

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

Интеллектуальные Системы.

Теория и приложения

ТОМ **25** ВЫПУСК 1 * **2021**

МОСКВА

УДК 519.95; 007:159.955
ББК 32.81

ISSN 2411-4448

Издается с 1996 г.*

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)

Секретарь 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.

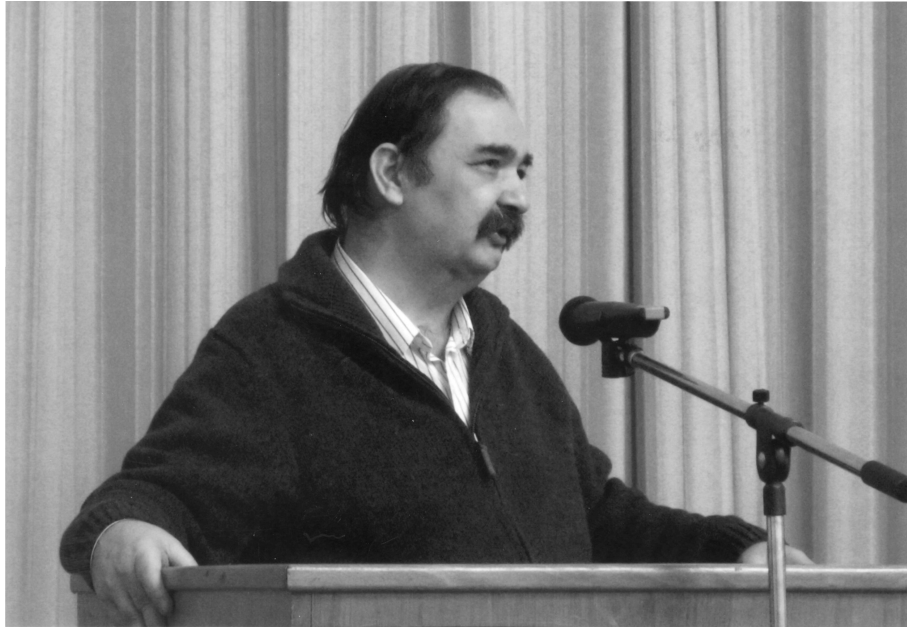
Адрес издателя: 115230, Россия, Москва, Хлебозаводский проезд, д. 7, стр. 9, офис 9. Тел. +7 (495) 939-46-37, e-mail: mail@intsysjournal.org

*) Прежнее название журнала: «Интеллектуальные системы».

© ООО «Интеллектуальные системы», 2021.

ОГЛАВЛЕНИЕ

Памяти Октя Мурадовича Касим-Заде	5
Часть 1. Общие проблемы теории интеллектуальных систем	
<i>Вторушин Ю.И.</i> Алгоритм очевидности для первопорядковой логики предикатов с равенством	21
<i>Чечкин А.В.</i> Тезис о наличии искусственного интеллекта	29
Часть 2. Специальные вопросы теории интеллектуальных систем	
<i>Алешин С.В.</i> Треугольники и уравнения Лагранжа	53
<i>Журавлев А.Д.</i> Возможный подход к задаче прогнозирования спортивных результатов методами анализа данных	63
<i>Соколов А.П., Алисейчик П.А., Моисеев С.В.</i> Распределенный алгоритм поиска траекторий-компаньонов	71
Часть 3. Математические модели	
<i>Валинуров Д.Ю.</i> Локально восстанавливаемые коды на графах	91
<i>Коновалов А.Ю.</i> Λ -выражения для примитивно-рекурсивных функций в иерархии Гжегорчика	107
<i>Кузнецова Е.В.</i> Число состояний универсального автомата бесконечного экрана, реализующего двунаправленное движение на луче	127
<i>Ронжин Д.В.</i> Распознавание A -полноты конечных систем линейных автоматов с добавками над кольцом двоично-рациональных чисел	149



Октай Мурадович Касим-Заде
(29.04.1953 – 22.12.2020)

22 декабря 2020 года ушел из жизни заведующий кафедрой дискретной математики механико-математического факультета МГУ имени М. В. Ломоносова профессор Октай Мурадович Касим-Заде. Механико-математический факультет потерял не только крупнейшего организатора и специалиста в области дискретной математики и математической кибернетики, но и удивительно доброжелательного, отзывчивого и чуткого человека, который тем не менее был принципиальным и твердым, когда это касалось общего дела, интересы которого он всегда ставил выше личных интересов.

Октай Мурад оглы Касим-Заде родился 29 апреля 1953 года в г. Баку, в семье ученого, заслуженного деятеля науки и техники Азербайджана, профессора, инженер-полковника Военно-Морского Флота, Касимзаде Мурада Салман оглы. И, наверное, поэтому интерес к науке и особенно к математике проявился у Октая с детских лет.

В школьные годы он неоднократно участвовал в Республиканских и

Всесоюзных Олимпиадах по математике, отличался широкой эрудицией, тягой к знаниям, организаторскими способностями.

В 1970 году, окончив бакинскую школу №6 с золотой медалью, Октай поступил в МГУ им. М. В. Ломоносова на только что открывшийся и делавший первый набор студентов факультет вычислительной математики и кибернетики. Там произошло судьбоносное событие в жизни Октая Мурадовича — встреча с академиком Олегом Борисовичем Лупановым, который стал для Октая Мурадовича не только научным руководителем, но и образцом математика, педагога и руководителя.

Получив в 1975 году диплом с отличием, Октай Мурадович по распределению начал трудовую деятельность во Всесоюзном научно-исследовательском, проектно-конструкторском и технологическом институте источников тока.

В ноябре 1976 года в порядке перевода Октай Мурадович перешел на работу в Институт прикладной математики АН СССР (ныне ИПМ им. М. В. Келдыша РАН) на должность младшего научного сотрудника отдела теоретической кибернетики, который возглавлял чл.-корр АН СССР С. В. Яблонский и в котором работал О. Б. Лупанов. В Институте прикладной математики Октай Мурадович проработал сначала младшим научным сотрудником, затем научным сотрудником и старшим научным сотрудником до 2006 года (с 1992 г. — по совместительству).

В 1980 году в Совете при ВЦ АН СССР Октаем Мурадовичем защищена кандидатская диссертация на тему «Об одной мере сложности схем из функциональных элементов» (научный руководитель — О. Б. Лупанов).

По инициативе О. Б. Лупанова, начиная с 1986 года, О. М. Касим-Заде ведет педагогическую работу на кафедре дискретной математики механико-математического факультета МГУ имени М. В. Ломоносова сначала как совместитель, а с января 1992 года механико-математический факультет становится его основным местом работы, где он работает сначала старшим научным сотрудником, затем доцентом и профессором кафедры дискретной математики.

В 1996 году Октаем Мурадовичем защищена докторская диссертация на тему «О синтезе некоторых классов управляющих систем, связанных с неявными и параметрическими представлениями булевых функций».

В том же 1996 году О. М. Касим-Заде становится заместителем заведующего кафедрой дискретной математики. В 2005 году ему присваивается ученое звание профессора.

В 2006 году после смерти О. Б. Лупанова Октай Мурадович Касим-Заде возглавляет кафедру дискретной математики и в должности заве-

дующего кафедрой работает до последних дней.

Октай Мурадович был замечательным преподавателем, имел великолепные ораторские способности, легко и непринужденно приковывал к себе внимание аудитории. Он мог на простых и понятных примерах объяснять сложные вопросы. Умел пробудить в слушателях серьезный интерес к рассказываемому материалу: к примеру, несколько раз в разные годы студенты просили Октая Мурадовича продолжать чтение спецкурса по теории графов в каникулы. В своих учениках Октай Мурадович всячески старался развивать нестандартное мышление и умение мыслить самостоятельно, при этом благодаря своему недюжинному кругозору практически всегда мог направить в новом неожиданном направлении, если ученик оказывался в тупиковой ситуации при решении той или иной задачи. Октай Мурадович говорил, что для плодотворной работы необходимо, чтобы руководитель был заинтересован в решении задачи, которая дается ученику, но при этом сам воздерживался от попыток ее решить. Эти принципы он неукоснительно соблюдал сам, никогда не позволяя собственному научному любопытству помешать творчеству учеников. Отдельно стоит отметить, что Октай Мурадович научными достижениями учеников гордился больше чем своими. Под руководством О. М. Касим-Заде защитили кандидатские диссертации шесть учеников.

Октай Мурадович вел большую научно-организационную работу. Он был членом Ученого совета механико-математического факультета МГУ, трех диссертационных советов, входил в состав редколлегии журналов «Дискретная математика» и «Дискретный анализ и исследование операций». Под редакцией О. М. Касим-Заде вышло несколько выпусков «Кибернетического сборника» (совместно с О. Б. Лупановым) и сборника «Математические вопросы кибернетики», а также пять сборников материалов Международных семинаров «Дискретная математика и ее приложения», председателем организационного и программного комитетов которых он был с 2006 года. Под руководством О. М. Касим-Заде проведено несколько школ-семинаров «Синтез и сложность управляющих систем» и молодежных научных школ по дискретной математике и ее приложениям. О. М. Касим-Заде был заместителем председателя оргкомитета нескольких серий всероссийских и международных конференций по дискретной математике и теоретической кибернетике.

Круг научных интересов О. М. Касим-Заде сформировался под сильным влиянием академика О. Б. Лупанова и был необычайно широк. Недаром О. Б. Лупанов еще в 1987 году привлек Октая Мурадовича в качестве соредатора-составителя к издательству «Кибернетического

сборника» — серии сборников переводов лучших работ зарубежных авторов по теоретической кибернетике.

В вопросах публикации собственных научных результатов Октай Мурадович был крайне щепетилен. Он отдавал в печать только те результаты, которые считал действительно принципиальными продвижениями в исследуемых задачах, при этом к своим результатам предъявлял очень серьезные требования. Во многом поэтому у О. М. Касим-Заде по современным меркам не так много публикаций (основных публикаций около 40). К сожалению, значительное число интересных результатов осталось неопубликованными. С другой стороны, практически каждая работа Октай Мурадовича представляет собой крупный вклад в решение соответствующих математических проблем. Все его работы написаны безукоризненно как с математической стороны, так и со стороны умения преподнести материал. Стоит отметить, что Октай Мурадович неоднократно отмечал, что есть, так сказать, три «математических языка»: первый — язык, на котором автор придумывает результаты, второй — язык, на котором следует рассказывать, а третий — язык, предназначенный для изложения результатов в виде научных статей; при этом, вообще говоря, эти три языка очень сильно отличаются друг от друга. Сам Октай Мурадович был непревзойденным мастером слова, как устного, так и письменного.

Среди всех научных работ О. М. Касим-Заде можно выделить три больших цикла. Основные результаты этих циклов относятся к одному из важнейших разделов дискретной математики и математической кибернетики — синтезу и сложности управляющих систем, однако возникающие при исследовании поставленных задач вопросы затрагивают большой спектр проблем из других разделов дискретной математики и математической кибернетики, а зачастую и выходят за рамки дискретной математики и математической кибернетики.

В работах [1–4, 6, 8, 9, 21; 1978–1998 гг] исследуются две меры сложности схем из функциональных элементов, называемые, соответственно, мощностью и активностью, которые тесно связаны с функционированием самих схем. Мощностью (активностью) схемы на входном наборе называется число элементов схемы, выходы которых принимают значение 1 (хотя бы на один вход которых подается значение 1, соответственно). Стандартным образом вводятся функции Шеннона мощности и активности в полном конечном базисе.

О. М. Касим-Заде установил общую картину расположения порядков роста функций Шеннона мощности, отвечающих конечным базисам,

на условной «шкале» порядков роста от n до $2^n/n$. Оказалось, что в зависимости от базиса рост либо не выше квадратичного, либо экспоненциальный, причем, с одной стороны, существует бесконечное число попарно различных порядков экспоненциального роста, а с другой стороны, установлено, что для почти всех конечных базисов рост функции Шеннона мощности линеен.

Октаем Мурадовичем также исследована задача о возможностях построения схем, асимптотически наилучших по сложности и оптимальных по порядку роста по мощности. Оказалось, что в зависимости от базиса ответ может быть как положительный, так и отрицательный.

Для близкой к мощности меры сложности — активности — О. М. Касим-Заде установил внешне схожие результаты, которые, тем не менее, имеют и принципиальные отличия, в частности, экспоненциальных порядков роста функции Шеннона активности конечное число — не более четырех и не менее двух.

Кроме того, несколько работ Октая Мурадовича посвящено вопросам изучения мощности, максимальной и средней, конкретных функций — конъюнкций и дизъюнкций n переменных — в нескольких естественных базисах. В этом направлении получен ряд точных по порядку роста, асимптотически точных, а в некоторых случаях и окончательных результатов.

Второй цикл работ О. М. Касим-Заде, в который, в частности, вошли работы [7, 12, 14–16, 20, 23, 29; 1989–2007 гг], охватывает задачи выразимости и сложности неявных и параметрических представлений булевых функций, а также функций многозначной логики.

Прежде всего, Октай Мурадович решил известную проблему, установив, что два обобщения понятия выразимости посредством суперпозиций — неявная выразимость и параметрическая выразимость — в случае двузначной логики эквивалентны, а следовательно на множестве булевых функций эквивалентны и понятия неявной выразимости и неявной сводимости.

Еще одна важная задача, проблема неявной полноты в k -значной логике, $k \geq 2$, решалась последовательно. Сначала О. М. Касим-Заде решил проблему неявной полноты в случае булевых функций, причем были установлены критерии неявной полноты как в терминах неявно предполных классов, т. е. максимальных по включению не полных неявно систем функций, так и в терминах минимальных по включению неявно полных явно замкнутых классов; затем его ученица Е. А. Орехова получила критерий неявной полноты в трехзначной логике в терминах минималь-

ных по включению неявно полных явно замкнутых классов, и, наконец, сам Октай Мурадович для произвольного $k \geq 2$ предложил эффективный критерий неявной полноты систем функций k -значной логики, основанный на проверке свойств функций, входящих в явное замыкание. М. В. Старостин, еще один ученик Октая Мурадовича, нашел ряд ранее неизвестных неявно предполных классов трехзначной логики и сформулировал критерий неявной полноты в трехзначной логике в терминах неявно предполных классов.

После решения вопросов выразимости, как это обычно и бывает, на первый план выходят проблемы представлений, в том или ином смысле наилучших. Для задач неявных и параметрических представлений булевых функций над заданным базисом Октаем Мурадовичем рассматривались две естественные меры сложности: 1) ранг, определяемый как число уравнений в системе, задающей функцию; 2) собственно сложность, определяемая как число функциональных символов, соответствующих функциям из базиса, во всех уравнениях системы.

Для соответствующих функций Шеннона, характеризующих ранги булевых функций от n переменных над базисом B , и называемых ранговой функцией для случая неявного представления и P -ранговой функцией для случая параметрического представления, О. М. Касим-Заде доказал, что множество всех ранговых и P -ранговых функций разбивается на шесть типов в зависимости от того, каким классом Поста является замыкание по суперпозиции множества B , а также установил точные значения ранговых и P -ранговых функций для пяти типов базисов B и порядок роста ранговых и P -ранговых функций для шестого типа.

Ряд интересных результатов о поведении ранговых функций в случае неявного представления функций многозначной логики получила Е. В. Михайлец, ученица Октая Мурадовича.

О. М. Касим-Заде провел глубокое исследование сложности параметрического представления булевых функций. Выяснилось, что по своей выразительной силе реализация булевых функций параметрическими представлениями близка к реализации булевых функций схемами из функциональных элементов. Октай Мурадович сначала установил асимптотику роста функции Шеннона сложности параметрического представления для произвольного параметрически полного конечного базиса, состоящего из булевых функций с произвольными положительными весами, а спустя несколько лет нашел асимптотику функции Шеннона и для любого конечного базиса. При этом вид нелинейных асимптотик определяется мощностной нижней оценкой и этот вид очень похож на вид

асимптотики функции Шеннона сложности реализации булевых функций схемами из функциональных элементов в конечных базисах, состоящих из булевых функций с произвольными положительными весами; единственное отличие заключается в том, что вместо приведенного веса базиса в асимптотике фигурирует так называемый редуцированный вес базиса, при определении которого в случае нелинейной функции исходный вес функции делится на число существенных переменных, а не на число существенных переменных без единицы.

Стоит отметить, что О. М. Касим-Заде удалось основные результаты о сложности параметрических представлений в параметрически полных базисах перенести на задачи о сложности сетей из функциональных элементов. Кроме того, применение основных идей асимптотически оптимального метода синтеза параметрических представлений позволило понизить верхнюю оценку и установить асимптотику сложности для одной математической модели электронных схем на дополняющих МОП-транзисторах (КМОП-схем).

В третьем цикле работ [10, 11, 18, 25, 27, 29, 31–33; 1994–2013 гг] О. М. Касим-Заде исследовал проблемы сложности (в работах этого цикла под сложностью понимается число элементов в схеме) и глубины булевых функций в бесконечных базисах.

В рамках сложностной составляющей этого цикла необходимо отметить следующие результаты.

Во-первых, Октай Мурадович, доказал, что существует такая абсолютная константа c , что для всякого функционально полного бесконечного базиса поведение функции Шеннона сложности булевых функций в этом базисе не превосходит величины $c2^{n/2}$.

Во-вторых, был разработан метод получения верхних оценок сложности в произвольном бесконечном полном базисе, позволяющий получать верхние оценки функции Шеннона, сопоставимые с мощностными нижними оценками. Последняя модификация этого метода получения двусторонних оценок сложности дает возможность устанавливать нижнюю и верхнюю оценки функции Шеннона, отличающиеся на множитель порядка n в случае, если порядок роста функции Шеннона не менее линейного.

В-третьих, О. М. Касим-Заде показал, что порядок роста функции Шеннона сложности в полных бесконечных базисах либо равен 1, либо лежит в одном из двух интервалов: или между функциями $\log_2 n$ и n , или между функциями n и $2^{n/2}$, а также что любая функция одного действительного переменного, выражимая в виде суперпозиции рацио-

нальных функций с действительными коэффициентами, логарифмов и экспонент и имеющая порядок роста не ниже n и не выше $2^{O(\sqrt{n})}$, является порядком роста функции Шеннона сложности схем над некоторым бесконечным базисом (но порядки роста не исчерпываются этими функциями — например, порядком роста функции Шеннона может быть функция $n \log_2^* n$, где $\log_2^* n$ — двоичный сверхлогарифм числа n).

