой, тогда из леммы 3 следует что мы можем получить все функции из множества $L'(1)$. Прибавим к функции $g(x)$ нужную функцию $l' \in L'(1)$ мы избавимся от всех выколотых точек. Свели к первым трем случаям. Лемма доказана. \square

Следствие 7. *Класс $PFS^{(1)}$ и класс $CFS^{(1)}$ является максимальными собственными 1-следами класса $PP^{(1)}$ и других таких классов не существует.*

Пусть функция $f(x) \in FP^{(1)} \setminus FS^{(1)}$, тогда верна следующая лемма.

Лемма 9. $FP^{(1)} \subseteq [FS^{(1)} \cup L \cup \{f(x)\}]$.

Доказательство. Пусть функция $g(x) \in FP^{(1)} \setminus FS^{(1)}$. Без ограничения общности можно считать что функция $g(x) = 0$, при $x < -N$ и $g(x) = c$, при $x \geq N$, где $N \in \mathbb{R}$. Рассмотрим функцию $g_s(x)$, такую что $g_s(x) = g(x)$, при $x < N$ и $g_s(x) = 0$, при $x \geq N$. Очевидно, функция $g_s(x)$ принадлежит множеству $FS^{(1)}$. Построим функцию $w(x) = g(x) - g_s(x)$. Данная функция будет равняться нулю, при $x < N$ и равняться константе c , при $x \geq N$. Следовательно функция $w(x)$ принадлежит множеству $PP^{(1)}$ одностепенных кусочно параллельных функций но не принадлежит классу $PFS^{(1)}$. Из леммы 5 следует что из функции $w(x)$ и линейных функций можно получить все множество $PP^{(1)}$. Пусть функция $f(x)$ произвольная функция из множества $FP^{(1)}$. Без ограничения общности можно считать что функция $f(x) = 0$, при $x < -N$ и $f(x) = c$, при $x > N$, где $N \in \mathbb{R}$. Представим функцию $f(x)$ как сумму двух функций:

$$f(x) = f_s(x) + f_p(x) \quad (16)$$

где

$$f_s(x) = \begin{cases} f(x), & \text{если } x < N \\ 0, & \text{если } x \geq N \end{cases} \quad (17)$$

$$f_p(x) = \begin{cases} 0, & \text{если } x < N \\ c, & \text{если } x \geq N \end{cases} \quad (18)$$

Так как функция $f_s(x)$ принадлежит множеству $FS^{(1)}$, а функция $f_p(x)$ принадлежит множеству $PP^{(1)}$, то мы можем получить произвольную функция $f(x)$ из множества $FP^{(1)}$. Лемма доказана. \square

Пусть функция $f(x) \in FP^{(1)} \setminus PP^{(1)}$, тогда верна следующая лемма.

Лемма 10. $FP^{(1)} \subseteq [PP^{(1)} \cup L \cup \{f(x)\}]$.

Доказательство. Пусть функция $g(x) \in FP^{(1)} \setminus PP^{(1)}$. Без ограничения общности можно считать что функция $g(x) = 0$, при $x < -N$ и $g(x) = c$, при $x \geq N$, где $N \in \mathbb{R}$. Из леммы 9 следует что функцию $g(x)$ можно представить как сумму двух функций $g_s(x) \in FS^{(1)}$ и $g_p(x) \in PP^{(1)}$. Следовательно можно получить функцию $g_s(x) = g(x) - g_p(x)$. Функция $g_s(x)$ принадлежит множеству $FS^{(1)}$, но не принадлежит множеству $PFS^{(1)}$. Так как $PFS^{(1)} \subseteq PP^{(1)}$, то из леммы 6 следует, что имея все множество $PFS^{(1)}$ и функцию не принадлежащую этому множеству можно получить все множество $FS^{(1)}$. Имея множество $PP^{(1)}$ и множество $FS^{(1)}$ из леммы 9 следует что можно получить произвольную функцию из $FP^{(1)}$. Лемма доказана. \square

Пусть функция $f(x) \in FP^{(1)} \setminus CFP^{(1)}$, тогда верна следующая лемма.

Лемма 11. $FP^{(1)} \subseteq [CFP^{(1)} \cup L \cup \{f(x)\}]$.