Кроме того, О. М. Касим-Заде исследовал задачу о сложности реализации булевых функций в бесконечном базисе антицепных функций, состоящем их характеристических функций антицепей и констант 0 и 1; на необходимость изучения теоретико-сложностных свойств этого базиса обращал внимание О. Б. Лупанов. Для сложности линейной функции в этом базисе установлена нижняя оценка вида $\Omega(\sqrt{n/\ln n})$, экспоненциальная по отношению к мощностной оценке и, как следствие, получены в некотором смысле близкие нижняя и верхняя оценки функции Шеннона. Точное значение сложности линейной функции и порядок роста функции Шеннона в базисе антицепных функций установила ученица Октай Мурадовича О. В. Подольская. Она же показала, что в базисе, состоящем из всех антицепных функций и линейных функций от любого числа переменных, порядок роста функции Шеннона равен $\sqrt{n \log n}$, тем самым ответив на вопрос, существует ли базис, для которого порядок роста функции Шеннона лежит строго в интервале между функциями $\log n$ и n .

В задачах реализации булевых функций в бесконечных базисах О. М. Касим-Заде исследовал не только вопросы сложности, но и глубины.

Октай Мурадович установил, что для каждого бесконечного базиса B булевых функций либо существует единственное простое p , для которого степени полиномов с идемпотентными переменными над полем вычетов по модулю p для функций из B имеют максимальное значение, обозначаемое $\deg_p B$, либо такого простого p не существует. Для произвольной отличной от констант и переменных функции f ее глубина $D_B(f)$ над базисом B во втором случае ограничена числами 1 и 6, а в первом удовлетворяет неравенствам $\lceil \log_\gamma \deg_p f \rceil \leq D_B(f) \leq \lfloor \log_\gamma \deg_p f \rfloor + 5$, где $\gamma = \deg_p B$, а $\deg_p f$ — степень полинома с идемпотентными переменными над полем вычетов по модулю p , представляющего функцию f . Как следствие для функции Шеннона глубины $D_B(n)$ во втором случае найдется константа β , $1 \leq \beta \leq 6$, такая, что при всех достаточно больших n выполняется равенство $D_B(n) = \beta$, а в первом случае при всех n справедливы неравенства $\lceil \log_\gamma n \rceil \leq D_B(n) \leq \lfloor \log_\gamma n \rfloor + 5$.

Ученик Октая Мурадовича А. В. Кочергин исследовал поведение функции Шеннона глубины для случая конечных и бесконечных базисов, состоящих из функций k -значной логики, где $k \geq 3$, и ему удалось полностью описать качественную картину асимптотического поведения функции Шеннона глубины в этом случае.

Следует отметить еще несколько полученных Октаем Мурадовичем результатов, относящихся к сложности управляющих систем.

Основываясь на идеях получения уже упоминавшейся асимптотически точной верхней оценки сложности КМОП-схем, О. М. Касим-Заде установил [22] асимптотику функции Шеннона сложности реализации функций k -значной логики в классе алгоритмов, являющихся обобщениями на k -значный случай некоторых видов бинарных программ.

В работе [5] Октаем Мурадовичем для известной задачи о монотонной сложности многочленов (минимальном числе операций умножения на положительные числа, сложения и умножения, достаточном для вычисления заданного многочлена от этих переменных с неотрицательными коэффициентами) в случае реализации многочленов с коэффициентами из множества $\{0, 1\}$ и имеющими по каждой переменной степень 1 установлен следующий результат. Сложность многочлена от n переменных, содержащего $2^{n/2} - 1$ слагаемое и задаваемого проверочной матрицей исправляющего две ошибки двоичного кода Боуза—Чоудхури—Хоквингема, имеющей размер $n \times (2^{n/2} - 1)$, имеет порядок роста $2^{n/2}$, при этом минимально возможное число операций сложения, достаточное для его вычисления, в точности равно $2^{n/2} - 2$.

Ряд интересных результатов О. М. Касим-Заде получил (см., например, [17, 24, 34]) также в задаче о построении идеализированных электрических схем из единичных сопротивлений наименьшей сложности и в задаче о разбиении прямоугольника на квадраты.

Ряд публикаций Октая Мурадовича нельзя отнести к проблематике теории сложности, однако зачастую, сама постановка исследуемых в них вопросов возникала при решении различных задач синтеза и сложности управляющих систем.

Так, например, рассматриваемая в [19] комбинаторная задача покрытия n -мерного булева куба такими антицепями, что в наборах каждой антицепи (кроме состоящей только из нулевого набора) есть общая единичная компонента, возникла при изучении ранга неявных представлений булевых функций над замкнутыми классами монотонных функций, не меньших какой-либо своей переменной. О. М. Касим-Заде нашел точное значение минимально возможного числа таких антицепей, покры-

вающих n -мерный булев куб, и для всех n явно построил минимальное покрытие.

В работах [13, 28] изучались два вида обобщенных инвариантных классов С. В. Яблонского — классы, инвариантные относительно подстановок произвольных одноместных функций, и классы, инвариантные относительно подстановок монотонных одноместных функций. В обоих случаях Октаем Мурадовичем установлены верхние оценки числа функций от фиксированного числа переменных в таких классах, позволившие доказать, что в обоих случаях любой ненулевой класс (класс с ненулевым параметром) совпадает с P_2 .

Отдельно стоит сказать про написанное О. М. Касим-Заде учебное пособие «Одна экстремальная задача комбинаторного анализа» [26], в котором рассматривается ряд вопросов, начиная с простейших фактов и заканчивая нерешенными проблемами, касающихся одной из важнейших экстремальных задач комбинаторного анализа и теории графов, известной как проблема Заранкевича. В общем случае проблема Заранкевича заключается в нахождении наибольшего натурального числа $N = N_{a,b}(m, n)$, такого, что существует булева матрица размера $m \times n$ с N единицами, не содержащая единичных подматриц размера $a \times b$, а также в построении таких булевых матриц.

В этом пособии ярко проявилось умение Октая Мурадовича постепенно, начиная с простых и понятных примеров, подводить читателя к все более и более серьезным утверждениям, выделять узловые моменты и подробно их объяснять, устанавливая общую сущность казалось бы совсем разных объектов, да и просто доступно, красиво и при этом предельно точно изъясняться.

К сожалению, многие результаты своих исследований Октай Мурадович не успел опубликовать...

Широта научных интересов Октая Мурадовича также нашла свое отражение в разнообразии тематик работ его учеников. Даже по опубликованным работам (а доведено до диссертационных работ или опубликовано было тоже, к сожалению, не все) видно, что результаты учеников относятся к самым разным разделам дискретной математики, начиная с вероятностного направления в дискретной математике и применения теории формальных языков и грамматик (А. Д. Яшунский и А. В. Кочергин) и заканчивая изучением свойств ветвящихся цепных дробей (Н. Р. Закиров), тестовыми задачами на графах (Е. В. Дебрев) и вопросами сложности реализации функций действительного переменного сетями из интеграторов и сумматоров (А. И. Проскураков). Отрадно, что значительная часть

учеников Октая Мурадовича продолжает вести активную научную работу.

Октай Мурадович был высокоэрудированным ученым, общительным, доброжелательным и отзывчивым человеком. Светлая память об Октае Мурадовиче Касим-Заде сохранится в сердцах родных, близких, коллег и друзей.

*С. Б. Гашков, О. С. Дудакова, М. Д. Ковалев,
Р. М. Колпаков, В. В. Кочергин, Н. П. Редькин,
А. В. Чашкин, В. Н. Чубариков, А. Д. Яшунский*

Избранные публикации О. М. Касим-Заде

1. Об одновременной минимизации сложности и мощности схем из функциональных элементов // Проблемы кибернетики. Вып. 33. — М.: Наука, 1978. — С. 215–220.
2. Об одной мере сложности схем из функциональных элементов // ДАН СССР. — 1980. — Т. 250, № 4. — С. 797–800.
3. Об одной мере сложности схем из функциональных элементов // Проблемы кибернетики. Вып. 38. — М.: Наука, 1981. — С. 117–179.
4. О мощности схем в базисах некоторых типов // Дискретная математика и математическая кибернетика. Серия «Вопросы кибернетики». — Т. 86. — М.: ВЦ АН СССР, 1982. — С. 77–93.
5. О сложности монотонных многочленов // Материалы Всесоюзного семинара по дискретной математике и ее приложениям. — М.: Изд-во МГУ, 1986. — С. 136–138.
6. О влиянии базиса на активность схем из функциональных элементов // Труды Всесоюзного семинара по дискретной математике и ее приложениям. — М.: Изд-во МГУ, 1989. — С. 143–145.
7. О сложности реализации булевых функций в одной модели электронных схем // Математические вопросы кибернетики. Вып. 2. — М.: Наука, 1989. — С. 145–160.

8. Об одной мере активности схем из функциональных элементов // Математические вопросы кибернетики. Вып. 4. — М.: Наука, 1992. — С. 218–228.
9. Kassim-Zade O. M. On some complexity measures for Boolean functions // Complexity and realization of Boolean functions. —Dagstuhl-Seminar-Report № 45, 24.8–28.8.92. — 1992.
10. О сложности схем в одном бесконечном базисе // Вестник Московского университета. Серия 1. Математика. Механика. — 1994. — № 6. — С. 40–44.
11. О сложности реализации булевых функций схемами в одном бесконечном базисе // Дискретный анализ и исследование операций. — 1995. — Т. 2, № 1. — С. 7–20.
12. О неявной выразимости булевых функций // Вестник Московского университета. Серия 1. Математика. Механика. — 1995. — № 2. — С. 44–49.
13. О классах булевых функций, инвариантных относительно подстановки функций от одной переменной // Вестник Московского университета. Серия 1. Математика. Механика. — 1995. — № 3. — С. 79–82.
14. О синтезе сетей из функциональных элементов // Доклады Академии наук. — 1996. — Т. 348, № 2. — С. 159–161.
15. О неявной выразимости в двузначной логике и криптоизоморфизмах двухэлементных алгебр // Доклады Академии наук. — 1996. — Т. 348, № 3. — С. 299–301.
16. Об одной метрической характеристике неявных и параметрических представлений булевых функций // Математические вопросы кибернетики. Вып. 6. — М.: Наука, 1996. — С. 133–188.
17. О сложности схем из единичных сопротивлений и о некоторых свойствах чисел Фибоначчи // Аналитическая теория чисел и приложения. Труды Математического Института им. В. А. Стеклова. — 1997. — Т. 218. — С. 233–247.
18. Общая верхняя оценка сложности схем в произвольном бесконечном полном базисе // Вестник Московского университета. Серия 1. Математика. Механика. — 1997. — № 4. — С. 59–61.
19. О минимальных покрытиях булева куба центрированными антицепя-

- ми // Дискретный анализ и исследование операций. Серия 1. — 1997. — Т. 4, № 3. — С. 9–17.
20. О сложности параметрических представлений булевых функций // Математические вопросы кибернетики. Вып. 7. — М.: Наука, 1998. — С. 85–160.
21. О мощности индивидуальных функций // Труды семинара по дискретной математике и ее приложениям. — М.: Изд-во мех.-мат. ф-та МГУ, 1998. — С. 63–65.
22. О сложности реализации функций в одном классе алгоритмов // Материалы IX Межгосударственной школы-семинара «Синтез и сложность управляющих систем». — М.: Изд-во мех.-мат. ф-та МГУ, 1999. — С. 25–30.
23. О поведении функций Шеннона сложности параметрических представлений булевых функций // Вестник Московского университета. Серия 1. Математика. Механика. — 2001. — № 3. — С. 66–68.
24. О минимальных разбиениях плоских прямоугольников на квадраты // Материалы VII Международного семинара «Дискретная математика и ее приложения». — М.: Изд-во ЦПИ при мех.-мат. ф-те МГУ, 2001. — С. 263–265.
25. Об одном методе получения оценок сложности схем над бесконечными базисами // Математические вопросы кибернетики. Вып. 11. — М.: Наука, 2002. — С. 247–254.
26. Одна экстремальная задача комбинаторного анализа. — М.: Изд-во мех.-мат. ф-та МГУ, 2004. — ISBN 978-5-458-26668-0. — 88 с.
27. Об одном методе получения оценок сложности схем над произвольным бесконечным базисом // Дискретный анализ и исследование операций. Серия 1. — 2004. — Т. 11, № 2. — С. 41–65.
28. О метрических свойствах обобщенных инвариантных классов // Математические вопросы кибернетики. Вып. 15. — М.: Наука, 2006. — С. 9–34.
29. О глубине булевых функций при реализации схемами над произвольным базисом // Вестник Московского университета. Серия 1. Математика. Механика. — 2007. — № 1. — С. 18–21.
30. О неявной полноте в k -значной логике // Вестник Московского уни-

верситета. Серия 1. Математика. Механика. — 2007. — № 3. — С. 10–14.

31. О глубине булевых функций над произвольным бесконечным базисом // Дискретный анализ и исследование операций. Серия 1. — 2007. — Т. 14, № 1. — С. 45–69.

32. О глубине булевых функций при реализации схемами над произвольным бесконечным базисом // Вестник Московского университета. Серия 1. Математика. Механика. — 2012. — № 6. — С. 55–57.

33. О порядках роста функций Шеннона сложности схем над бесконечными базисами // Вестник Московского университета. Серия 1. Математика. Механика. — 2013. — № 3 — С. 55–57.

34. О точных значениях сложности чисел при реализации схемами из единичных сопротивлений // Материалы XII Международного семинара «Дискретная математика и ее приложения» имени академика О.Б. Лупанова. — М., 2016. — С. 134–137.

Часть 1.
Общие проблемы теории
интеллектуальных систем

Алгоритм очевидности для первопорядковой логики предикатов с равенством

Вторушин Ю.И.¹

Статья является продолжением статьи [1]. Рассматривается алгоритм верификации формализованных математических доказательств для логики предикатов первого порядка с равенством. Доказываются теоремы о его корректности и полноте.

Ключевые слова: автоматическое доказательство теорем, система автоматизации дедукции, верификация доказательств, язык первого порядка, исчисление предикатов, продукционная система, искусственный интеллект.

В статье [1] был описан алгоритм очевидности для логики предикатов без равенства, который эффективно решает задачу верификации формализованных математических текстов. Данная статья является продолжением статьи [1]. В настоящей работе рассматривается алгоритм очевидности для логики предикатов с равенством. При этом используются принятые в [1, 2, 3] понятия и обозначения.

Кратко напомним обозначения. В работе [1] для $i = 1, 2, 3$ рассмотрены системы натуральной дедукции \mathcal{D}_i и алгоритмы очевидности \mathcal{E}_i , которые формулируются с помощью продукционных систем \mathcal{V}_i . При этом случай $i = 1$ соответствует случаю классической логики, $i = 2$ — интуиционистской, а $i = 3$ — минимальной. В данной статье для каждого $i = 1, 2, 3$ формулируется алгоритм очевидности \mathfrak{E}_i и продукционная система \mathfrak{V}_i , которые соответственно включают \mathcal{E}_i и \mathcal{V}_i как часть. Все положительные свойства алгоритма \mathcal{E}_i наследуются алгоритмом \mathfrak{E}_i .

Отметим, что системы натуральной дедукции \mathcal{D}_i рассмотрены в статье [2]. Исчисление \mathcal{D}_i получается из \mathcal{D}_i в результате добавления аксиомы $t = t$ и правила вывода

$$\frac{t = s; \quad \Phi(t)}{\Phi(t/s)},$$

¹Вторушин Юрий Игоревич — ведущий программист компании DSS Lab, e-mail: urchick@mail.ru.

Vtorushin Yuriy Igorevich — leader programmer, DSS Lab.

где через $\Phi(t//s)$ обозначена формула, которая может быть получена из формулы Φ в результате замены некоторых вхождений (быть может, всех) терма t на терм s .

Рассматриваемый алгоритм очевидности \mathfrak{E}_i решает задачу проверки правильности формальных доказательств, записанных в стиле обычных математических статей, и может быть реализован в виде эффективной процедуры. Для его обоснования в этой статье формулируются те же теоремы, что и в статье [1], только сейчас речь идет не о \mathcal{D}_i , \mathcal{V}_i и \mathcal{E}_i , а о \mathfrak{D}_i , \mathfrak{V}_i и \mathfrak{E}_i .

Пусть псевдоформула Φ обладает свойством чистоты переменных [3]. Пусть r_1, r_2, \dots, r_k есть список некоторых попарно непересекающихся вхождений псевдотермов в Φ , которые не содержат связанных переменных. Определим операцию *замены* псевдотермов r_1, \dots, r_k на псевдотерм t . Результат ее будет обозначаться символом

$$\left(\Phi \frac{r_1, r_2, \dots, r_k}{t} \right). \quad (1)$$

Рассмотрим псевдоформулу Ψ такую, что $\Phi = \Psi \begin{pmatrix} x_1 & \dots & x_k \\ r_1 & \dots & r_k \end{pmatrix}$, где каждая свободная предметная переменная x_i входит в Ψ ровно один раз. Тогда (1) есть по определению псевдоформула $\Psi \begin{pmatrix} x_1 & \dots & x_k \\ t & \dots & t \end{pmatrix}$.

Продукционная система \mathfrak{V}_i получается из системы \mathcal{V}_i в результате добавления новых примитивных задач и новых правил декомпозиции. Кроме примитивных задач продукционной системы \mathcal{V}_i система \mathfrak{V}_i включает также примитивные задачи следующего вида:

$$\Gamma \triangleright (t = t).$$

По определению будем считать, что задачи вида $\Gamma \triangleright (t_1 = t_2)$ примитивизируются подстановками θ вида $\mathbf{MGU}(t_1, t_2)$, если псевдотермы t_1 и t_2 унифицируемы.

Кроме семи правил декомпозиции задач системы \mathcal{V}_i продукционная система \mathfrak{V}_i включает следующие два правила декомпозиции.