Доказательство. Пусть функция $g(x) \in FP^{(1)} \setminus CFP^{(1)}$. Без ограничения общности можно считать что функция $g(x) = 0$, при $x \leq 0$ и $g(x) = c$, при $x \geq N$, где $N \in \mathbb{R}$. Рассмотрим функцию $g'(x) = g(x) + g(-x)$. Функция $g'(x)$ равняется константе c , при $|x| > N$, следовательно данная функция принадлежит множеству $FS^{(1)}$. Так как функция $g'(x) = g(x)$ при $x > 0$, то она не принадлежит множеству $CFS^{(1)}$. Из леммы 7 следует, что имея все множество $CFS^{(1)}$ и функцию не принадлежащую этому множеству можно получить все множество $FS^{(1)}$. Из леммы 9 следует, что имея все множество $FS^{(1)}$ и функцию не принадлежащую этому множеству можно получить все множество $FP^{(1)}$. Лемма доказана. \square

Пусть $M \subseteq FP^{(1)}$ тогда верна следующая лемма.

Лемма 12. $FP^{(1)} \subseteq [M \cup L] \iff M \not\subseteq PP^{(1)}, M \not\subseteq CFP^{(1)} \text{ и } M \not\subseteq FS^{(1)}$.

Доказательство. Пусть функция $g_{cfp}(x) \in FP^{(1)} \setminus CFP^{(1)}$. Без ограничения общности можно считать что функция $g(x) = 0$, при $x \leq 0$ и $g(x) = c$, при $x \geq N$, где $N \in \mathbb{R}$. Аналогично утверждению 5 получим функцию $g'_{cfs}(x)$, которая принадлежит множеству $FS^{(1)}$, но не принадлежит множеству $CFS^{(1)}$. Пусть функция $g_{pp}(x) \notin PP^{(1)}$. Без ограничения общности можно считать что функция $g(x) = 0$, при $x \leq 0$ и $g(x) = c$, при $x \geq N$, где $N \in \mathbb{R}$. Рассмотрим функцию $g_{pfs}(x) = g_{pp}(x) + g_{pp}(-x)$.

Функция $g_{pfs}(x)$ равняется константе c , при $|x| > N$, следовательно данная функция принадлежит множеству $FS^{(1)}$. Так как функция $g_{pfs}(x) = g_{pp}(x)$ при $x > 0$, то она не принадлежит множеству $PFS^{(1)}$. Из теоремы 6 следует что имея функцию $g_{pfs}(x)$ и функцию $g_{cfs}(x)$, а также все линейные функции можно получить все множество $FS^{(1)}$. Из утверждения 3 следует что имея множество $FS^{(1)}$ и функцию не принадлежащую этому множеству, а также все линейные функции можно получить все множество $FP^{(1)}$ Лемма доказана. \square

Следствие 8. *Класс $PP^{(1)}$, класс $FS^{(1)}$ и класс $CFP^{(1)}$ является максимальными собственными 1-следами класса $FP^{(1)}$ и других таких классов не существует.*

Пусть функция $f(x) \in PL^{(1)} \setminus FP^{(1)}$, тогда верна следующая лемма.

Лемма 13. $PL^{(1)} \subseteq [FP^{(1)} \cup L \cup \{f(x)\}]$.

Доказательство. Пусть функция $g(x) \in PL^{(1)} \setminus FP^{(1)}$. Без ограничения общности можно считать что функция $g(x) = c$, при $x < N$ и $g(x) = a \cdot x$, при $x \geq 0$, где $N, a, c \in \mathbb{R}$. Возьмем функцию $g_{fp}(x) \in FP^{(1)}$.

$$g_{fp}(x) = \begin{cases} g(x), & \text{если } x < 0 \\ 0, & \text{если } x \geq 0 \end{cases} \quad (19)$$

Отнимем функцию $g_{fp}(x)$ от функции $g(x)$.

$$d(x) = g(x) - g_{fp}(x) = \begin{cases} 0, & \text{если } x < 0 \\ a \cdot x, & \text{если } x \geq 0 \end{cases} \quad (20)$$

Из работы [2] известно, что с помощью функции $d(x)$ и всех линейных функций можно получить все множество $CPL^{(1)}$.