8. Пусть s_1, \dots, s_n — список всех тех псевдотермов псевдоформулы Ψ , которые унифицируются с псевдотермом s . Пусть $\sigma_i = \mathbf{MGU}(s, s_i)$, вхождения в Ψ псевдотермов s_{i_1}, \dots, s_{i_k} попарно не пересекаются и $\sigma_{i_1 \dots i_k}$ есть комбинация подстановок $\sigma_{i_1}, \dots, \sigma_{i_k}$. Тогда если разрешима задача $\Gamma \triangleright \left(\Psi \frac{s_{i_1}, \dots, s_{i_k}}{t} \right) \sigma_{i_1 \dots i_k}$ и θ — ее допустимая подстановка, то задача $\langle \alpha, t = s \rangle \Gamma \triangleright \Psi$ также считается разрешимой и комбинация подстановок

θ и $\sigma_{i_1 \dots i_k}$ считается ее допустимой подстановкой:

$$\frac{\Gamma \triangleright \left(\Psi \frac{s_{i_1}, s_{i_2}, \dots, s_{i_k}}{t} \right) \sigma_{i_1 i_2 \dots i_k}}{\langle \alpha, t = s \rangle \Gamma \triangleright \Psi}.$$

9. Учитывая, что предикат равенства обладает свойством симметричности, мы получаем еще одно правило:

$$\frac{\Gamma \triangleright \left(\Psi \frac{s_{i_1}, s_{i_2}, \dots, s_{i_k}}{t} \right) \sigma_{i_1 i_2 \dots i_k}}{\langle \alpha, s = t \rangle \Gamma \triangleright \Psi}.$$

Формулировка продукционных систем \mathfrak{V}_i завершена. Каждая продукционная система \mathfrak{V}_i задает алгоритм поиска решения дедуктивной задачи без уточнения стратегии построения дерева поиска типа И/ИЛИ. Зафиксировав конкретную стратегию, мы получим некоторую реализацию так сформулированного алгоритма. Напомним, что стратегия построения дерева поиска заключается в задании способа выбора листа для формирования у него, во-первых, множества всех примитивизирующих этот лист подстановок, во-вторых, всех дочерних связей вершин, соответствующих правилам декомпозиции. Затем каждая примитивизирующая подстановка “поднимается” по соединяющей лист с корнем ветви с помощью операции комбинации [3]. Тем самым в ходе процесса построения дерева поиска у его вершин формируются наборы допустимых подстановок. В качестве символических названий сформулированных правил декомпозиции 8 и 9 будем соответственно использовать обозначения $(LR = \triangleright)$ и $(RL = \triangleright)$.

Теорема. *Алгоритм проверки непосредственных обоснований всегда завершает свою работу.*

Доказательство. Обоснование этого утверждения для \mathfrak{V}_i почти не отличается от доказательства [1] такой же теоремы, сформулированной для \mathcal{V}_i . Пусть n есть общее число вхождений всех подформул в исходную задачу

$$\langle \alpha_1, \Phi_1 \rangle \dots \langle \alpha_m, \Phi_m \rangle \triangleright \Psi. \quad (2)$$

Чтобы установить применимость алгоритма к любой задаче (2) покажем, что дерево поиска конечно.

Действительно, применение правил декомпозиции $(\triangleright \&)$, $(\triangleright \vee)$, $(\triangleright \exists)$, $(\vee \triangleright)$, $(\& \triangleright)$, $(\supset \triangleright)$, $(\forall \triangleright)$, $(LR = \triangleright)$, $(RL = \triangleright)$ всегда приводит к тому, что значение параметра n у каждой дочерней задачи становится меньше значения этого параметра у родительской задачи. Отсюда следует, что высота дерева поиска задачи (2) не превосходит n . Таким образом, поисковое дерево конечно. \square

Изменив в алгоритме очевидности \mathcal{E}_i обработку непосредственных обоснований в соответствии с продукционной системой \mathfrak{Y}_i , мы получаем алгоритм очевидности \mathfrak{E}_i для первопорядковой логики предикатов с равенством. При этом новая процедура \mathfrak{E}_i наследует все положительные свойства процедуры \mathcal{E}_i .

Очевидным следствием теоремы о конечности поисковых деревьев у продукционной системы \mathfrak{Y}_i и формулировки алгоритма очевидности \mathfrak{E}_i является следующее утверждение.

Следствие. *Алгоритм очевидности \mathfrak{E}_i всегда завершает свою работу.*

В качестве примера рассмотрим формальное доказательство теоремы Пифагора “если $\vec{a} \cdot \vec{b} = 0$, то $(\vec{a} + \vec{b})^2 = \vec{a}^2 + \vec{b}^2$ ”. Такое доказательство в [2] раздроблено до уровня применений правил вывода системы натуральной дедукции \mathfrak{D}_i . По сравнению с [2] алгоритм очевидности \mathfrak{E}_i позволяет записать это доказательство в стиле обычной математической статьи:

```

environ
  type Real;
  func s(Real, Real) -> Real;
  func o() -> Real;
  reserve x,y for Real;
  1: for x holds s(x,o()) = x;
  2: for x,y holds s(x,y) = s(y,x);

  type Vector;
  func f(Vector, Vector) -> Vector;
  func h(Vector,Vector) -> Real;
  reserve u,v,w for Vector;
  3: for u,v,w holds h(f(u,v),w) = s(h(u,w),h(v,w));
  4: for u,v,w holds h(u,f(v,w)) = s(h(u,v),h(u,w));
  5: for u,v holds h(u,v) = h(v,u);
text
for a,b being Vector st h(a,b) = o()
  holds h(f(a,b), f(a,b)) = s(h(a,a), h(b,b))
proof
  let a be Vector, b be Vector;
  assume 6: h(a,b) = o();
  7: s(o(), h(b,b)) = s(h(b,b), o()) by 2
      . = h(b,b) by 1;
  thus h(f(a,b), f(a,b)) = s(h(a, f(a,b)), h(b, f(a,b))) by 3
      . = s(s(h(a,a), h(a,b)), h(b, f(a,b))) by 4
      . = s(s(h(a,a), o()), h(b, f(a,b))) by 6
      . = s(h(a,a), h(b, f(a,b))) by 1
      . = s(h(a,a), s(h(b,a), h(b,b))) by 4
      . = s(h(a,a), s(o(), h(b,b))) by 6,5
      . = s(h(a,a), h(b,b)) by 7;
end;

```


В этом доказательстве рассмотрим более подробно непосредственное обоснование

$$s(h(a, a), s(h(b, a), h(b, b))) = s(h(a, a), s(o(), h(b, b))) \text{ by 6,5.}$$

Для проверки этого утверждения алгоритм очевидности \mathfrak{E}_i сформирует дерево поиска типа И/ИЛИ, в котором содержится соответствующее допустимой подстановке $\theta = \begin{pmatrix} U & V \\ a & b \end{pmatrix}$ следующее имеющее вид только одной ветви деривационное поддерево:

$$\begin{array}{c} \text{Yes} \\ \uparrow \lambda \\ \triangleright s(h(a, a), s(h(a, b), h(b, b))) = s(h(a, a), s(h(a, b), h(b, b))) \\ \uparrow \nu \\ \langle 6, h(a, b) = o() \rangle \triangleright \left(\begin{array}{l} s(h(a, a), s(h(a, b), h(b, b))) \\ = s(h(a, a), s(o(), h(b, b))) \end{array} \right) \\ \uparrow \mu \\ \left. \begin{array}{l} \langle 5, h(U, V) = h(V, U) \rangle \\ \langle 6, h(a, b) = o() \rangle \end{array} \right\} \triangleright \left(\begin{array}{l} s(h(a, a), s(h(b, a), h(b, b))) \\ = s(h(a, a), s(o(), h(b, b))) \end{array} \right) \\ \uparrow \\ \left. \begin{array}{l} \langle 5, \forall v (h(U, v) = h(v, U)) \rangle \\ \langle 6, h(a, b) = o() \rangle \end{array} \right\} \triangleright \left(\begin{array}{l} s(h(a, a), s(h(b, a), h(b, b))) \\ = s(h(a, a), s(o(), h(b, b))) \end{array} \right) \\ \uparrow \\ \left. \begin{array}{l} \langle 5, \forall u \forall v (h(u, v) = h(v, u)) \rangle \\ \langle 6, h(a, b) = o() \rangle \end{array} \right\} \triangleright \left(\begin{array}{l} s(h(a, a), s(h(b, a), h(b, b))) \\ = s(h(a, a), s(o(), h(b, b))) \end{array} \right) \end{array}$$

Лист этого дерева примитивизируется подстановкой $\lambda = \varepsilon$ и получается из корня дерева поиска в результате следующей цепочки применений правил декомпозиции продукционной системы \mathfrak{V}_i : $(\forall \triangleright)$, $(\forall \triangleright)$, $(LR = \triangleright)$, $(LR = \triangleright)$. Комбинация подстановок $\lambda, \nu = \varepsilon$ и $\mu = \begin{pmatrix} U & V \\ a & b \end{pmatrix}$ дает допустимую подстановку θ для корня дерева поиска.

Обоснование теоремы о корректности для \mathfrak{V}_i получается из доказательства такой же теоремы в [1], сформулированной для \mathcal{V}_i . В связи с этим имеет смысл рассмотреть только примитивные задачи и правила декомпозиции, которых нет в \mathfrak{V}_i .

Теорема о корректности. *Если задача $\langle \alpha_1, \Phi_1 \rangle \dots \langle \alpha_m, \Phi_m \rangle \triangleright \Psi$ разрешима в \mathfrak{V}_i , то существует цепочка непосредственных обоснований, которая устанавливает выводимость формулы Ψ из формул Φ_1, \dots, Φ_m в \mathfrak{D}_i .*

Доказательство. Пусть θ есть подстановка из набора допустимых подстановок задачи $P = (\langle \alpha_1, \Phi_1 \rangle \dots \langle \alpha_m, \Phi_m \rangle \triangleright \Psi)$. Мы предполагаем, что

подстановка θ является основной — в противном случае мы можем заметить “нерешенные” метапеременные локальными константами. Пусть Θ является соответствующим этой подстановке θ деривационным деревом. Будем обозначать штрихом следующее соответствие: $\Phi' = \Phi\theta$. Индукцией по высоте дерева Θ покажем, как в рамках \mathfrak{D}_i построить цепочку непосредственных обоснований, которая устанавливает выводимость формулы Ψ' из формул Φ'_1, \dots, Φ'_m в \mathfrak{D}_i . В рамках \mathfrak{D}_i рассмотрим только те варианты порождения дерева Θ , которых нет в [1].

Пусть P примитивизируется подстановкой θ . Тогда $\Psi = (t_1 = t_2)$, $\Psi' = (t'_1 = t'_2)$ и $t'_1 = t'_2$. В этом случае считается, что в \mathfrak{D}_i формула Ψ' является непосредственным следствием формул Φ'_1, \dots, Φ'_m .

Пусть $P = (\langle \alpha_i, t = s \rangle \Pi \triangleright \Psi)$ получается по правилу ($LR = \triangleright$), а дочерняя задача $\Pi \triangleright \Upsilon\sigma_{i_1\dots i_k}$ является корнем поддерева Θ_1 , где $\Upsilon = \left(\Psi \frac{s_{i_1}, \dots, s_{i_k}}{t} \right)$. Заметим, что $\Upsilon' = \Upsilon\theta = (\Upsilon\sigma_{i_1\dots i_k})\theta$, $\Upsilon = \Upsilon(t)$, $\Upsilon' = (\Upsilon(t))' = \Upsilon'(t')$ и Ψ' имеет вид $\Upsilon'(t'//s')$. Пусть по индуктивному предположению деривационному поддереву Θ_1 , соответствующему задаче $\Pi \triangleright \Upsilon\sigma_{i_1\dots i_k}$, сопоставляется цепочка непосредственных обоснований

$$\Delta(\Upsilon' \text{ by } \gamma_1, \dots, \gamma_n).$$

Указанная цепочка устанавливает выводимость Υ' из Φ'_1, \dots, Φ'_m в \mathfrak{D}_i . Тогда дереву Θ мы можем сопоставить цепочку непосредственных обоснований

$$(\alpha_i : t' = s') \Delta(\beta : \Upsilon' \text{ by } \gamma_1, \dots, \gamma_n) (\Upsilon'(t'//s') \text{ by } \alpha_i, \beta).$$

Полученная цепочка устанавливает выводимость Ψ' из Φ'_1, \dots, Φ'_m в \mathfrak{D}_i .

Аналогично рассматривается случай ($RL = \triangleright$). Если P имеет вид $(\langle \alpha_i, s = t \rangle \Pi \triangleright \Psi)$, то для обоснования выводимости Ψ' из Φ'_1, \dots, Φ'_m требуется более длинная цепочка непосредственных умозаключений:

$$\begin{aligned} &(\alpha_i : s' = t') (\gamma : s' = s') \\ &(\delta : t' = s' \text{ by } \alpha_i, \gamma) \Delta(\beta : \Upsilon' \text{ by } \gamma_1, \dots, \gamma_n) (\Upsilon'(t'//s') \text{ by } \delta, \beta). \end{aligned}$$

□

Корректность алгоритма очевидности \mathfrak{E}_i является следствием корректности \mathfrak{A}_i . Это нетрудно обосновать с помощью тех же рассуждений, что и в [1], в связи с чем мы доказательство опускаем.

Следствие. Если алгоритм очевидности \mathfrak{E}_i распознал, что “ Δ есть обоснование формулы Ψ с посылками Φ_1, \dots, Φ_m ”, то существует натуральный вывод Δ' в \mathfrak{D}_i формулы Ψ из посылок Φ_1, \dots, Φ_m .

Полнота \mathfrak{A}_i устанавливается также как и в [1].

Теорема о полноте. Если формула Ψ является непосредственным следствием формул Φ_1, \dots, Φ_m в системе натуральной дедукции \mathfrak{D}_i , то задача $\langle \alpha_1, \Phi_1 \rangle \dots \langle \alpha_m, \Phi_m \rangle \triangleright \Psi$ разрешима в \mathfrak{A}_i .

Доказательство. Пусть формула Ψ получает обоснование с помощью одного из прямых правил вывода из формул Φ_1, \dots, Φ_m :

$$(\alpha_1 : \Phi_1) \dots (\alpha_m : \Phi_m) \Delta (\Psi \text{ by } \alpha_1, \dots, \alpha_m).$$

Рассмотрим только те варианты непосредственных обоснований в \mathfrak{D}_i формулы Ψ из формул Φ_1, \dots, Φ_m , которые отсутствуют в \mathfrak{D}_i .

Пусть формула Ψ является аксиомой $t = t$. Тогда задача

$$\langle \alpha_1, \Phi_1 \rangle \dots \langle \alpha_m, \Phi_m \rangle \triangleright \Psi$$

является примитивной. Следовательно, она разрешима в \mathfrak{A}_i .

Пусть формула $\Psi = \Upsilon(t//s)$ получает обоснование с помощью правила (=в) из формул $t = s$ и $\Upsilon(t)$:

$$(\alpha_1 : t = s) (\alpha_2 : \Upsilon(t)) \Delta (\Upsilon(t//s) \text{ by } \alpha_1, \alpha_2).$$

Тогда задача $(\langle \alpha_1, t = s \rangle \langle \alpha_2, \Upsilon(t) \rangle \Gamma \triangleright \Upsilon(t//s))$ разрешима в \mathfrak{A}_i . Действительно, в этом случае $\Upsilon(t) = \left(\Upsilon(t//s) \frac{s, s, \dots, s}{t} \right) \varepsilon$ и, таким образом, поисковое дерево содержит соответствующее допустимой подстановке ε следующее деривационное поддерево, которое возникает в результате применения правила декомпозиции ($LR = \triangleright$):

$$\begin{array}{c} \langle \alpha_2, \Upsilon(t) \rangle \Gamma \triangleright \Upsilon(t) \\ \uparrow \\ \langle \alpha_1, t = s \rangle \langle \alpha_2, \Upsilon(t) \rangle \Gamma \triangleright \Upsilon(t//s) \end{array}$$

□

Также как в [1] в результате получаем полноту алгоритма очевидности \mathfrak{E}_i .

Следствие. Если Δ есть формальное доказательство формулы Ψ из посылок Φ_1, \dots, Φ_m в \mathfrak{D}_i , то алгоритм очевидности \mathfrak{E}_i установит, что “ Δ есть обоснование формулы Ψ с посылками Φ_1, \dots, Φ_m ”.

Список литературы

- [1] Вторушин Ю.И., “О верификации формализованных математических доказательств”, *Интеллектуальные системы*, **24**:1 (2020), 7–24.
- [2] Вторушин Ю.И., “О поиске вывода в системе натуральной дедукции логики предикатов”, *Интеллектуальные системы*, **13**:3 (2009), 263–288.
- [3] Охотников О.А., “О поиске натурального классического логического вывода с использованием частичной скульемизации”, *Интеллектуальные системы*, **23**:4 (2019), 39–90.

References

- [1] Vtorushin Yu.I., “About verification of formalized mathematical proofs (in Russian)”, *Intelligent systems*, **24**:1 (2020), 7–24.
- [2] Vtorushin Yu.I., “O poiske vyvoda v sisteme natural’noj dedukcii logiki predikatov (in Russian)”, *Intelligent systems*, **13**:3 (2009), 263–288.
- [3] Okhotnikov O.A., “About proof-search in classical natural deduction calculus using partial skolemization (in Russian)”, *Intelligent systems*, **23**:4 (2019), 39–90.

Evidence algorithm for first-order logic with equality Vtorushin Yu.I.

The article is a continuation of the article [1]. An algorithm for verifying formalized mathematical proofs for first-order predicate logic with equality is considered. Theorems about its correctness and completeness are proved.

Keywords: automated theorem proving, system for automated deduction, first order language, predicate calculus, production system, artificial intelligence.

Тезис о наличии искусственного интеллекта

Чечкин А.В.¹

Формулируется и обосновывается тезис о необходимом и достаточном требовании к наличию искусственного интеллекта умных систем различного назначения. Необходимость – это присутствие избыточных показаний первой сигнальной системы или первичного сенсориума умных систем. Достаточность – это присутствие второй сигнальной системы или языка общения умных систем как языковой надстройки, языкового сенсориума осознанной части первичного сенсориума. Обсуждаются проблемы создания и безопасной эксплуатации умных систем. Статья написана с целью активного участия в актуальной дискуссии глубокоуважаемых специалистов МГУ имени М.В. Ломоносова по теме «Искусственный интеллект: проблемы и перспективы» [1].

Ключевые слова: искусственный интеллект, умная система, радикал, язык, интернет вещей, интерфейс, естественный интеллект, информационно-системная безопасность интеллекта.

1. Введение. Что такое искусственный интеллект умных систем

В последнее время все шире используют искусственные системы различного назначения, которые относят к классу умных систем (УС). Это — роботы и беспилотники, умные города, умные дороги, мосты, умные светофоры, умные дома, двери, кресла, кровати, телефоны, умные часы, умные инвестиционные проекты (портфели), биржевые роботы, и многие другие умные системы. УС похожи на автоматы. Но автоматы всегда ориентированы на выполнение только своих штатных задач назначения и в условиях только своих театров действий, своих, заранее оговорен-

¹Чечкин Александр Витальевич — доктор физ.-мат. наук, профессор, Военная академия РВСН имени Петра Великого & Финансовый университет при правительстве Российской Федерации e-mail: a.chechkin@mail.ru

Chechkin Alexander Vitalievich — Doctor of Phys.-Math. Sci., Professor, Strategic Missile Forces Military Academy named after Peter the Great & Financial University under the Government of the Russian Federation.

ных, штатных внешних и внутренних ситуаций. Для автоматов доступно только свое внутреннее штатное информационно-системное, в частности, программное обеспечение. Если для автомата требуется обновление, или модификация, или изменение штатного обеспечения, то приходится в этом участвовать разработчику (человеку) извне и менять обеспечение автомата. Поэтому сегодняшние автоматы не относятся к классу *интеллектуальных систем*. Они относятся к классу *кибернетических систем*.

УС, в отличие от автоматов, больше похожи на живые системы. Они способны сами обновляться, развиваться, модифицироваться, адаптироваться и, если надо, могут сами частично преобразовывать свой театр действий, свою проблемную область. Поэтому принято УС относить к классу *интеллектуальных систем*. Сегодня искусственные УС все больше влияют на эффективность поведения людей, контролируют и усиливают их потенциальные возможности [1]. В настоящее время естественный интеллект (ЕИ), оснащенный умными сервисными системами, уверенно позволяет человеку эффективно и безопасно выполнять не только примитивные и рутинные операции своей жизнедеятельности, но благодаря когнитивным технологиям, человек многократно повышает результативность своих мыслительных процессов и во многом обогащает свои эмоции. В настоящей статье рассматривается класс умных систем, главная особенность которых – наличие у них элементов искусственного интеллекта (ИИ). Такие УС реализуют рефлексивное интеллектуальное управление своим поведением в рамках своего театра действий (своей проблемной области). ИИ таких УС обеспечивает функционирование этих систем с учетом своего назначения, с требованием информационно-системной безопасности УС в режиме постоянного обновления, развития самого ИИ УС и преобразования самого театра действий УС.

Главная отличительная черта УС – это наличие у них элементов ИИ, которые обеспечивают *информационно-системную безопасность* [2] их поведения. ИИ УС использует избыточный информационно-системный ресурс и особое *рефлексивное интеллектуальное управление* своим поведением в рамках своего *театра действий* и в режиме *постоянного обновления и развития*. При этом УС не ограничиваются решением только своих штатных задач в штатных ситуациях и только своего театра действия, но постоянно настроены на обновления, на попытки находить решения в некоторых нештатных для себя условиях и ситуациях и, даже, решать некоторые нештатные задачи в рамках театра своих действий.

Напомним, что по учению великого физиолога И.П. Павлова для человека автоматические действия определяются его первой сигнальной системой, связанной напрямую со стимулами, с рецепторами, с ощущениями, с рефлексом на подсознательном уровне. Рефлексы в живых системах соответствуют действиям кибернетических систем, автоматов.

Тогда как интеллектуальное поведение систем, например, сознательные, осознанные действия человека, определяются действиями его второй сигнальной системы, связанной с языком, со словами, с речью [3].

Сегодняшние умные системы уже общаются между собой и человеком в технологиях «интернета вещей» (IoT), в режимах разнообразных интерфейсов. В связи с этим сформулируем главную мысль настоящей статьи о наличии ИИ УС в форме следующего тезиса.

Основной тезис о наличии ИИ УС. По аналогии с естественным интеллектом человека [3], [4] для наличия ИИ УС должно быть выполнено два условия:

1) **Необходимое условие** — наличие *первой сигнальной системы УС* в форме избыточного *первичного сенсориума УС*. Предназначением такой системы является создание в УС и поддержание избыточного ресурса *сенсорных (образных) моделей* как результата мониторинга всего театра действий УС, включая саму УС при помощи сенсорных датчиков УС и исполнителей УС;

2) **Достаточное условие** — наличие *второй сигнальной системы УС* в форме *языкового сенсориума УС*. Такая система является *языковой надстройкой* над первичным сенсориумом УС. Назначением такой надстройки является символическое выделение номинацией (словами и устойчивыми словосочетаниями) осознанной части первичного сенсориума. Языковой сенсориум выполняет роль, во-первых, *языковой координатной семантической системы (лексика)*, во-вторых, является *языковой избыточной информационно-системной моделью (распределенной БД УС и БЗ УС)* осознанной части как самой УС, так и осознанной части театра действий УС и, в-третьих, выполняет роль *языковой операционной системы планирования и интеллектуального управления* поведением УС в рамках своего театра действий УС.

Далее в статье обсудим этот тезис.

2. Первичный сенсориум УС как избыточное отражение самой УС и всего театра ее действий

Согласно тезису о наличии ИИ УС, необходимость *избыточного* моделирования УС подразумевает новый тип моделирования УС в рамках всего театра ее действий. Новый тип моделирования является избыточным и принципиально требует особую модульную форму организации модели, форму *среды (сети) радикалов*. Радикалами (корнями) модели назовем отдельные функциональные модули, которые имеют внешнедоступный механизм включения и выключения их функционирования. Радикалы всегда должны иметь два вида состояний, *активные* и *пассивные*. Ак-

тивные радикалы выполняют свои функции, а пассивные радикалы - нет. Пассивные радикалы по определению находятся в резерве, в запасе, в хранении, в готовности и ожидании до момента их активации. Такой новый тип избыточного моделирования УС называется *радикальным моделированием УС* [1], [2]. Термином «радикальное моделирование» в этом названии подчеркивается с одной стороны *избыточность* модели, а с другой ее свойство *среды радикалов*. Свойство среды радикалов позволяет изменять состояния частей среды радикалов, в пассивное и в активное. Информационно-системный ресурс УС в пассивной форме — это его хранение, но при этом всегда есть возможность в любой момент активировать каждую его часть для выполнения поведенческих функций УС. Напомним, что в УС имеются две взаимосвязанных среды радикалов. Одна — это первичный сенсориум и другая — это языковой сенсориум. Для первичного сенсориума радикалами являются образные (чувственные) модели действительности и связанные с ними рефлексы, навыки УС. Для языкового сенсориума радикалами являются слова и устойчивые словосочетания (лексика), которые тесно взаимно однозначно связаны только с осознанной (выделенной языковыми символами) частью первичного сенсориума, являющейся семантикой лексики языка. Главное качество здесь то, что активация радикалов осознанной части первичного сенсориума одновременно (взаимно однозначно) вызывает активацию соответствующих им языковых радикалов, для которых радикалы первичного сенсориума являются семантикой.

Таким образом, в УС имеются две части первичного сенсориума, под-сознательная и осознанная. Обе части первичного сенсориума являются радикальными образными моделями действительности для УС. Самое главное для УС то, что осознанная часть первичного сенсориума сама представляет единую, но двойственную радикальную модель осознанной части самой УС и осознанной части театра действий ее. Двойственность радикальной модели осознанной части первичного сенсориума означает присутствие двух независимых друг от друга возможных способа активации радикалов этой модели. С одной стороны — это реальный (образный) способ активации, идущий от реальных объектов и отношений действительности через сенсоры УС. При этом способе происходит сначала активация образных радикалов и только после этого, как следствие, активация соответствующих им языковых (символьных) радикалов, т.е. происходит языковое осознание действительных объектов и отношений. С другой стороны второй способ активации осознанной части первичного сенсориума — это языковой способ активации, идущий от языковых символов, от слов и устойчивых словосочетаний, от мыслей и только вслед за этим, как следствие, активация образных радикалов. Второй способ активации обладает важными последствиями такой активации для

интеллектуальных, когнитивных возможностей для УС. Языковая активация - это мысленная активация, не требующая наличия перед УС действительных, реальных объектов и отношений. Она позволяет вести коммуникационные процессы информационно-системного общения между различными УС. Кроме того, она позволяет вести рефлексивные когнитивные процессы УС с самой собой. В частности, вести планирование и проектирование своего поведения, включая свое развитие и разумное преобразование своего театра действий. С появления языкового сенсориума происходит бурное все ускоряющееся его развитие и расширение осознанной части первичного сенсориума за счет сокращения подсознательной части первичного сенсориума. Кроме этого языковый сенсориум способствует расширению и углублению подсознательной части первичного сенсориума.

Термин «радикальное» подчеркивает тот факт, что всегда в избыточной модели такого типа активируется только часть ее ресурсов, достаточных для обеспечения текущих целей поведения УС. А избыточные, ненужные УС в данный момент радикалы такой модели остаются пассивными, в хранении и не мешают текущему поведению УС. Пассивные радикалы находятся в состоянии ожидания, готовности на случай, когда потребуется для УС их использовать. Поэтому избыточность радикальной модели УС не является обременительной, негативной, вредной стороной радикальной модели, усложняющей функционирование УС.

3. Языковый сенсориум УС

Созданию языкового сенсориума УС предваряет выделение в первичном сенсориуме доминирующих (главных) составляющих (объектов) и связей (отношений или рефлексов) между ними как в самой УС, так и в ее театре действий при помощи взаимно однозначного соответствия им символьных, языковых конструкций. Сначала — это присвоение им уникальных символов–имен. В дальнейшем, при развитии и уточнении языкового сенсориума объекты и связи могут добавляться и модифицироваться. Языковый сенсориум УС находится в постоянном обновлении и развитии вширь и в глубину. На языке математической логики такое выделение — это номинирование, наименование собственными именами выделенных *термов логики предикатов* или сети предметов и функциональных отношений между ними [1]. Математически сеть осознанных радикалов может быть представлена (изображена) *мультиграфом*. Непосредственно языковый сенсориум или вторая сигнальная система УС появляется тогда, когда в первичном сенсориуме некоторым радикалам–образам, чувственным моделям конкретных объектов, отношений или радикалам–рефлексам конкретных отношений между объектами присва-

иваются уникальные символы языка (индивидуальные слова–имена) на основе механизма появления в УС приобретенного условного рефлекса в физиологии по И.П. Павлову. В результате этого в радикальной модели первичного сенсориума УС появляется выделенная, осознанная часть, которая имеет собственные имена вершин (объектов) и дуг (отношений). Эта выделенная, поименованная часть первичного сенсориума является семантикой языкового сенсориума УС. Языковому сенсориуму соответствует осознанная часть первичного сенсориума, в которой всем вершинам и всем дугам присвоены собственные имена (идентификаторы, обозначения). На рис.1 изображена осознанная часть первичного сенсориума в виде мультиграфа.

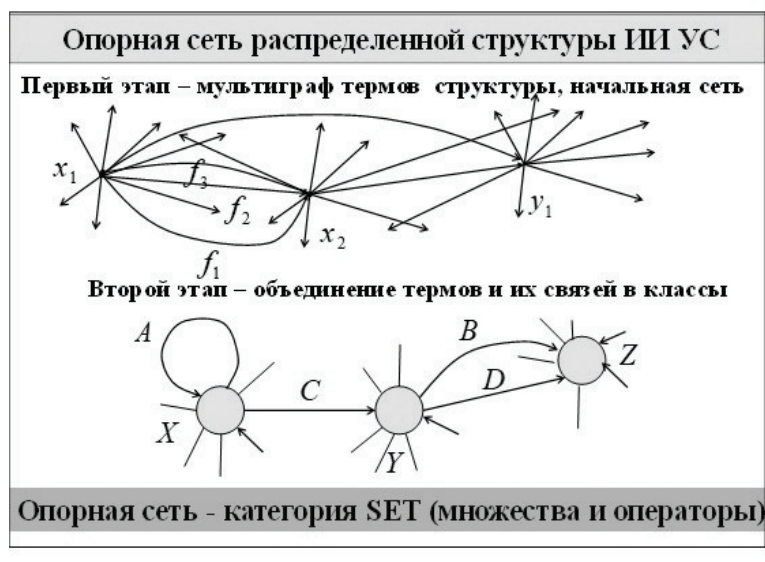


Рис. 1. Создание начальной сети и далее опорной координатной сети языкового сенсориума

Напомним, что такой *мультиграф* — это сеть термов, в котором вершины (точки) обозначены символами, именами $x_1, x_2, \dots, y_1, \dots$ конкретных объектов, составляющих проблемной области. Дуги тоже нагружены именами f_1, f_2, f_3, \dots — это конкретные функциональные отношения между ними. Теперь в первичном сенсориуме на некоторую часть радикалов–образов и радикалов–рефлексов навешиваются уникальные символы–имена. На рис. 1 изображена эта осознанная часть первичного сенсориума, т.е. языковой сенсориум УС. Все его элементы обозначены, пронумерованы, проименованы. Например, ко множеству вершин, конкретных объектов, относятся конкретные: *дом; дерево; человек;*

дорога; город; фирма; банк; кредит; сотрудник; компьютерная программа; штатная задача; двигатель; топливный бак; насос; система стабилизации; корпус ракеты носителя; космический аппарат и т.д.

Множество дуг, конкретных функциональных отношений, например, может состоять из таких конкретных отношений: «около конкретного дома стоит конкретное дерево», «конкретный человек идет по конкретной дороге в конкретный город», «конкретную фирму обслуживает конкретный банк», «конкретный банк выдал конкретный кредит»; «конкретный сотрудник зачислен в конкретной должности оператора»; «конкретная компьютерная программа предназначена для решения конкретной задачи»; «конкретный двигатель содержит внутри себя конкретный насос»; «конкретное топливо поступает из конкретного бака в конкретный двигатель».

Радикальность мультиграфа, т.е. то что вершины и дуги являются радикалами, можно математически формализовать динамикой меняющегося цвета вершин и дуг мультиграфа. Например, красным отмечать активные радикалы в данный момент для УС, а черным — пассивные. Такой динамический мультиграф, бицветный мультиграф, является математической радикальной моделью языкового сенсориума УС. Его активация и деактивация отображается сменой цветовой раскраски мультиграфа, рис. 1. Динамический процесс смены цвета мультиграфа моделирует работу, динамику активации разных частей языковой радикальной (избыточной) модели УС. Сами символы языка могут быть разной природы, звуковые (аудио — слова, речь), графические (видео — слова, письмо), жестикогулярные (динамические образы) и др. Например, в естественном языке (ЕЯ) уникальные символы — это индивидуальные, уникальные, собственные имена, названия, обозначения и др. (см. мультиграф на рис. 1). Это качество языка называется номинативность языка.

4. Синтаксис и семантика языкового сенсориума

В рамках естественного интеллекта (ЕИ) человека и его центральной нервной системы язык человека является языковым *нейронным дубликатом* осознанной части первичного сенсориума, рис. 2. Рассмотрим вторую сигнальную систему человека с точки зрения современного модельного подхода, развиваемого на биологическом факультете МГУ имени М.В. Ломоносова доктором биологических наук Воронковым Г.С., [3], [4]. С появлением ЕЯ у человека происходит ускорение познания осознанной части первичного сенсориума и одновременно развитие самого языкового сенсориума. Вторая сигнальная система человека сразу выступает как *когнитивная система*. Обе системы, первичный сенсориум

(образы и рефлексы) и языковой сенсориум (осознанные объекты и навыки), теперь взаимодействуют между собой и способствуют оперативному взаимовлиянию друг на друга и ускоренному развитию друг друга, рис. 2.



Рис. 2. Языковая система человека как дубликат осознанной части первичного сенсориума человека по Воронкову Г.С. [3], [4]

При этом, по Воронкову Г.С. [3], [4], у человека лидером (ведущим) такого взаимодействия двух сенсориумов становится то один сенсориум, то другой сенсориум. Оба сенсориума бурно взаиморазвиваются. При этом, благодаря языку, сам первичный сенсориум человека постоянно расширяется и углубляется за счет появления новых непознанных объектов и новых отношений между ними, а его осознанная часть тоже все более глубоко изучается, структурируется и тоже постоянно расширяется. Проблему возможности полного осознания всего непознанного мира человека оставим в стороне.

Каждый символ (имя) языкового сенсориума должен подчиняться правилам *грамматики языка*, т.е. правилам *морфологии* (построения слов) и правилам *синтаксиса* (построения словосочетаний, текстов). При этом каждое слово и словосочетание должны быть представлены в языковой радикальной модели УС в виде отдельного радикала-слова, т.е. радикала-уникума, радикала-имени и т.д. Причем такой радикал-слово обязательно должен быть взаимно однозначно связан по И.П. Павло-

ву посредством механизма приобретенного условного рефлекса со своей *семантикой*, с радикалом–образом конкретного объекта или отношения, или рефлекса в первичном сенсориуме. Теперь в языковом сенсориуме УС всякое активирование радикала–имени происходит одновременно с активированием соответствующего ему радикала–образа конкретного объекта или конкретного отношения. Например, в ЕЯ для человека слова «твоя бабушка», «этот дом», «порядок в доме твоей мамы» и др. – все эти словосочетания – названия (указания) единичных объектов и единичных отношений. Теперь в УС, с появлением языка некоторые радикалы–образы конкретных объектов или конкретных отношений могут быть активированы *двумя способами*.

Один способ – чувственный, образный через следящие системы, соответствующие сенсорам УС, терминальным датчикам УС и исполнителям УС. Этот первый способ, чувственный способ, осуществляется через механизм активирования соответствующей следящей системы сенсоров УС на реальные объекты, отношения и рефлексы. Например, узнаваемый для УС объект, узнаваемое для УС место.

Другой способ, который появился с языком, с радикалами–именами – это активирование радикалов–образов через другой механизм активирования, способ активирования через канал языковой связи, через активирование радикалов–имен. Это способ восприятия не самого конкретного объекта, отношения, рефлекса, а через восприятие индивидуального имени, языкового символа. Например, как у человека в ЕЯ через слова «твоя бабушка», а не через появление перед человеком самой его бабушки. С появлением языкового сенсориума УС может реагировать на конкретные объекты, на конкретные отношения между ними кроме первичного наглядного способа еще вторичным языковым способом, реагировать только лишь на их символичные, словесные заменители. При этом в УС происходит следующий процесс. По каналу языковой связи УС воспринимает и распознает символ–имя (слово), что немедленно приводит в УС к активированию (воспоминанию) чувственного образа, семантики этого слова. Это важнейшее качество языкового сенсориума или второй сигнальной системы УС называется *информативностью языка*. Такое название подчеркивает то, что для активации радикала–образа по символу (имени) используется не восприятие непосредственно объектов или отношений, или рефлексов, а обработка только их символов через языковой (информационный) канал связи. Таким образом, благодаря языку, УС может мысленно идентифицировать и далее экспериментировать с объектами, их отношениями, рефлексам в рамках радикальной модели УС посредством языковых символов (имен) и не требовать наличия самих объектов, самих отношений, самих рефлексов.