Пусть $f(x)$ произвольная функция из множества $PL^{(1)}$. Без ограничения общности можно считать что функция $f(x) = c$, при $x < N$ и $g(x) = a \cdot x$, при $x \geq 0$, где $N, a, c \in \mathbb{R}$. Представим функцию $f(x)$ в виде суммы двух функций.

$$f(x) = f_{fp}(x) + f_{cpl}(x) \quad (21)$$

Где функция $f_{fp}(x) \in FP^{(1)}$.

$$f_{fp}(x) = \begin{cases} f(x), & \text{если } x < 0 \\ 0, & \text{если } x \geq 0 \end{cases} \quad (22)$$

Функция $f_{cpl}(x) \in CPL^{(1)}$.

$$f_{cpl}(x) = \begin{cases} 0, & \text{если } x < 0 \\ a \cdot x, & \text{если } x \geq 0 \end{cases} \quad (23)$$

Лемма доказана. \square

Пусть функция $f(x) \in PL^{(1)} \setminus CPL^{(1)}$, тогда верна следующая лемма.

Лемма 14. $PL^{(1)} \subseteq [CPL^{(1)} \cup L \cup \{f(x)\}]$.

Доказательство. Пусть функция $g(x) \in PL^{(1)} \setminus CPL^{(1)}$. Без ограничения общности можно считать что функция имеет разрыв в точке ноль, значение функции в точке разрыва равно единице, а на интервале $(-e, 0)$, где $e \in \mathbb{R}_+$, функция равна константе ноль. Возьмем функцию $g_{cpl}(x) \in CPL^{(1)}$.

$$g_{cpl}(x) = \begin{cases} -e, & \text{если } x < -e \\ x, & \text{если } -e \leq x < 0 \\ 0, & \text{если } x \geq 0 \end{cases} \quad (24)$$

Подставим функцию $g_{cpl}(x)$ в функцию $g(x)$. Получим функцию, которая равна нулю при $x < 0$ и равна единице при $x \geq 0$, то есть функцию $\Theta(x)$. Из работы [1] известно что имея функцию $\Theta(x)$ и все линейные функции можно получить все множество $FP^{(1)}$. Так как функция $g_{cpl}(x)$ принадлежит множеству $FP^{(1)}$, но не принадлежит множеству $PP^{(1)}$, то из утверждения 4 следует что можно получить все множество $FP^{(1)}$. А имея множество $CPL^{(1)}$ и множество $FP^{(1)}$ можно получить все множество $PL^{(1)}$ (утверждение 7). Лемма доказана. \square

Пусть $M \subseteq PL^{(1)}$ тогда верна следующая лемма.

Лемма 15. $PL^{(1)} \subseteq [M \cup L] \iff M \not\subseteq FP^{(1)} \text{ и } M \not\subseteq CPL^{(1)}$

Доказательство. Покажем что можно получить все множество $FP^{(1)}$. Пусть функция $g_{nc}(x)$ не принадлежит множеству $CPL^{(1)}$. Без ограничения общности можно считать что функция $g_{nc}(x) = 0$, при $x \geq 0$ и

$g_{nc}(x) = a \cdot x + b$, при $x < -N$, где $N, a, b \in \mathbb{R}$. Рассмотрим функцию $g'_{nc}(x) = g_{nc}(x) - g_{nc}(x + N)$.

$$g'_{nc}(x) = \begin{cases} 0, & \text{если } x \geq 0 \\ g_{nc}(x), & \text{если } -N < x < 0 \\ \cdot \\ \cdot \\ const, & \text{если } x < -2 \cdot N \end{cases} \quad (25)$$

Из уравнения следует что функция $g'_{nc}(x)$ принадлежит множеству $FP^{(1)}$ и так как на интервале $(-N, 0)$, функция $g'_{nc}(x)$ равна функции $g_{nc}(x)$, она не содержится в множестве $CFP^{(1)}$. Далее рассмотрим функцию $g_{nfp}(x)$ которая не содержится в множестве $FP^{(1)}$. Без ограничения общности можно считать что функция $g_{nfp}(x) = 0$, при $x \geq 0$ и $g_{nfp}(x) = a \cdot x + b$, при $x < -N$, где $N, a, b \in \mathbb{R}, a \neq 0$. Рассмотрим функцию $g'_{nc}(x) = g_{nc}(x) - g_{nc}(x + N + e), e \in \mathbb{R}$.

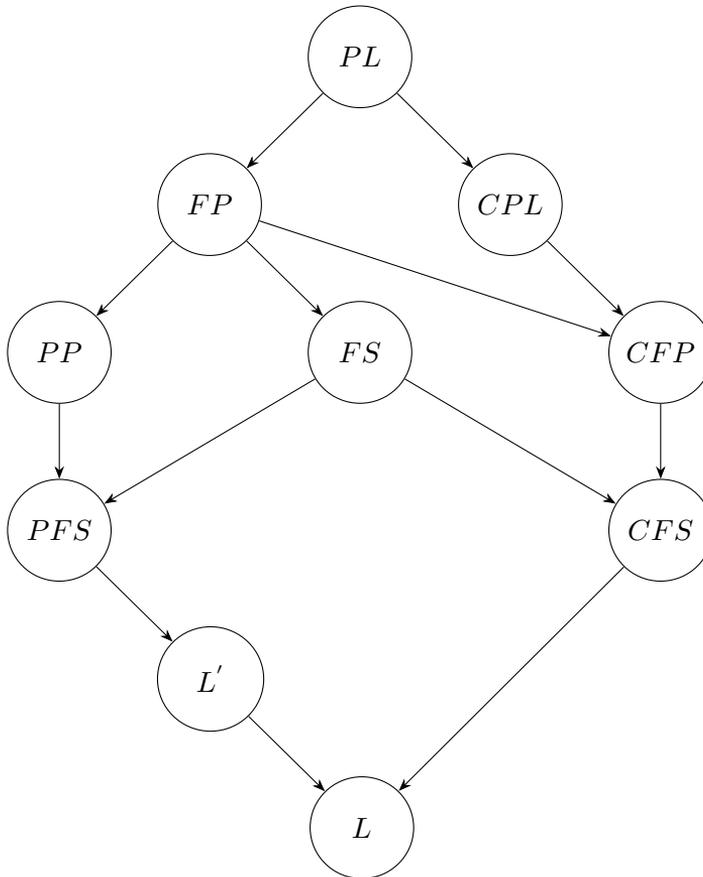
$$g'_{nfp}(x) = \begin{cases} 0, & \text{если } x \geq 0 \\ g_{nfp}(x), & \text{если } -N < x < 0 \\ \cdot \\ \cdot \\ a \cdot x + b, & \text{если } -(N + e) < x < -N \\ \cdot \\ \cdot \\ const, & \text{если } x < -2 \cdot N + e, const \neq 0 \end{cases} \quad (26)$$

Из уравнения следует что функция $g'_{nfp}(x)$ принадлежит множеству $FP^{(1)}$, но так как константа $const \neq 0$, не принадлежит множеству $FS^{(1)}$. На интервале $(-(N + e), -N)$ функция $g'_{nfp}(x)$ равна линейной функции отличной от константы, следовательно данная функция не принадлежит множеству $PP^{(1)}$. Из лемма 12 следует что множество $(g'_{nc}, g'_{nfp} \cup L)$ порождает множество $FP^{(1)}$. Из лемма 9 следует что имея все множество $FP^{(1)}$ и функцию $g_{nfp}(x) \notin FP^{(1)}$, можно получить все множество $PL^{(1)}$. Лемма доказана. \square

Следствие 9. *Класс $FP^{(1)}$ и класс $CPL^{(1)}$ является максимальными собственными 1-следами класса $PL^{(1)}$ и других таких классов не существует.*

5. Основная теорема решетки 1-следов замкнутых классов кусочно-линейных функций

Рассмотрим граф G_1 :



Решетка 1-следов

где вершинами графа являются соответствующие 1-следы замкнутых классов, а направленное ребро из вершины A в вершину B означает, что B является максимальным собственным подмножеством 1-следа A . Тогда верна следующая теорема.

Теорема 1. *Граф G_1 описывает полную решетку 1-следов замкнутых классов кусочно-линейных функций.*

Доказательство. Доказательство данной теоремы вытекает из лемм 1 – 15 и следствий 1 – 9. □

6. Заключение.