5. Появление абстрактного уровня языка УС

С возникновением второй сигнальной системы УС, происходит постоянное усложнение и развитие языка УС. Следовательно, в УС происходит развитие ИИ УС. Обсудим очередной важный этап в развитии языка — создание нового иерархического абстрактного уровня в языковом сенсориуме УС, когда происходит создание *понятий* в языке или выделение классов схожих объектов, классов схожих отношений между объектами, классов схожих рефлексов. Этот процесс приводит к появлению в языке *языковой координатной семантической структуры*, которую естественно называть *координатной сетью понятий* или *опорной сетью* языкового сенсориума УС, рис. 1. Опорная координатная сеть языка УС представляет семантическую структуру осознанной части первичного сенсориума УС в удобной для интерфейса человека языковой форме. В физиологии человека опорной координатной сетью является *лексика ЕЯ*, семантическая сеть основных слов-понятий. Теперь в осознанной части первичного сенсориума УС наряду с выделением отдельных конкретных объектов, отдельных отношений, отдельных рефлексов (мультиграф, рис.1), возникает новый иерархический уровень слов-понятий, когда некоторые семейства объектов, семейства отношений, семейства рефлексов группируются в классы эквивалентности (сходства). Появляется *абстрактный* уровень языкового сенсориума, уровень понятий, уровень классов (множеств) и операторов (морфизмов). На таком абстрактном уровне ИИ УС, на уровне опорной координатной сети языкового сенсориума представлены только классы объектов (слова-понятия) и классы отношений (слова-термины). Математически классы объектов удобно называть *опорными множествами* и присваивать им соответствующие собственные индивидуальные имена, слова-понятия X, Y, Z, \dots . Конечно, каждый класс будет представлен в языковой части сенсориума однозначным ему именем в форме радикала-контейнера. Радикалы-контейнеры — это символы, имена, термины абстрактных понятий языка УС. Их активирование вызывает активирование всего класса соответствующих радикалов-уникумов (имен). Аналогично происходит в ЕЯ человека, рис. 1. Теперь на этом новом иерархическом уровне языкового сенсориума УС структурное представление осознанной части театра действий УС (действительности) упрощается и модель вместо формы подробного мультиграфа имеет вид *языковой опорной координатной сети*, рис. 1. Далее таким же образом поступим со всеми функциональными отношениями в области термов, группируя их, объединив их в опорные блоки (классы), которые будем называть *опорными операторами* и обозначать свими индивидуальными словами-понятиями A, B, C, D, \dots . При этом каждое слово-понятие будем реализовывать (оформлять) в модели радикалом-

морфизмом и жестко связывать их активирование с активированием всего блока частных отношений так же, как в ЕЯ человека, рис. 1 и 2.

Примерами опорных множеств могут быть следующие семейства однотипных объектов: *класс домов = деревянный дом; семейство деревьев = дуб; класс людей = программист; набор дорог = шоссе; множество городов = город миллионик; все финансовые средства фирмы; все целевые задачи УС; все компьютерные программы УС; весь персонал УС; все должности УС; все составляющие определенного двигателя; все узлы ракеты носителя; все составляющие космического аппарата; все составляющие наземного комплекса управления* и т.п. В математической логике абстрактным понятиям соответствуют *одноместные предикаты*. При этом, опорное множество — это семантика понятия, интерпретация одноместного предиката, рис. 1. В лингвистике ЕЯ человека это соответствует словам-понятиям семантике слов-понятий, объемам понятий, т.е. относится к лексике языка. Аналогично, опорным операторам на практике соответствуют блоки (классы) *соответствий, ссылок* или блоки (классы) *функциональных отношений* и т.п., рис. 2. На практике опорными операторами могут быть: *финансовые расходы (потoki) средств фирмы; назначения (распределение) всего персонала по должностям фирмы; распределение программного обеспечения (оснащения) по компьютерным средствам; соответствие и распределение рекомендаций математических методов (алгоритмов, программ) по соответствующим задачам*. Заметим, что опорным операторам часто соответствуют в ЕЯ человека не отдельные слова-понятия, а устойчивые словосочетания и даже устойчивые предложения, поясняющие их семантику (смысл).

В терминах современной математики семейство всех множеств (опорных множеств) и морфизмов (опорных операторов) между ними называют *категорией SET* [1], [2]. Категория в нашем случае должна быть средой радикалов и расширена динамикой выделения (активирования) в ней отдельных ее частей, т.е. категория должна быть динамичной, работающей. Например, иметь возможность изменять цвет, быть радикальной бихроматической динамической категорией.

6. Два ультраоснащения опорной координатной сети понятий УС

Далее обсудим два следующих этапа развития языка УС и, следовательно, два этапа развития ИИ УС. Напомним, что язык УС структурирует свою семантику, осознанную часть первичного сенсориума. В языке УС появляется новый, информационный уровень абстракции и, следовательно,

но, расширяются когнитивные возможности языка УС и, следовательно, когнитивные качества ИИ УС. На этих этапах развития языка УС и, следовательно, ИИ УС происходят два ультраоснащения (усложнения) языковой опорной координатной сети понятий, *ультрамножественное* и *ультраоператорное* оснащения опорной сети УС, рис. 1.

6.1. Ультрамножественное оснащение опорной сети УС

Цель такого усложнения состоит в организации новой, информационной возможности идентифицировать конкретные объекты и отношения в осознанной части первичного сенсориума. Если до этого в языке использовалась преимущественно только номинация, индивидуальные имена, то теперь вместо собственного имени для идентификации конкретных объектов, отношений, рефлексов все чаще используются данные о них, набор сведений о них, факты о них. В языке появляется следующий иерархический абстрактный уровень языкового сенсориума УС, сведения, факты и данные о чем-либо. С математической точки зрения в сети происходит ультраоснащение каждого опорного множества и преобразование (замещение) его в соответствующее ему ультрамножество [1], [5]. С практической точки зрения ультраоснащение всякого опорного множества состоит в построении *локальной базы данных* (ЛБД) для объектов только данного опорного множества и только с учетом фиксированного набора координатных свойств этих объектов, а также только своей фиксированной решетки достоверностей [1], [5]. Этот процесс называется *ультрамножественным оснащением* языковой опорной координатной сети [1], [5]. В результате такого ультраоснащения опорная координатная сеть превратится в распределенную структуру из различных ЛБД об объектах каждого из опорных множеств, рис. 3.

Каждое опорное множество порождает свою ЛБД, может быть не единственную. Например, опорное множество обслуживающего персонала фирмы может быть ультраоснащено и породить следующие ультрамножества: *ЛБД отдела кадров*, *ЛБД бухгалтерского отдела*, *ЛБД медицинского отдела* или другие *ЛБД персонала*. Конечно, каждая ЛБД в такой сети УС должна быть сама поименована своим индивидуальным специальным символом, уникальным именем и организована в форме языкового радикала–ЛБД. Например, в математике таким знаком-именем ЛБД может быть символ соответствующего опорного множества с «галочкой» сверху, рис. 3.

Поскольку различные ЛБД, связаны между собой через опорные множества категории SET соответствующими опорными операторами (морфизмами, ссылками), то они образуют сеть ЛБД, точнее распределенную избыточную БД УС в форме сети радикалов, рис. 3. Такая

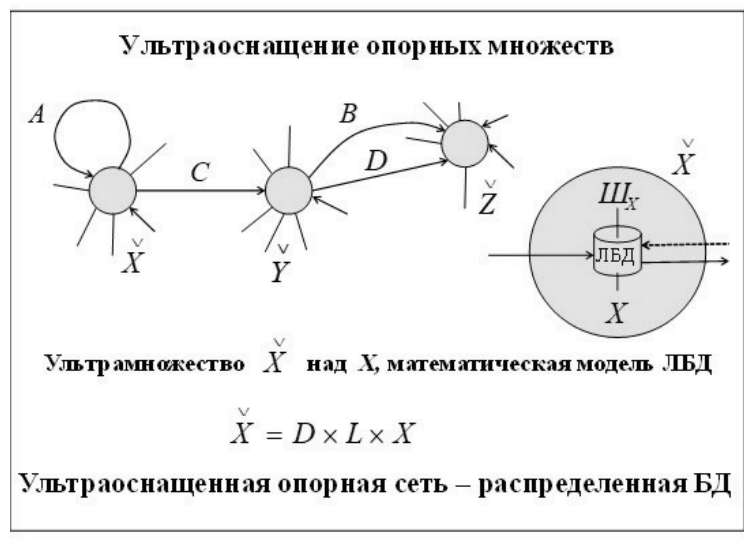


Рис. 3. Ультрамножественная распределенная структура БД УС как сеть ЛБД

ультрамножественная сеть радикалов–ЛБД в пассивном состоянии является радикальной моделью единой, распределенной базы данных (БД) УС, в которой хранятся гипертекстовые (фреймовые) данные об осознанных объектах и осознанных отношениях между ними всего театра действий УС, включая саму УС. При этом радикальная модель такой распределенной сети ЛБД является динамической, имеет полихроматическую разметку и отражает в динамике смены цвета динамику процессов активации в сети, например, динамику распространения запросов и ответов на запросы к отдельным ЛБД такой сети.

Вывод 1. После завершения ультрамножественного усложнения, которое является первым ультраоснащением языковой опорной координатной сети понятий, в языковом сенсориуме УС появится новый абстрактный иерархический уровень в форме сети радикалов–ЛБД или радикалов–ультрамножеств. При этом каждое ультрамножество (ЛБД) будет ориентировано на свое особое опорное множество (тип объектов). Теперь в языковом сенсориуме появляется возможность активировать непосредственно радикалы–ЛБД через канал языковой связи, активируя только их собственные имена ЛБД. Например, устойчивые словосочетания, названия ЛБД. Навигационными преимуществами такой языковой распределенной структуры БД является возможность непосредственно обращаться с запросами ко всей структуре сети, но активировать толь-

ко нужную ЛБД по каналу языковой связи посредством активирования ее индивидуального имени и сенсорного языкового механизма. При этом активируется непосредственно и сразу только нужная ЛБД, минуя остальные ЛБД. Вся остальная сеть ЛБД остается в пассивном состоянии. Напомним, что одновременно с получением ответа на запрос к ЛБД в радикальной модели активируется семантика ответа в форме смысловых радикалов-образов первичного сенсориума. Подчеркнем, что такая распределенная структура БД является языковой, радикальной и семантической.

В распределенной структуре БД УС можно всегда вести любую локальную, автономную модификацию и обновление отдельных ЛБД, вплоть до их ликвидации, или создавать новые ЛБД, не нарушая принципиально всей остальной структуры БД УС. При этом также можно с минимальными негативными последствиями, почти бесконфликтно вводить в распределенную структуру БД новые опорные множества, опорные операторы и далее их ультраоснащать и для них организовывать ЛБД. Этим самым языковая конструкция позволяет не только оперативно вести навигацию, поиск информации (частных данных и частных моделей), но еще безопасно для целостности всей радикальной модели ее обновлять, расширять и развивать. Конечно, одновременно в УС необходимо постоянно проводить тестирование и сертификацию на бесконфликтность всей сети радикалов, рис. 3. Следует заметить, что только это преимущество распределенной структуры БД гарантирует УС отсутствие появления многих системных ошибок и возможных конфликтов в процессе развития и реструктуризации радикальной модели УС, если она необходима.

6.2. Ультраоператорное оснащение распределенной структуры БД УС

Рассмотренное ранее в п. 6.1 усложнение опорной координатной сети понятий радикальной модели УС было связано с *ультрамножественным оснащением опорных множеств*. В результате такого ультраоснащения опорная сеть превратилась из категории SET в языковую структуру единой распределенной БД объектов, отношений между ними, рефлексов осознанной части первичного сенсориума (театра действий УС, включая саму УС), рис. 3. Теперь можно перейти ко второму ультраоснащению, т.е. к оснащению *опорных операторов* категории SET до ультраоператоров [1], [5].

Такое ультраоснащение называется *ультраоператорным оснащением* опорной координатной сети понятий УС. В информатике это оснащение связано с достраиванием опорных операторов до ультраоператоров или до локальных баз знаний (ЛБЗ). Напомним, что ЛБЗ – это локальная экспертная система, отражающая частные знания и умения в узкой, профессиональной, автономной деятельности УС [1], [5]. Отметим, что локализация базы знаний в ЛБЗ происходит каждый раз по своему частному опорному оператору и частным входным и выходным опорным множествам с их частными решетками достоверностей, рис. 4.

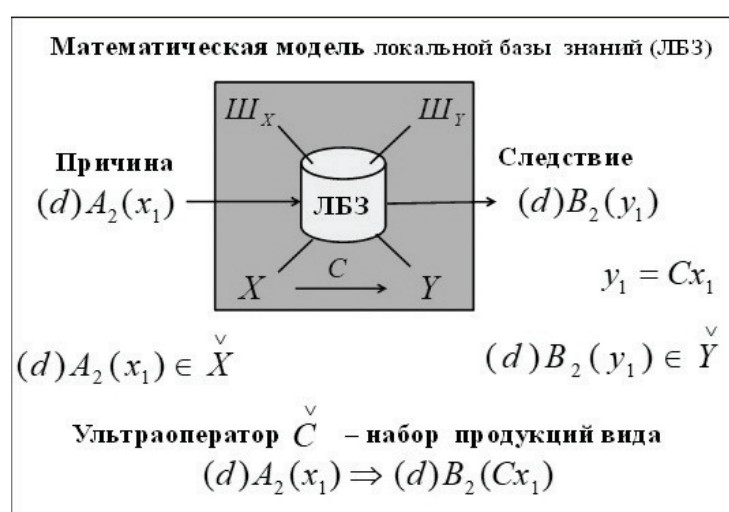


Рис. 4. ЛБЗ соответствует ультраоператору и является математической моделью локальной экспертной системы

Итак, в результате проведения двух ультраоснащений в *языковом сенсории* УС наряду с ультрамножественной сетью, с сетью *распределенной структуры* из ЛБД получим еще совмещенную с ней *ультраоператорную сеть распределенной структуры* из ЛБЗ. В результате языковой сенсории УС превратится в общую распределенную сеть из ЛБД и ЛБЗ. Этим самым в УС появится избыточный информационно-системный ресурс УС в форме единой радикальной модели данных, знаний и рефлексов УС, отражающей семантическую структуру и взаимосвязи в осознанной УС части первичного сенсорииума. Следовательно, такое развитие языка УС способствует развитию ИИ УС до уровня информационно-системного (ультрасистемного) отражения в единой радикальной модели УС осознанной части действительности, т.е. мысленного осмысления самой УС в рамках ее театра действий.

Вывод 2. В каждой УС, осуществив сначала ультраоснащение опорных множеств и затем опорных операторов, завершается построение семантической языковой распределенной структуры единой радикальной модели УС, единой языковой распределенной общей БД и БЗ такой УС и всего театра ее действий. В целом вся сеть является избыточным информационно-системным ресурсом УС или моделью в форме среды радикалов языкового сенсориума. В математике распределенная БД и распределенная БЗ образуют вместе новый тип математической модели в форме динамической полихромной категории нового типа, которую назовем *категорией ULTRASET*, рис. 5.

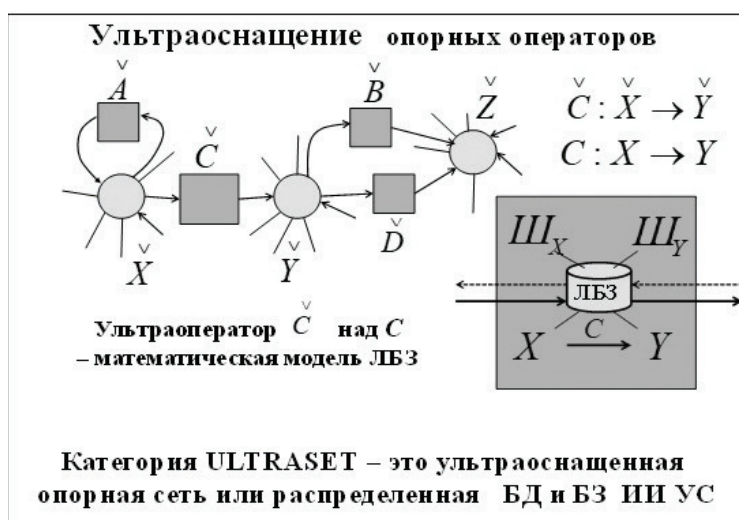


Рис. 5. Семантическая распределенная структура БД и БЗ единой радикальной модели УС

Конечно, такая категория должна быть динамической. Например, она может быть полихроматической, отражающей различные стороны процесса активации радикальной модели, например, раскраской [1], [5]. Распределенная структура БД и БЗ языкового сенсориума является единой радикальной моделью данных, знаний и умений УС, избыточным информационно-системным ресурсом ИИ УС, являющегося ядром ИИ УС.

7. Заключение

Для каждой УС распределенность единой радикальной модели данных, знаний и умений языкового сенсориума этой УС в форме динамической категории ULTRASET имеет следующие принципиальные преимущества перед нераспределенной структурой, рис. 5:

- модель позволяет организовать эффективную *точечную навигацию* в избыточной единой информационно-системной модели УС, используя языковой канал связи и имена ЛБД, ЛБЗ;

- модель обладает высокой *живучестью* и *устойчивостью* функционирования при её коррекции, обновлении, дополнении, модификации и реструктуризации. Распределенную модель УС легко локально дополнять, корректировать, модифицировать, реструктурировать, не искажая при этом другие модули, и, тем самым, не разрушая в целом структуру единого языкового сенсориума, его полной радикальной модели УС и всего театра действий УС. Конечно после всякого обновления радикальной модели необходимо сертифицировать и тестировать эту модель на бесконфликтность, учитывая последствия модификаций;

- модель *некритична* к типам программной реализации отдельных ЛБД и ЛБЗ. Поэтому в единую радикальную модель легко встраиваются уже готовые программные продукты, разработанные в прежних компьютерных технологиях, и ориентированные на автономные технические средства, т.е. можно использовать разные программные модули и программно - технические средства;

- модель обладает свойством *семантической согласованности*. Сетевое семантическое согласование заключается в том, что каждая ЛБЗ должна быть согласована со всеми своими входными ЛБД и со всеми своими выходными ЛБД, по опорным множествам, по шкалам решеток понятий и по решеткам достоверностей, т.е. *по семантике* [1], [5], рис. 6.