В данной статье была описана решетка 1-следов замкнутых классов кусочно-линейных функций. Дальнейшей задачей станет описание решетки 2-следов в этом классе. А также для каждого множества из решетки 1-следов найти все замкнутые классы проекция на одноместные функции которых совпадает с данным множеством.

Автор выражает искреннюю признательность А. А. Часовских за постановку задачи, обсуждение результатов, советы и замечания.

Список литературы

- [1] В. С. Половников, *Об оптимизации структурной реализации нейронных сетей*, диссертация на соискание ученой степени канд. физ.-матем. наук, Москва, 2007.
- [2] В. Б. Кудрявцев, С. В. Алешин, А. С. Подколзин, *Введение в теорию автоматов*, Наука, Москва, 1985, 320 pp.
- [3] А. Н. Кан, “Вопросы выразимости в классе нейронных функций”, *Интеллектуальные системы. Теория и приложения*, **19:1** (2015), 15–21
- [4] А. Н. Кан, “Вопросы полноты в классе кусочно-линейных непрерывных функций”, *Интеллектуальные системы. Теория и приложения*, **21:4** (2017), 46–56
- [5] А. Н. Кан, “Вопросы выразимости в классе согласованных функций”, *Интеллектуальные системы. Теория и приложения*, **23:2** (2019), 125–133
- [6] П. П. Степанов, “Искусственные нейронные сети”, *Молодой учёный*, **138:4** (2017), 185–187

The lattice of 1-traces of closed classes of piecewise linear functions Kan A.N.

In the present paper, we find the lattice of 1-traces of closed classes of piecewise linear functions.

Key words: piecewise linear functions, continuous piecewise linear functions, finite-parallel functions, continuous finite-parallel functions, continuous finite linear functions, piecewise parallel functions, finite linear functions, parallel finite linear functions, 1-trace lattice, Heaviside function, 2-precomplete class.

Часть 4.
Материалы семинаров кафедры
MaTIC

Доклады семинара «Вопросы сложности алгоритмов поиска»

В весеннем семестре 2019 – 2020 учебного года на научном семинаре «Вопросы сложности алгоритмов поиска» под руководством профессора Эльяра Эльдаровича Гасанова состоялось 20 докладов.

19 февраля 2020 года

О прогнозировании сверхслов автоматами

студ. магистратуры Кузнецова Е.В.

В докладе рассматривается клеточный автомат, заданный на бесконечной в одну сторону полосе. Изучается класс S^1 законов движения этого автомата, в котором нет двух единиц подряд. Основной целью является определение наименьшего числа состояний клеточного автомата, при котором можно реализовать все законы движения из данного класса. Построен клеточный автомат с пятью состояниями, реализующий законы движения из рассматриваемого класса. Также показано, что оценка на количество состояний не улучшаема, то есть четырех состояний автомата не достаточно для реализации всех законов движения из класса.

26 февраля 2020 года

О синтезе колонии жуков с линейным ростом

студ. специалитета Воронников А. С.

Рассматривается следующая динамическая система. Дано бесконечное поле с ненулевым однородно расположенным запасом еды. На этом поле появляется жук, который перемещается по полю, ест имеющуюся еду и размножается. Поле моделируется целочисленной решёткой на плоскости, в которой каждому узлу в начальный момент сопоставлено некоторое одинаковое количество еды. Эту целочисленную решётку в дальнейшем будем называть картой. Жук действует по алгоритму, который упрощённо представляется схемой: искать еду \rightarrow есть, пока не насытишься (если еды не хватило, снова искать) \rightarrow размножаться. На все действия расходуется энергия, которая получается жуком из еды,

расположенной на поле. Если запас энергии жука падает ниже поля, то жук умирает — исчезает с поля. Под размножением подразумевается деление жука на двух одинаковых жуков, обладающих половиной запаса энергии родителя.

Для некоторого класса прямых необходимо построить класс колоний, такой, что для произвольной прямой найдётся колония, чей график численности бесконечное число раз пересекает выбранную прямую. Помимо этого требуется описать такой класс прямых.

Указанная система моделируется однородными структурами. Предлагается использовать подход автоматного моделирования.