Преимущества такого согласования состоят в том, что при соблюдении согласования снимаются многие конфликтные ситуации, связанные с непониманиями и возможным разнобоем ЛБЗ и ЛБД друг с другом. Такая сетевая *семантическая* согласованность единой радикальной модели способствует устойчивости и живучести структуры модели при ее развитии, модификации и реструктуризации. Она не позволяет встроить “чужеродную” ЛБД или ЛБЗ в сеть без соответствующих согласований, рис. 6;

- модель обладает разнообразными полезными качествами, связанные с семантической сетевой согласованностью ЛБД и ЛБЗ, когда используется *ультраоснащение тождественных операторов* опорных множеств. Напомним, что ультраоснащение тождественных операторов приводит к

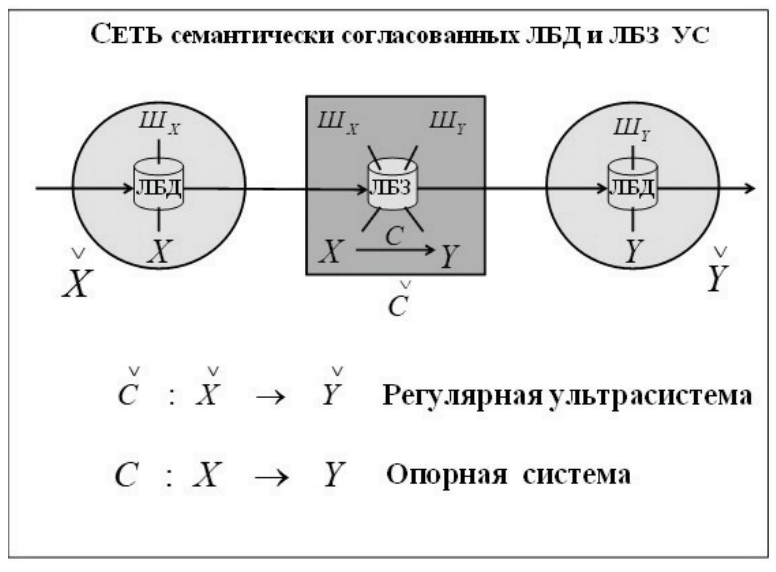


Рис. 6. Семантическое согласование всех ЛБД и ЛБЗ сети

особому важному классу ультраоператоров, называемых *сингулярными* [1], [5]. Такое название подчеркивает принципиальную разницу между сингулярными ультрасистемами и регулярными ультрасистемами, рис. 7.

Примерами сингулярных ультрасистем являются *классификаторы, трансляторы, обобщающие и уточняющие* ультрасистемы. Главная особенность таких ультрасистем, которая отражена в названии *сингулярная ультрасистема* заключается в том, что у них на входе и выходе всегда участвует один и тот же семантический указатель, идентификатор объекта, т.е. одна и та же точка (ультраоснащается тождественный оператор). Однако при этом на входе модели и на её выходе может быть разная по форме информация об этой точке. Сингулярные ультраоператоры необходимы для организации безконфликтного *сетевого семантического согласования* частей единой радикальной модели УС различного происхождения. Например, такое встречается в случаях, когда ЛБД и ЛБЗ разрабатываются различными разработчиками или в радикальной модели должны находиться разнообразные ЛБЗ, ориентированные на одно и то же опорное множество;

- модель позволяет эффективно учитывать семантическое разнообразие сенсорных элементов УС на любом иерархическом уровне языкового сенсориума. При этом используется тот факт, что частным, но важным

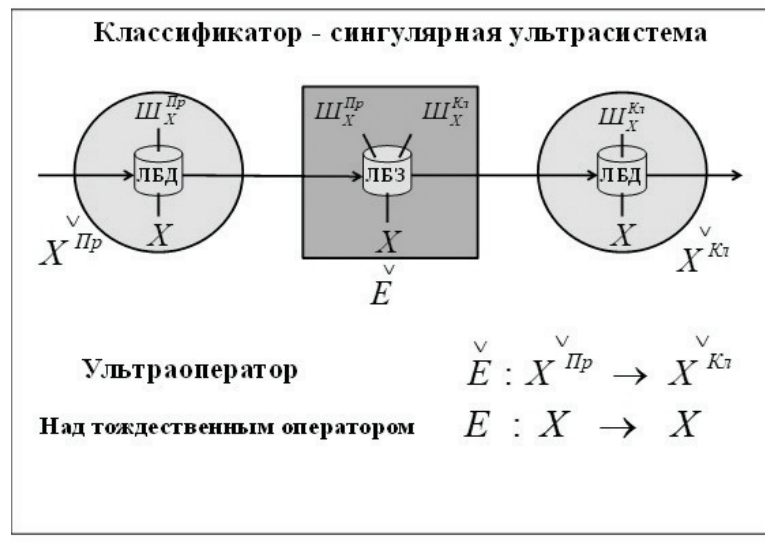


Рис. 7. Сетевая согласованность сингулярной ультрасистемы

случаем сингулярных ультрасистем являются сенсорные элементы полной радикальной модели УС, т.е. датчики и исполнители УС [5], рис. 8.

Таким образом, сопоставляя начальную языковую опорную координатную сеть (лексику) радикальной модели УС в виде динамической категории SET, рис. 2, и окончательную сеть в форме динамической категории ULTRASET, рис. 5, после проведения двойного ультраоснащения категории SET, можно отслеживать поэтапный процесс познания осознанной части первичного сенсориума. Таким образом можно организовать управление процессом дальнейшего развития языка и, следовательно, процессом дальнейшего развития ИИ УС в рамках театра ее действий. Одновременно с этим можно организовать процесс усложнения структуры языкового сенсориума, т.е. управлять дальнейшим развитием единой радикальной модели УС. Эти важнейшие вопросы развития ИИ УС относятся к работе следующего *когнитивного уровня* языкового сенсориума, следующего за уровнем единой радикальной модели УС в рамках ее театра действий. Назовем этот новый иерархический уровень ИИ УС *языковой операционной системой планирования и управления поведением УС*. Вопросы этого когнитивного уровня языка УС тесно связаны с вопросами рефлексии УС и интеллектуального управления развитием УС будут рассмотрены в следующей статье: «Основной принцип искусственного интеллекта».

Сенсорные элементы ИИ УС

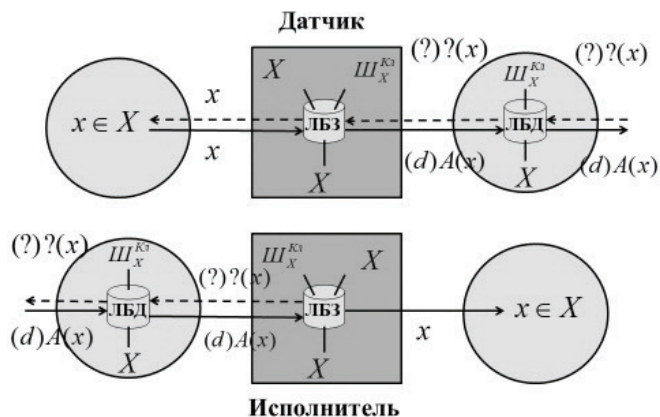


Рис. 8. Сенсорные элементы как сингулярные ультрасистемы

Список литературы

- [1] Кудрявцев В.Б., Козлов В.Н., Рыжов А.П., Мазуренко И.Л., Боков Г.В., Петюшко А.А., “Искусственный интеллект: проблемы и перспективы”, *Интеллектуальные системы: теория и приложения*, **24**:4 (2020), 33-44.
- [2] Потюпкин А.Ю., Чечкин А.В., *Искусственный интеллект на базе информационно – системной избыточности*, «КУРС», Москва, 2019, 382 pp.
- [3] Васенин В.А., Пирогов М.В., Чечкин А.В., *Информационно – системная безопасность критических систем*, «КУРС», Москва, 2018, 348 pp.
- [4] Воронков Г.С., Чечкин А.В., “Проблемы моделирования сенсориума и языковой системы естественного интеллекта индивидуума”, *Интеллектуальные системы*, **2**:1-4 (1997), 23-34
- [5] Воронков Г.С., Потюпкин А.Ю., Чечкин А.В., “Вторая сигнальная система искусственного интеллекта”, *Тезисы докладов XVIII Всероссийская научная конференция “Нейрокомпьютеры и их применение”. 17 марта 2020 г., М.: ФГБОУ ВО МГППУ*, 2020, 111-112
- [6] Чечкин А.В., *Математическая информатика*, «Наука», Москва, 1991, 412 pp.

References

- [1] Chechkin A.V., “Thesis about the availability of artificial intelligence. (in Russian)”, *Intellectual systems*, **25**:1 (2021), 29-49
- [2] Potyupkin A. Yu., Chechkin A.V., *Artificial intelligence based on information and system redundancy (in Russian)*, «KURS», Moscow, 2019, 382 pp.

- [3] Voronkov G.S., Chechkin A.V., “Problems of modeling the sensorium and the language system of natural intelligence of the individual (in Russian)”, *Intellectual systems*, **2**:1–4 (1997), 23–34
- [4] Chechkin A.V., *Mathematical informatics (in Russian)*, «Nauka», Moscow, 1991, 412 pp.
- [5] Chechkin A.V., “Classification of basic ultrasystems. (in Russian)”, *Proceedings of the V International Scientific and Practical Conference “Contemporary Mathematics and the Concepts of Innovative Mathematical Education” Moscow, June 1, 2018 - Moscow: MFO Publishing House.*, 2018, 104–119

The thesis about the presence of artificial intelligence Chechkin A.V.

The thesis about the necessary and sufficient requirement for the presence of artificial intelligence in smart systems for various purposes is formulated and substantiated. Necessity is the presence of redundant readings of the first signaling system or the primary sensorium of smart systems. Sufficiency is the presence of a second signaling system or language of communication of smart systems as a linguistic superstructure, a linguistic sensorium of the conscious part of the primary sensorium. The problems of creation and safe operation of smart systems are discussed. The article was written for the purpose of active participation in the current discussion of highly respected specialists of the M.V. Lomonosov on the topic “Artificial Intelligence: Problems and Prospects” [1].

Keywords: artificial intelligence, smart system, radical, language, internet of things, interface, natural intelligence, information and system security of intelligence.

Часть 2.
Специальные вопросы теории
интеллектуальных систем

Треугольники и уравнения Лагранжа

Алешин С.В.¹

Рассматривается задача о равновесном взаимном расположении двух треугольников. Предлагается решение этой задачи на основе метода множителей Лагранжа. Двукратная итерация этого метода дает аналитическое решение.

Ключевые слова: метод Лагранжа, преобразование на плоскости, подобие, вращение, сдвиг, треугольник.

Рассматривается задача о равновесном взаимном расположении двух треугольников. В работе [1] предложена система уравнений, решение которой дает ответ на эту задачу. В данной заметке мы предлагаем применить метод множителей Лагранжа. Решение ищется с помощью двукратной итерации этого метода, что дает аналитическое решение задачи Библ. 2 назв.

На плоскости (X, Y) заданы два треугольника - $\triangle ABC$ с вершинами $A(a_1, a_2)$, $B(b_1, b_2)$, $C(c_1, c_2)$ и $\triangle HGW$ с вершинами $H(h_1, h_2)$, $G(g_1, g_2)$, $W(w_1, w_2)$. Требуется указать отображение $\xi : (X, Y) \rightarrow (X', Y')$,

$$X' = d_1X + d_2Y + \alpha; \quad Y' = -d_2X + d_1Y + \beta, \quad (1)$$

являющееся суперпозицией преобразований подобия и вращения (они определены матрицей D) и сдвига на вектор s :

$$D = \begin{vmatrix} d_1 & d_2 \\ -d_2 & d_1 \end{vmatrix}; \quad s = (\alpha, \beta).$$

Искомое отображение ξ переводит треугольник $\triangle HGW$ в треугольник $\triangle H'G'W'$, такой, что расстояния от «новых» вершин H', G', W' до соответствующих вершин A, B, C равны и минимальны, то есть

$$\rho(A, H') = \rho(B, G') = \rho(C, W').$$

¹Алешин Станислав Владимирович — доктор физ-мат наук, профессор, каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: stanislav.aleshin@rambler.ru.

Aleshin Stanislav Vladimirovich — Dr.Sc., Professor, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

Таким образом, мы приходим к задаче
 Найти $\min \rho(A, H')$
 при условии

$$\begin{cases} \rho(A, H') = \rho(B, G') = \rho(C, W'), \\ H' = D \cdot H + \begin{pmatrix} \alpha \\ \beta \end{pmatrix}; G' = D \cdot G + \begin{pmatrix} \alpha \\ \beta \end{pmatrix}; W' = D \cdot W + \begin{pmatrix} \alpha \\ \beta \end{pmatrix}. \end{cases} \quad (2)$$

В результате решения задачи треугольник $\triangle HGW$ преобразован в треугольник $\triangle H'G'W'$, который находится в равновесном положении с треугольником $\triangle ABC$, при этом расстояние ρ - минимально.

Интерес может представить то, что решение получается в два шага, и каждый из них это стандартный метод Лагранжа [2], то есть в итоге получаем некоторый комбинированный метод.

Чтобы избежать громоздких формульных преобразований, мы рассмотрим несколько численных примеров, которые, на наш взгляд, хорошо иллюстрируют этот метод.

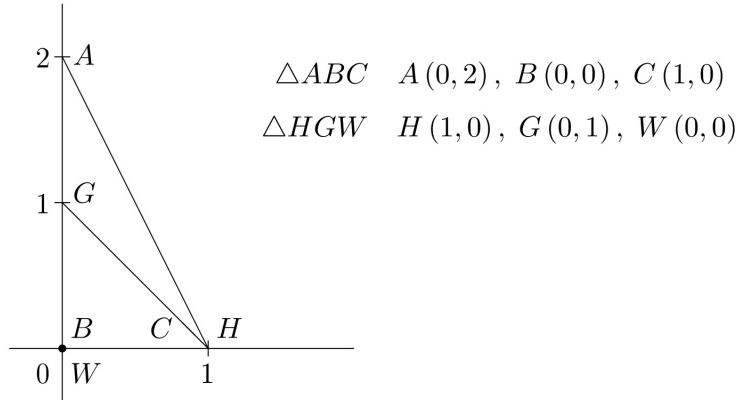


Рис. 1. Пример 1.

Преобразование определяется соотношениями

$$H' = (d_1 h_1 + d_2 h_2 + \alpha; -d_2 h_1 + d_1 h_2 + \beta) = (d_1 + \alpha; -d_2 + \beta),$$

$$G' = (d_1 g_1 + d_2 g_2 + \alpha; -d_2 g_1 + d_1 g_2 + \beta) = (d_2 + \alpha; d_1 + \beta),$$

$$W' = (d_1 w_1 + d_2 w_2 + \alpha; -d_2 w_1 + d_1 w_2 + \beta) = (\alpha; \beta).$$

Вместо параметра ρ в равенствах берем ρ^2 , то есть

$$\rho^2(A, H') = \rho^2(B, G') = \rho^2(C, W'),$$

$$\begin{aligned} [0 - (d_1 + \alpha)]^2 + [2 - (-d_2 + \beta)]^2 &= [0 - (d_2 + \alpha)]^2 + [0 - (d_1 + \beta)]^2 = \\ &= [1 - \alpha]^2 + \beta^2. \end{aligned}$$

Таким образом, экстремальная задача для данной пары треугольников имеет вид

$$\begin{aligned} [1 - \alpha]^2 + \beta^2 &\rightarrow \min, \\ \begin{cases} d_1^2 + d_2^2 + 2d_1\alpha - 2d_2\beta + 2\alpha - 4\beta + 4d_2 + 3 = 0, \\ d_1^2 + d_2^2 + 2d_2\alpha + 2d_1\beta + 2\alpha - 1 = 0. \end{cases} \end{aligned} \quad (3)$$

В качестве функционала взята величина $\rho^2(C, W') = (1 - \alpha)^2 + \beta^2$, что в дальнейшем приводит к упрощению уравнений.

Для решения задачи применим метод множителей Лагранжа. Функционал Лагранжа имеет вид

$$\mathcal{L} = (1 - \alpha)^2 + \beta^2 + \lambda_1 \cdot R_1 + \lambda_2 \cdot R_2.$$

Здесь R_1 (соответственно R_2) – левая часть первого (соответственно второго) равенства системы (3).

Система уравнений Лагранжа имеет вид

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial d_1} = \lambda_1(2d_1 + 2\alpha) + \lambda_2(2d_1 + 2\beta) = 0, \\ \frac{\partial \mathcal{L}}{\partial d_2} = \lambda_1(2d_2 + 4 - 2\beta) + \lambda_2(2d_2 + 2\alpha) = 0, \\ \frac{\partial \mathcal{L}}{\partial \alpha} = -2(1 - \alpha) + \lambda_1(2d_1 + 2) + \lambda_2(2d_2 + 2) = 0, \\ \frac{\partial \mathcal{L}}{\partial \beta} = 2\beta + \lambda_1(-2d_2 - 4) + \lambda_2(2d_1) = 0. \end{cases} \quad (4)$$

Исключая множители λ_1, λ_2 из первых двух уравнений, получаем равенство (левую часть этого равенства обозначим R_3):

$$R_3 = (d_1 + \alpha)(d_2 + \alpha) - (d_1 + \beta)(d_2 + 2 - \beta) = 0. \quad (5)$$

Это соотношение берем в качестве уравнения связи второй экстремальной задачи

$$\begin{cases} (1 - \alpha)^2 + \beta^2 \rightarrow \min, \\ R_3 = 0. \end{cases} \quad (6)$$

Для этой задачи уравнения Лагранжа имеют вид

$$F = (1 - \alpha)^2 + \beta^2 + \lambda \cdot R_3,$$

$$\frac{\partial F}{\partial d_1} = \lambda \cdot \frac{\partial R_3}{\partial d_1} = \lambda[d_2 + \alpha - d_2 - 2 + \beta] = \lambda[\alpha - 2 + \beta] = 0,$$

$$\frac{\partial F}{\partial d_2} = \lambda \cdot \frac{\partial R_3}{\partial d_2} = \lambda[d_1 + \alpha - d_1 - \beta] = \lambda[\alpha - \beta] = 0.$$

Отсюда $\alpha - \beta = 0, \alpha + \beta = 2$, т.е. $\alpha = \beta = 1$.

Оставшиеся два уравнения

$$\frac{\partial F}{\partial \alpha} = -2(1 - \alpha) + \lambda[2\alpha + d_1 + d_2] = 0,$$

$$\frac{\partial F}{\partial \beta} = 2\beta + \lambda[2\beta - d_2 + d_1 - 2] = 0.$$

Отсюда, при $\alpha = 1$

$$2 + d_1 + d_2 = 0.$$

Подставляя $d_1 = -2 - d_2, \alpha = 1, \beta = 1$ в любое из уравнений (3), получаем решение, которое должно удовлетворять ограничениям экстремальной задачи (3):

$$2d_2^2 + 4d_2 + 1 = 0 \quad \text{или} \quad d_2 = -1 \pm \frac{\sqrt{2}}{2}.$$

В итоге имеем два решения задачи (3):

$$\alpha = 1, \beta = 1, d_1 = -1 - \frac{\sqrt{2}}{2}, d_2 = -1 + \frac{\sqrt{2}}{2}$$

или

$$\alpha = 1, \beta = 1, d_1 = -1 + \frac{\sqrt{2}}{2}, d_2 = -1 - \frac{\sqrt{2}}{2}.$$

Новое положение треугольника $\Delta H'G'W'$ определяется вершинами (рис. 2)

$$H' = \left(-\frac{\sqrt{2}}{2}; -\frac{\sqrt{2}}{2} + 2 \right); G' = \left(\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2} \right); W' = (1, -1).$$

При этом выполнены нужные равенства.

$$\rho(A, H') = \rho(B, G') = \rho(C, W').$$

Второй пример показывает, что уравнения Лагранжа могут давать не наилучшее, а допустимое решение.