Сначала выбирается пять условий, которым должна удовлетворять колония, затем показывается, к каким ограничениям приводят эти условия. Далее следует анализ поведения жуков на карте, из которого вытекает необходимость корректировки некоторых условий. В результате непосредственного разбора поведения всех жуков, удаётся выписать явную функцию численности популяции с некоторого достаточно большого момента времени. Оказывается, что она лежит в конусе, ограниченном двумя прямыми, причём достигает границ конуса бесконечное число раз.

Таким образом, для произвольной прямой из класса

$$\mathcal{A} = \{y = ax + b \mid 0 < a \leq \frac{40}{11}, b \in \mathbb{R}\}$$

удалось построить такую колонию, что её график численности популяции бесконечное число раз пересечёт выбранную прямую.

Обозначены требующие дальнейшего детального разбора способы расширения класса прямых до класса

$$\mathcal{B} = \{y = ax + b \mid a > 0, b \in \mathbb{R}\}.$$

4 марта 2020 года

Частичное предугадывание на бесконечности и на отрезке

студ. МГТУ им Н.Э.Баумана Маншилин О.

В докладе рассматривается частичное прогнозирование (предугадывание) множеств сверхслов в алфавите $0, 1$ с помощью конечных автоматов. Определяется степень прогнозирования на бесконечности автоматом как нижний предел доли верно угаданных автоматом символов

на начальном отрезке длины t при стремлении t к бесконечности. Также вводится степень прогнозирования на отрезке. Это наименьшая доля угаданных символов на отрезке длины N на всех таких отрезках, начиная с некоторого момента. Если степень ненулевая, множество частично прогнозируемо на бесконечности или на отрезке.

Показано, что общерегулярное множество частично прогнозируемо автоматом на бесконечности тогда и только тогда, когда оно частично прогнозируемо на некотором отрезке. Для произвольных множеств это утверждение неверно.

8 апреля 2020 года

О вопросе сложности расшифровки функций фиксированного веса запросами на сравнение

асп. Быстрыгова А.В.

В докладе рассматривается задача расшифровки функций фиксированного веса запросами на сравнение. Получена верхняя оценка сложности расшифровки запросами на сравнение этого класса функций. На примере функций веса один показано, что данная верхняя оценка совпадает с нижней оценкой.

15 апреля 2020 года

Сложность расшифровки функций веса 2 запросами на сравнение

асп. Быстрыгова А.В.

В докладе рассматривается задача расшифровки функций веса 2 запросами на сравнение. Показано, что верхняя оценка, приведенная в предыдущем докладе, совпадает с нижней оценкой, тем самым сложность расшифровки этого класса запросами на сравнение равна $\lfloor 2^{n+1}/3 \rfloor$.

22 апреля 2020 года

На семинаре были представлены на рассмотрение две выпускные работы.

Моделирование двунаправленного движения на луче клеточными автоматами

студ. магистратуры Кузнецова Е.В.

В работе рассматривается реализация одного класса законов движения клеточным автоматом на бесконечном экране. Показано, что минимальное число состояний клеточного автомата, моделирующего направленное движение точки на луче, при котором точка не совершает подряд два движения вправо, равно пяти.

Сложность реализации сверхбольших перестановок

студ. бакалавриата Ковальский А.С.

В работе рассматривается задача организации хранения в оперативной памяти информации о файлах, при условии что в оперативной памяти может разместиться лишь $1/10$ всей таблицы соответствия физических адресов логическим адресам. Цель — уменьшение количества обращений к диску при трех типах запросах (простейшие запросы, случайные запросы и запросы постоянной длины с фиксированным распределением). В работе предложены три алгоритма реализации хранения информации о файлах, которые основаны на использовании очереди, очереди с приоритетами и комбинации очереди с приоритетами с обычной очередью соответственно. Экспериментальным путем показано, что лучшим для реализации оперативной памяти при простейших запросах является алгоритм с применением только очереди с приоритетами. На запросах постоянной длины алгоритмы с применением приоритетной очереди дают схожие результаты. Более того, оказалось, что для случая, когда заранее не известен тип запросов к файлам, лучшим является алгоритм, основанный на комбинации очереди с приоритетами и обычной очереди.

6 мая 2020 года

На семинаре были представлены на рассмотрение три курсовые работы.