Исходная конфигурация приведена на рис. 3. Она похожа на задачу 1, однако отличается выбором пар соответствующих вершин. Вершине $A(0, 2)$ теперь соответствует вершина $G(0, 1)$, вершине $B(0, 0)$ - вершина $W(0, 0)$, и вершине $C(1, 0)$ - вершина $H(0, 0)$.

Нетрудно заметить, что положение равновесия достигается, если поднять треугольник ΔWGH вверх по оси Y на $\frac{1}{2}$. При этом расстояния между соответствующими вершинами станут равны $\frac{1}{2}$. Такому решению соответствуют значения параметров $d_1 = 1, d_2 = 0, \alpha = \frac{1}{2}, \beta = \frac{1}{2}$.

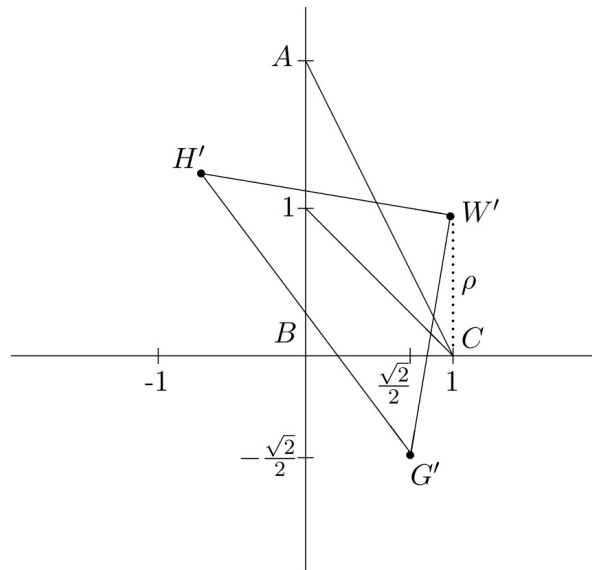


Рис. 2.

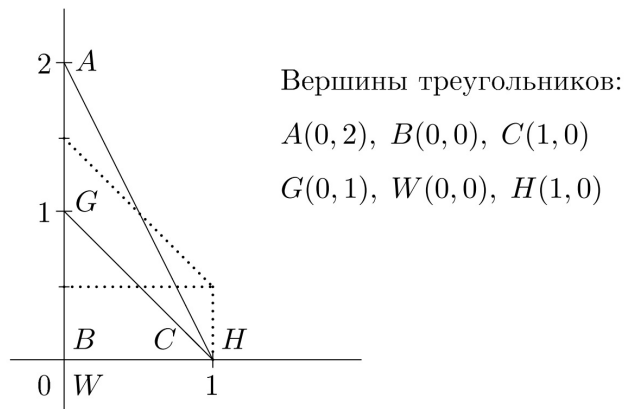


Рис. 3.

Экстремальная задача имеет вид

$$\min(\alpha^2 + \beta^2),$$

$$\begin{cases} (d_2 + \alpha)^2 + (2 - d_1 - \beta)^2 - (\alpha^2 + \beta^2) = 0, \\ (1 - d_1 - \alpha)^2 + (-d_2 + \beta)^2 - (\alpha^2 + \beta^2) = 0. \end{cases} \quad (7)$$

Функционал Лагранжа:

$$\Phi = \alpha^2 + \beta^2 + \lambda_1[1] + \lambda_2[2].$$

Здесь, [1] (соответственно, [2]) - левая часть первого (соответственно, второго) равенства системы (7).

Система уравнений Лагранжа:

$$\frac{\partial \Phi}{\partial d_1} = \lambda_1[d_1 + \beta - 2] + \lambda_2[d_1 + \alpha - 1] = 0,$$

$$\frac{\partial \Phi}{\partial d_2} = \lambda_1[d_2 + \alpha] + \lambda_2[d_2 - \beta] = 0.$$

Устраняя λ_1 и λ_2 , получаем равенство

$$R = \alpha^2 + \beta^2 + d_1\beta - d_2\beta + d_1\alpha + d_2\alpha + d_2 - \alpha - 2\beta = 0, \quad (8)$$

которое используем для получения второй системы Лагранжа. Функционал Лагранжа

$$\hat{\Phi} = \alpha^2 + \beta^2 + \lambda \cdot R.$$

Система уравнений

$$\frac{\partial \hat{\Phi}}{\partial d_1} = \lambda \frac{\partial R}{\partial d_1} = \lambda[\beta + \alpha] = 0, \text{ т.е. } \beta = -\alpha,$$

$$\frac{\partial \hat{\Phi}}{\partial d_2} = \lambda \frac{\partial R}{\partial d_2} = \lambda[-\beta + \alpha + 1] = 0, \text{ т.е. } -2\beta + 1 = 0, \beta = \frac{1}{2}, \alpha = -\frac{1}{2},$$

$$\frac{\partial \hat{\Phi}}{\partial \alpha} = 2\alpha + \lambda \frac{\partial R}{\partial \alpha} = -1 + \lambda[-2 + d_1 + d_2] = 0,$$

$$\frac{\partial \hat{\Phi}}{\partial \beta} = 2\beta + \lambda \frac{\partial R}{\partial \beta} = 1 + \lambda[-1 + d_2 - d_1] = 0.$$

Из двух первых равенств этой системы получаем $\beta = \frac{1}{2}, \alpha = -\frac{1}{2}$. Используя два оставшихся равенства, получаем

$$-3 + 2d_1 = 0, \quad d_1 = \frac{3}{2}.$$

Подставляя в основные уравнения (7), получаем

$$d_2^2 - d_2 - \frac{1}{4} = 0, \quad d_2 = \frac{1}{2} \pm \frac{\sqrt{2}}{2}.$$

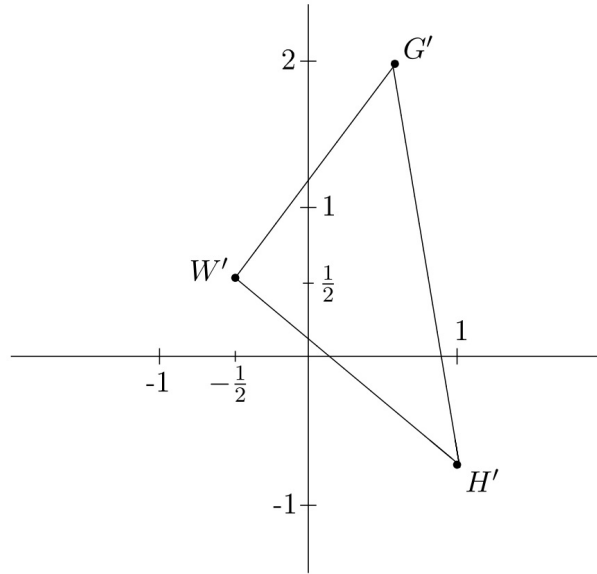


Рис. 4.

Таким образом, решение задачи

$$G' = \left(\frac{\sqrt{2}}{2}, 2 \right); \quad H' = \left(1, -\frac{\sqrt{2}}{2} \right); \quad W' = \left(-\frac{1}{2}, \frac{1}{2} \right).$$

Это решение приведено на рис. 4.

Наконец, пример с тупоугольным треугольником, который уже находится в «равновесном» положении, и решение уравнений Лагранжа должно это отражать. На рис. 5 приведена исходная ситуация.

Следуя той же схеме решения, что в двух предыдущих задачах, рассмотрим «неподвижный» треугольник $\triangle ABC$ и «подвижный» тупоугольный треугольник $\triangle HGW$.

Задача имеет вид

$$\begin{aligned} \rho^2(A, H') &\rightarrow \min, \\ \begin{cases} \rho^2(A, H') - \rho^2(C, W') = 0, \\ \rho^2(B, G') - \rho^2(C, W') = 0. \end{cases} \end{aligned} \quad (9)$$

Уравнения Лагранжа:

$$\frac{\partial}{\partial d_1} = \lambda_1 [\beta - \alpha] + \lambda_2 \left(2 \left(\frac{5}{4} - \frac{\sqrt{2}}{2} \right) d_1 + \alpha \left(-\frac{3}{2} + \frac{\sqrt{2}}{2} \right) \right) +$$

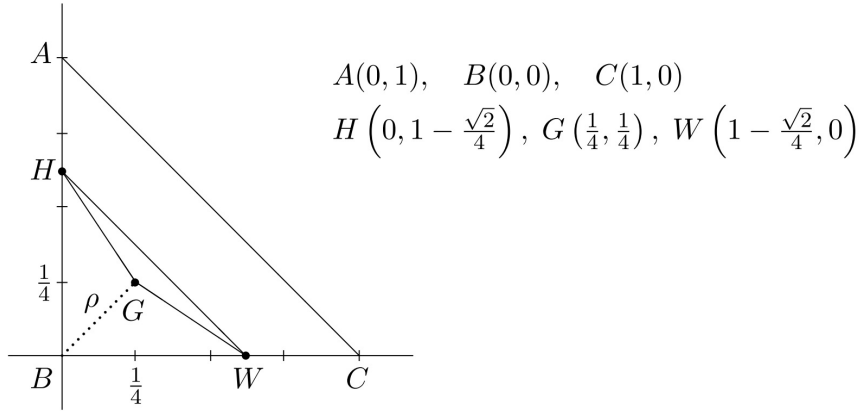


Рис. 5.

$$\begin{aligned}
 & +\lambda_2 \left(+\frac{3}{2}\beta - 2 \left(1 - \frac{\sqrt{2}}{4} \right) \right) = 0, \\
 \frac{\partial}{\partial d_2} & = \lambda_1 (\alpha + \beta) + \lambda_2 \left(2 \left(-1 + \frac{\sqrt{2}}{2} \right) d_2 + \frac{1}{2}\alpha \right) + \\
 & +\lambda_2 \left(-\beta \left(-\frac{3}{2} + \frac{\sqrt{2}}{2} \right) \right) = 0.
 \end{aligned} \tag{10}$$

Исключая λ_1 и λ_2 , получаем функционал Φ для второй задачи Лагранжа

$$\begin{aligned}
 \Phi & = (\beta - \alpha) \left(2d_2 \left(-1 + \frac{\sqrt{2}}{2} \right) + \frac{1}{2}\alpha - \beta \left(-\frac{3}{2} + \frac{\sqrt{2}}{2} \right) \right) - \\
 & - (\alpha + \beta) \left(2d_1 \left(\frac{5}{4} - \frac{\sqrt{2}}{2} \right) + \alpha \left(-\frac{3}{2} + \frac{\sqrt{2}}{2} \right) + \frac{1}{2}\beta + 2 \left(1 - \frac{\sqrt{2}}{4} \right) \right) = 0.
 \end{aligned} \tag{11}$$

Вторая задача Лагранжа дает уравнения

$$\begin{aligned}
 \frac{\partial \Phi}{\partial d_1} & = \left[(\alpha + \beta) \cdot 2 \left(\frac{5}{4} - \frac{\sqrt{2}}{2} \right) \right] \lambda = 0, \\
 \frac{\partial \Phi}{\partial d_2} & = \left[(\beta - \alpha) \cdot 2 \left(-1 + \frac{\sqrt{2}}{2} \right) \right] \lambda = 0,
 \end{aligned}$$

из которых следует

$$\alpha + \beta = 0, \beta - \alpha = 0, \alpha = \beta = 0.$$

Подставляя эти значения в (9), получим

$$\begin{aligned} & \frac{1}{8}d_1^2 + \frac{5}{16}d_2^2 + \frac{1}{8}d_1d_2 = \\ & = 2 - 2d_1 \left(1 - \frac{\sqrt{2}}{4}\right) + d_2^2 \left(\frac{9}{8} - \frac{\sqrt{2}}{2}\right) - \frac{\sqrt{2}}{2} + \frac{1}{8}. \end{aligned}$$

Это уравнение выполнено при $d_1 = 1, d_2 = 0$, таким образом, решение

$$\alpha = 0, \beta = 0, d_1 = 1, d_2 = 0.$$

То есть треугольник $\triangle GHW$ остается на месте, при этом расстояния совпадают

$$\rho(A, H') = \rho(B, G') = \rho(C, W').$$

Список литературы

- [1] Козлов В.Н., *Введение в математическую теорию зрительного восприятия*, Издательство центра прикладных исследований МГУ, Москва, 2007.
- [2] Тихомиров В.М., *Рассказы о максимумах и минимумах*, МЦНМО, Москва, 2006.

References

- [1] Kozlov V.N., *Vvedenie v matematicheskuyu teoriyu zritel'nogo vospriyatiya [Introduction to the mathematical theory of visual perception]*, Publishing House of the Center for Applied Research of Moscow State University, Moscow, 2007 (in Russian).
- [2] Tikhomirov V.M., *Rasskazy o maksimumakh i minimumakh [Stories about the highs and lows]*, MCCME, Moscow, 2006 (in Russian).

Triangles and Lagrange Equations

Aleshin S.V.

The paper considers the problem of the equilibrium mutual arrangement of two triangles and proposes a solution to this problem based on the method of Lagrange multipliers. Two-time iteration of this method gives analytical solution.

Keywords: Lagrange method, transformation on a plane, similarity, rotation, shift, triangle.

Возможный подход к задаче прогнозирования спортивных результатов методами анализа данных

Журавлев А. Д.¹

В данной статье построен алгоритм сведения задачи прогнозирования спортивных результатов к задаче бинарной классификации. При этом обоснована оптимальность этого алгоритма с точки зрения применения результатов прогнозирования в игре против букмекерских контор.

Ключевые слова: машинное обучение, прогнозирование спортивных результатов, бинарная классификация.

1. Введение

Современные методы машинного обучения (ассоциативные правила, деревья решений, модель гауссовых смесей, алгоритмы регрессии, нейронные сети, байесовские сети и т. д.) используются во многих областях для решения проблем ассоциации, классификации, сегментации, диагностики и прогнозирования. Вполне логично, что эти алгоритмы находят применение в такой экстремальной человеческой деятельности, как спорт уровня высоких достижений.

Спортивная аналитика – новый быстрорастущий рынок, объем которого превысит \$4,7 млрд к 2021 г. (по прогнозам WinterGreen Research). Около \$1 млрд из них придется на долю хоккея, считают представители стартапа Iceberg Sports Analytics [1].

Принимая во внимание огромное количество исторических данных по хоккею, можно предложить подход к прогнозированию хоккейных матчей – машинное обучение. Параметры игроков и матча вместе с результатом могут составить обучающую выборку. Алгоритм машинного

¹ *Журавлев Артем Дмитриевич* — аспирант каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: artemzhuravlev.msu@gmail.com.

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

обучения с учителем может использовать эту выборку для построения функции предсказания результатов новых матчей.

В данной статье рассмотрен один из возможных способов формализации задачи прогнозирования спортивных результатов и сведение ее к задаче бинарной классификации, для возможности дальнейшего применения классических алгоритмов машинного обучения.

2. Рынок спортивных ставок

Существуют две основные категории ставок на хоккей: предматчевые и live-ставки, различающиеся уровнем коэффициентов. Кроме того, сделать ставку можно не только на победителя матча, но и на множество других факторов, например, на счет в отдельных периодах, победителя с учетом форы и т. д. Прогностические модели в основном ориентированы на предматчевые ставки на победителя, так как именно на этот тип ставок доступно больше всего исторических данных по коэффициентам, что позволяет провести наиболее полную оценку эффективности прогностической модели.

Ставки на хоккейные матчи можно размещать либо в букмекерских конторах (онлайн и оффлайн), либо на биржах ставок. Традиционные букмекеры, например Pinnacle, устанавливают коэффициенты на различные исходы матча, а клиент играет против букмекера.

Коэффициент ставки означает прибыль, которую получит клиент, если верно угадает исход события. Например, если клиент верно спрогнозировал победу команды, коэффициент на которую составляет 3.00, то он получит 2 доллара на каждый поставленный доллар (в добавок к сумме самой ставки, которая возвращается). Если прогноз клиента оказался неверен, он теряет только сумму своей ставки независимо от коэффициентов.

Коэффициенты выражают предполагаемую вероятность исхода матча, то есть оценку букмекером истинной вероятности. В описанном выше примере с коэффициентом 3,00 (1 к 3) предполагаемая вероятность p победы игрока в матче равна 33%.

2.1. Обзор литературы

ИИ все чаще стал применяться к областям, связанным с интеллектуальными играми, такими как, к примеру, игра в го. Программа AlphaGo, разработанная компанией DeepMind (одна из дочерних компаний Google), выиграла у профессионального игрока пять игр подряд. Подробнее результат описан в [2]. Помимо го, системам ИИ покорилась еще одна сверхсложная игра – покер. В марте прошлого года канадские

программисты из университета Альберты создали искусственный разум DeepStack, способный играть в одну из простейших версий покера. Ему удалось стать победителем на одном из турниров по покеру, который проводился под эгидой Международной федерации покера. Подробнее результат описан в [3].

Спорт высоких достижений является инновационной областью применения искусственного интеллекта. В работе [4] авторы описали подход к прогнозированию спортивных событий на примере футбольных матчей. В итоге исследователи создали распределение описывающее футбольный матч и с помощью него получили "теоретический" перевес над букемерскими конторами в 5.5%. К недостаткам можно отнести то множество допущений, при которых было проведено исследование, что в итоге не позволило применить в жизни созданный алгоритм. Так как данные исследования имеют огромную финансовую значимость для рынка спортивных ставок, то других известных результатов в последние годы не было представлено.

2.2. Сбор статистических данных

Исторические данные по хоккейным матчам широко доступны в интернете. Официальные сайты турниров, например, www.khl.ru предоставляют информацию об игроках и результатах матчей, а также результативность спортсмена по каждому матчу. Некоторые источники предоставляют исторические данные в структурированной форме (CSV или Excel файлы). Доступны и платные базы данных – более комплексные, на более длинные периоды и с лучшей точностью. Наиболее релевантные данные, которые можно взять из подобных баз данных, представлены на рисунке 1.