Алгоритм достижения консенсуса в криптовалютах с помощью лотореи

студ. магистратуры Суюнбекова М.Б.

Данная работа посвящена алгоритму достижения консенсуса при выборе одного участника из многих как победителя.

Предлагается применение алгоритма в концепции отраслевой криптовалюты для решения следующего вопроса: блок транзакций которого из участников нужно добавить следующим в блокчейн. Посчитаны верхняя и нижняя оценки количества операций обмена для одного участника, а также для всех участников в процессе выявления победителя. Также в работе рассмотрен некий класс мошенников и возможные пути борьбы с ними.

Синтез одной колонии жуков с линейным ростом численности

студ. специалитета Воротников А.С.

Рассматривается динамическая система, описывающая поведение популяции жуков. Жуки живут на поле, которое в начальный момент представляет собой целочисленную решётку, к каждой клетке которой в начальный момент находится одинаковое количество еды для жуков. Жуки в соответствии с некоторым алгоритмом перемещаются по полю, едят расположенную в нём еду и размножаются, причём на все действия расходуется энергия. Система моделируется однородными структурами. В работе показано, что для любой прямой из некоторого класса существует колония, чей линеаризованный график численности бесконечное число раз пересекает выбранную прямую.

Моделирование аэродинамики крыла клеточными автоматами

студ. специалитета Гордеева А.С.

В работе исследуется задача моделирования движения полета крыла в воздушном потоке. Предлагается для моделирования использовать клеточные автоматы. Причем считается, что имеются клеточные автоматы, моделирующие движение воздуха, и имеется автомат, моделирующий крыло. Крыло имеет некоторую форму. Клеточные автоматы изображают прямолинейное движение частиц, но при столкновении с крылом «обтекают» его, причем скорость частиц, движущихся по более длинной стороне крыла, больше скорости частиц с другой стороны. Из-за этого возникает подъемная сила. Автомат, моделирующий крыло, «видит» клеточные автоматы из некоторой окрестности и высчитывает скорость частиц. На основе этого вычисляется вектор подъемной силы. В результате чего крыло меняет свои координаты. Написана компьютерная программа, которая отображает функционирование предложенных автоматов. Проведены эксперименты с крыльями разных форм.

13 мая 2020 года

На семинаре были представлены на рассмотрение четыре курсовые работы.

Компьютерное моделирование дорожных ситуаций в соответствии с китайскими правилами дорожного движения

студ. магистратуры Ван Кэ

В работе анализируются правила дорожного движения с целью создания модели дорожного движения. Модель движения создаётся в два этапа. Первый — преобразование текста правил в модель, второй — анализ конкретной дорожной ситуации на перекрестке и анализ состояния движения нескольких движущихся объектов на перекрестке (порядок прохождения). В работе используется система семантического анализа для моделирования законов и правил дорожного движения. Помимо этого, для анализа правил применяются два метода: семантическая зависимость и сегментация китайских слов.

Семантический анализ китайских правил дорожного движения

студ. магистратуры Сюэ Фенхао

Работа посвящена моделированию возможных правил дорожного движения с помощью документов китайского законодательства о дорожном движении. Сделана программа, которая считывает юридический документ и выводит соответствующую ему формальную модель. В этом документе приводятся несколько возможных дорожных ситуаций и их соответствующие модели. Из-за наличия грамматических и семантических особенностей китайско-тибетского языкового семейства, данная работа посвящена изучению того, как анализировать и интерпретировать документы китайского дорожного права с помощью семантического анализа и преобразовывать ключевые слова документа для компьютерного распознавания.

Построение кратчайших путей с помощью клеточных автоматов с локаторами

студ. специалитета Пропажин А.А.

В работе рассматривается задача построения кратчайшего пути с помощью нового объекта — клеточных автоматов с локаторами. Этот объект позволяет ускорить процесс нахождения в несколько раз, при этом сделав активными меньшее количество клеток, чем в способе, описанном в ранее опубликованных работах. В работе приводится способ построения кратчайшего пути с препятствием.

Моделирование выбора оптимальной ставки по кредиту

студ. бакалавриата Кдырхан Т.Н.

В работе рассматривается задача моделирования наиболее выгодной для коммерческих банков или банков второго уровня, ставки по кредиту. По умолчанию, услугами банка пользуются предприятия. В задаче используется вычислительная способность компьютера, включающий в себя программу, способная посчитать наиболее оптимальную ставку по кредиту. Показаны графики зависимости максимальной выгоды банка от ставки по кредиту.

20 мая 2020 года

На семинаре были представлены на рассмотрение две курсовые работы.

Моделирование перемагничивания спиновых вентилей под действием поля рассеяния наночастиц

студ. специалитета Аллаяров Р.С.

В работе проанализировано локальное перемагничивания тонкопленочных структур $\text{MgO} / \text{CoFeB} / \text{MgO} / \text{Ta} / \text{GaAs}$ и $\text{MgO} / \text{CoFeB} / \text{Ta} / \text{CoFeB} / \text{MgO} / \text{Ta} / \text{GaAs}$ с перпендикулярной магнитной анизотропией под действием дипольного магнитного поля рассеяния наночастиц $\text{Fe} / \text{Fe}_3\text{O}_4$ и их кластеров, нанесенных на поверхность структур, произведено моделирование областей перемагничивания одиночной пленки и двуслойной платформы. Полученные результаты опубликованы в двух статьях, одна из которых находится в печати. Работа выполнена при поддержке гранта 2644.2020.2 Президента Российской Федерации для государственной поддержки ведущих научных школ.

Моделирование конкуренции группы игроков на криптовалютной бирже в терминах теории игр

студ. специалитета Андрюнин А.А.

В работе рассматривается задача моделирования конкуренции двух групп игроков на криптовалютной бирже. По умолчанию, если одна группа игроков побеждает в конкуренции, то вторая группа игроков проигрывает. В модели используется теоретико-игровая концепция для выявления оптимальных стратегий и мотивации игроков к продолжению конкуренции. Показана развернутая форма теоретико-игровой модели, приведены строгие доказательства выбора основной стратегии игроками.

27 мая 2020 года

На семинаре были представлены на рассмотрение три курсовые работы.

Распознавание некоторых свойств графов автоматами

студ. специалитета Демидова А.А.

В работе исследовано применение автоматов для определения того, является граф деревом или нет. Описан автомат, решающий эту задачу для графов, степени вершин которых не превосходят 3. Также описан автомат, перемещающийся по графу с неограниченным ветвлением, и установлено количество красок, необходимое данному автомату для того, чтобы установить, является граф деревом или же содержит в себе циклы.

Исследование задачи предотвращения столкновений

студ. специалитета Горяев А.Ф.

В работе рассматривается задача о столкновении объектов в области. Объекты поступают с двух прямых, которые проходят через эту область. Столкновением считается ситуация, когда два любых объекта оказались в какой-то момент времени одновременно в области. Рассмотрены вопросы, как нужно хранить объекты и как обнаруживать столкновения. Помимо этого, с помощью программы смоделирован процесс, определяющий столкновения. Также промоделирован процесс понижения скорости движения объектов для избежания возникновения столкновений. Рассмотрены примеры изменения скоростей объектов для серии определенных входных данных.

Распознавание ключевых элементов изображения при анализе дорожной ситуации

студ. специалитета Горелова А.А.

В данной работе представлен способ извлечения информации о дорожных знаках и дорожной разметке из изображения с некоторой дорожной

ситуацией. В результате получается схематичное изображение дорожной разметки, а также из изображения выделяются дорожные знаки. Изображения для анализа были взяты из экзаменационных вариантов для сдачи экзамена ПДД. Программа, соответствующая данной работе реализована на языке Python с использованием библиотеки для работы с изображениями OpenCV. Код программы доступен на GitHub <https://github.com/anna-gor/pdd/>.

3 июня 2020 года

На семинаре была рассмотрена курсовая работа студента филиала МГУ им.М.В.Ломоносова в Ташкенте.

Моделирование конкуренции биологических сообществ

студ. бакалавриата Раджабов Р.А.

В данной работе предложена модель мира, существ в нём и их поведение. Рассматривается конкуренция двух популяций без внутривидовой борьбы. В работе также выявляются, изучаются и объясняются факторы, влияющие на конкуренцию и численности популяций.

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 ЛАТ_EX, 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: 15.09.2020

Дата выхода: 25.09.2020

Тираж: 200 экз.

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).