3. Постановка задачи классификации

Определение: Пусть X - множество описаний объекта, Y - множество номеров классов. Существует целевая зависимость - y^* : $X \rightarrow Y$, значения которой известны на объектах конечной обучающей выборки $X^m = \{(x_1, y_1), \dots, (x_m, y_m)\}$. Требуется построить алгоритм $z : X \rightarrow Y$, способный сопоставить произвольному объекту $x \in X$, объект $y \in Y$.

Определение: Пусть $Y^n = (y_1, \dots, y_n)$ - множество событий в матче такого типа, при которых $X_i = \{x_{i1}, x_{i2}\}$ - множество исходов, состоит из двух элементов, где x_{i1} означает, что событие вида y_i наступило, а x_{i2} , что событие вида y_i не наступило. Тогда исходы x_{i1} и x_{i2} назовем противоположными.



Рис. 1. пример статистических данных о матче

Пример: Пусть множество Y - это суммарное количество шайб заброшенных двумя командами, тогда события «было заброшено больше 4.5 шайб» и «было заброшено меньше 4.5 шайб» являются противоположными, а множество Y конечно ввиду ограниченности времени матча.

Определение: Введем функцию

$$f(x_{i1}, x_{i2}) = |x_{i1} - x_{i2}|, \quad (3.1)$$

где x_{i1}, x_{i2} - коэффициенты на противоположные исходы события y_i , тогда событие y_i будем называть оптимальным для прогнозирования если

$$f_i(x_{i1}, x_{i2}) = \min_{y_j \in Y} f(x_{j1}, x_{j2}) \quad (3.2)$$

Что это будет значить с точки зрения оценки вероятностей? Это значит, что события имеют коэффициенты около 2 и эти исходы практически равновероятны. Почему данный вид ставки будет оптимален? Обратимся к теории случайных процессов.

Определение: Одномерное случайное блуждание - это случайный процесс $\{Y_n\}_{n \geq 0}$ с дискретным временем, имеющим вид

$$Y_n = Y_0 + \sum_{i=1}^n X_i \quad (3.3)$$

где Y_0 - начальное состояние, X_i - независимые случайные величины равные 1 с вероятностью p и -1 с вероятностью $1-p$, если $p = q = \frac{1}{2}$ то блуждание симметрично.

Теорема 1 (О возвратности блуждания). *Вероятность $P(Y_n = 0, n > 1 | Y_0 = 0)$ того, что случайное симметричное блуждание вернется в точку старта в одномерном случае равна 1.*

Подробное доказательство приводится в [5]

Определение: ROI (return on investment) - $ROI = \frac{P_n * 100}{s * n}$, где P_n это прибыль на дистанции в n матчей, s -сумма одной ставки, n - количество ставок. ROI – это основной показатель успешности и, соответственно, целевой показатель эффективности прогностической модели.

Замечание: Очевидно, при событии вида «в сумме забросили меньше 4 шайб» пространство исходов состоит из трех элементов: забросили меньше 4 шайб, забросили ровно 4 шайбы, забросили больше 4 шайбы. Но по правилам спортивных ставок, в случае если результат совпал с прогнозируемым событием, то идет возврат ставки, поэтому можно считать что исхода два, так как при третьем не идет никакой потери.

Определение: Игрой “наугад” назовем игру, где на каждом шаге мы равновероятно выбираем один из двух возможных видов ставки.

Гипотеза: Благодаря выбору оптимальных событий для прогнозирования, даже используя стратегию ставок “наугад”, мы играем с минимально возможным отрицательным матожиданием для себя.

Проверка гипотезы: Если предположить, что букмекеры обладали бы истинными вероятностями событий и определяли величину ставки исходя из них, то тогда на большой дистанции наш выигрыш был бы равен нулю. Это утверждение легко следует из расчета математического ожидания выигрыша.

$$\begin{aligned} & \frac{1}{2} * \frac{1}{k_1} * (k_1 - 1) + \frac{1}{2} * (1 - \frac{1}{k_1}) * (-1) + \\ & + \frac{1}{2} * \frac{1}{k_2} * (k_2 - 1) + \frac{1}{2} * (1 - \frac{1}{k_2}) * (-1) = 0 \end{aligned} \quad (3.4)$$

где k_1, k_2 - коэффициенты на то, что событие произойдет или не произойдет. Очевидно, что для того, чтобы букмекерским конторам это стало выгодно, необходимо занижать коэффициенты вводя комиссию. Предположение состоит в том, что в зависимости от величины функции (3.1) размер комиссии разный и необходимо найти события, где при игре “наугад” на длинной дистанции мы будем иметь наименьшее отрицательное математическое ожидание. Проведем моделирование на выборке из 5000 матчей, где решение о выборе ставки принимается наугад и посчитаем среднее ROI при повторении симуляции 10000 раз. На всех возможных событиях для прогнозирования получили наименьшее отрицатель-

ное ROI при событиях оптимальных для прогнозирования, и оно составило $-4,8\%$ против $-7,2\%$ и $-9,4\%$ для неоптимальных. Моделирование для оптимальных событий на рисунке ниже, по оси x ROI (рис 2).

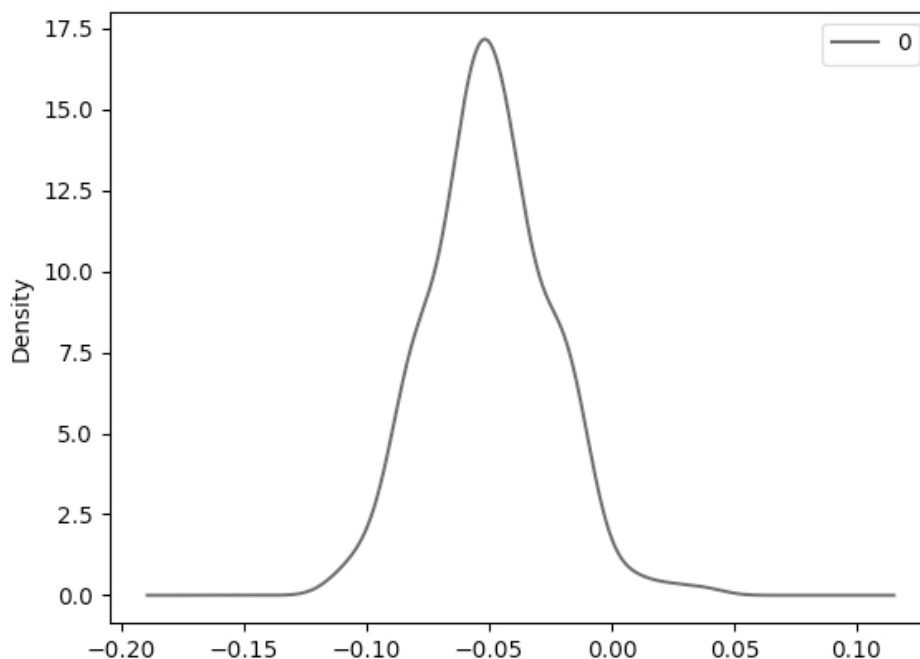


Рис. 2. Плотность распределения ROI.

4. Результаты и выводы

Опираясь на данные моделирования теперь мы можем сформулировать задачу бинарной классификации. В качестве множества событий матча выберем суммарное количество заброшенных шайб в основное время. Это множество конечно. Среди множества событий гарантировано есть событие, оптимальное для прогнозирования, которое позволяет нам использовать теорему 1, а значит минимизировать риски. В качестве целевой переменной будем использовать пространство исходов события, оптимального для прогнозирования. В качестве одной из метрик будем использовать ROI. Таким образом, мы сформулировали задачу бинарной классификации, которую можно теперь решать с помощью методов анализа данных.

Список литературы

- [1] http://azure.cnews.ru/articles/2017-02-14_hokkej_budushchego_intellektualnye_matchi_v_oblake, “Хоккей будущего: интеллектуальные матчи в "облаке””.
- [2] “Google AI algorithm masters ancient game of Go”, *Nature*, 2016, № 529,, 445–446.
- [3] “Superhuman AI for heads-up no-limit poker: Libratus beats top professionals”, *Science*, 2017.
- [4] Lisandro Kaunitz, Shenjun Zhong , Javier Kreiner, “Beating the bookies with their own numbers - and how the online sports betting market is rigged”.
- [5] Марк Кельберт, Юрий Сухов, “Вероятность и статистика в примерах и задачах. Том 2. Марковские цепи как отправная точка теории случайных процессов и их приложения”, 2017, 57.

References

- [1] http://azure.cnews.ru/articles/2017-02-14_hokkej_budushchego_intellektualnye_matchi_v_oblake, “Hokkey buduschego: intellektualnie matchi v oblake [Hockey of the future: intelligent matches in the "cloud] (in Russian)”.
- [2] “Google AI algorithm masters ancient game of Go”, *Nature*, 2016, № 529,, 445–446.
- [3] “Superhuman AI for heads-up no-limit poker: Libratus beats top professionals”, *Science*, 2017.
- [4] Lisandro Kaunitz, Shenjun Zhong , Javier Kreiner, “Beating the bookies with their own numbers - and how the online sports betting market is rigged”.
- [5] Mark Kelbert, Yuri Suhov, “Veroyatnost i statistica v primerah i zadachyah: Tom 2. Markovskie cepi kak otpravnaya tochka teorii sluchainih processov i ih prilozhenia [Probability and statistics in examples and problems. Volume 2. Markov chains as a starting point of the theory of random processes and their applications] (in Russian)”, 2017, 57.

Possible approach to the problem of predicting sports results using data analysis methods

Zhuravlev A. D.

This article is devoted to the construction of an algorithm for reducing the problem of predicting sports results to the problem of binary classification. At the same time, the optimality of this algorithm has been substantiated from the point of view of the application of forecasting results in the game against bookmakers.

Keywords: machine learning, predicting sports results, binary classification.

Распределенный алгоритм поиска траекторий-компаньонов

Соколов А.П.¹, Алисейчик П.А.², Моисеев С.В.³

В работе рассматривается задача построения распределенного алгоритма поиска траекторий-компаньонов. Под траекториями-компаньонами понимается пара траекторий такая, что обе траектории обладают участком достаточной длины, на котором оба объекта в каждый момент времени находятся на достаточно близком расстоянии друг от друга.

Рассматривается задача поиска в базе данных траекторий, которая из-за своего размера не может быть размещена и обработана на одной рабочей станции. Подобные задачи принято называть задачами на больших данных или просто big-data-задачами. Определена процедура квантизации траекторий, вводится понятие клеточного индекса, введено определение похожих траекторий, построен эффективный алгоритм поиска похожих траекторий. На базе данного алгоритма построены локальный и распределенный алгоритмы поиска траекторий-компаньонов, получены оценки их сложности.

Ключевые слова: большие данные, анализ пространственно-временных данных, анализ траекторий, поиск траекторий-компаньонов, поиск близких траекторий, анализ трафика.

¹ *Соколов Андрей Павлович* — к.ф.-м.н., м.н.с. каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: sokolov@intsys.msu.ru

Sokolov Andrey Pavlovich — junior scientific researcher, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

² *Алисейчик Павел Александрович* — к.ф.-м.н., в.н.с. каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: paa.idea@mail.ru

Aliseychik Pavel Alexandrovich — leading scientific researcher, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

³ *Моисеев Станислав Владимирович* — выпускник каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: stanislav.moiseev@gmail.com

Moiseev Stanislav Vladimirovich — graduate student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

1. Введение

В настоящее время сотовыми телефонами пользуются практически все. Более того, на многих автомобилях установлены терминалы, подключенные к сотовым сетям связи. В определенных условиях сотовые телефоны и смартфоны могут формировать ценную информацию о своем местоположении. Эта информация возникает на оборудовании оператора сотовой сети в моменты переключения оборудования пользователя от одной базовой станции к другой. Информация о местоположении пользователя определяется местоположением базовой станции, к которой было осуществлено подключение. Заметим, что данная информация возникает даже при отсутствии в оборудовании пользователя высокоточного навигационного оборудования.

В результате у оператора сотовой сети может быть получена и накоплена информация о местоположении и передвижении всех пользователей. Эта информация может быть использована для решения множества полезных практических задач, таких как: анализ транспортных потоков, определение так называемых транзитных точек, через которые проходит большинство маршрутов, поиск траекторий с заданными свойствами, поиск пар траекторий с взаимными свойствами, моделирование транспортной сети и многие другие.

Одной из интересных задач является задача поиска множества траекторий-компаньонов.

Под траекториями-компаньонами понимается пара траекторий такая, что каждая из траекторий содержит в себе участок достаточной длины, на протяжении которого в каждый момент времени оба объекта находятся на достаточно близком друг от друга расстоянии.

Заметим, что задача поиска всех пар траекторий-компаньонов может быть обобщена на случай поиска множеств траекторий-компаньонов заданной мощности. Алгоритмы, описанные в данной работе, могут быть легко обобщены на этот случай.

Задачи поиска множеств траекторий-компаньонов, осуществляющих совместное перемещение в пространстве, рассматриваются, в частности, в публикациях [1], [2], [3].

Работа [1] посвящена поиску множеств объектов, осуществляющих совместное передвижение. Рассматриваются различные типы поведения таких групп: стадо, конвой и рой. Вводятся понятия «толпы» и «скопления». Далее для данных понятий строятся алгоритмы поиска как для статических баз данных, так и для динамически пополняемых.

В работе [2] рассматривается задача поиска и поддержания в актуальном состоянии множеств траекторий-компаньонов на основе непрерывного потока траекторных данных.

В работе [3] вводятся понятия «рой» и «замкнутый рой». Построен алгоритм для эффективного поиска «замкнутых роев». Получены оценки сложности алгоритма.

Настоящая работа посвящена разработке распределенного алгоритма поиска всех пар траекторий-компаньонов. Распределенный алгоритм описан с использованием операций map/reduce [4], которые широко применяются в современных распределенных вычислительных платформах. Таких как, например, Apache Spark.

2. Определения и постановка задачи

Обозначим $O = \{o^1, o^2, \dots, o^n\}$ — множество движущихся объектов, которые присутствуют в базе данных. Далее, $T = \{t_1, t_2, \dots, t_m\}$ — множество моментов времени, для которых заданы точки траекторий. Множество T задает единую «временную сетку», в которой задаются положения объектов. Будем полагать, что каждая траектория содержит ровно m точек.

Множество моментов времени из T , начинающееся с t_a и заканчивающееся t_b , будем называть отрезком и обозначать $[t_a, t_b]$.

Траекторией объекта o_i будем называть ломаную, заданную конечной последовательностью точек на плоскости для отрезка времени $[t_1, \dots, t_m]$

$$o^i = \langle p_1^i, \dots, p_m^i \rangle.$$

Здесь $p_j^i = (x_j^i, y_j^i, t_j^i)$, где $x_j^i, y_j^i \in \mathbb{R}$ — координаты объекта o_i в пространстве, t_j^i — момент времени.

Заметим, что в реальных базах данных траектории могут иметь различные моменты времени начала и моменты окончания. Также отметим, что возможны ситуации, когда для какого-то момента времени t_j в траектории для объекта o^i может не быть информации. То есть точка p_j^i может отсутствовать в базе данных. В рамках данной работы мы полагаем, что такие ситуации исключены, то есть данные о положении объектов заданы для всех моментов времени из T .

Для простоты будем отождествлять объекты со своими траекториями. Поэтому под o^i будем понимать как объект, так и его траекторию. Множество всех траекторий будем обозначать O .

Заметим, что на практике положения объектов обычно задаются в географических координатах: широта и долгота. Однако в случае, если все рассматриваемые траектории сосредоточены в относительно небольшой области на поверхности Земли, то можно перейти к рассмотрению некоторого отображения всех точек поверхности Земли на плоскость. Такие отображения обычно называют картографическими проекциями.

Одним из примеров такого отображения является равнопромежуточная проекция [7]. В этом случае переход от географических координат к координатам на плоскости выполняется так. Пусть $p' = (\psi, \phi)$ — точка на поверхности Земли с долготой ψ и широтой ϕ , пусть $p'_r = (\psi_r, \phi_r)$ — некоторая базовая точка, расположенная в интересующей нас области. Обозначим $p = (x, y)$ — образ точки p' на плоскости. Тогда:

$$x = R \cdot \cos(\psi) \cdot (\psi - \psi_r),$$

$$y = R \cdot (\phi - \phi_r),$$

где R — средний радиус Земли.

Данное отображение приведено в качестве примера. Дальнейшее изложение принципиально не зависит от выбора той или иной картографической проекции.

Дадим определение траекторий-компаньонов. Рассмотрим $o^i, o^j \in O$ — две траектории. Введем параметры $t_{min}, d_{min} \in \mathbb{R}$. Пару траекторий (o^i, o^j) будем называть (t_{min}, d_{min}) -компаньонами, если существуют моменты времени $t_a < t_b$, $t_b - t_a \geq t_{min}$, $t_a, t_b \in T$, такие что для любого момента времени $t_k \in [t_a, t_b]$ имеет место $d(p_k^i, p_k^j) \leq d_{min}$, где d — обозначает евклидово расстояние. Отрезок $[t_a, t_b]$ будем называть совместным участком траекторий-компаньонов.

Параметры t_{min}, d_{min} характеризуют минимальную продолжительность *совместного участка* двух траекторий и максимально допустимое расстояние между объектами на протяжении совместного участка. Данные параметры являются фиксированными и не зависят от выбора объектов.

Далее, для краткости, траектории, являющиеся (t_{min}, d_{min}) -компаньонами, будем называть просто *траекториями-компаньонами*.

Задача поиска пар траекторий-компаньонов ставится следующим образом: дано множество траекторий $O = \{o^1, o^2, \dots, o^n\}$ и значения параметров $t_{min}, d_{min} \in \mathbb{R}$, необходимо найти все пары (o^i, o^j) , такие что траектории o^i и o^j являются траекториями-компаньонами.

Дальнейшее изложение построено следующим образом. Сначала мы опишем алгоритм вычисления предиката компаньонства и дадим оценку сложности тривиального алгоритма поиска всех пар компаньонов. Затем мы опишем способ квантизации траекторий, позволяющий перейти от рассмотрения вещественных координат, к целочисленным. Далее будет описан алгоритм построения клеточного индекса — специальной структуры данных для эффективного поиска на множестве траекторий. Далее будет введено понятие похожих траекторий и описан алгоритм

для их быстрого поиска. Затем мы построим локальный алгоритм поиска траекторий-компаньонов и дадим оценку его временной сложности. Затем мы опишем распределенный алгоритм поиска траекторий-компаньонов в терминах операций map/reduce.

3. Алгоритм вычисления предиката компаньонства

Введем предикат компаньонства $r(o^u, o^v)$, который равен 1, если траектории o^u, o^v являются компаньонами, и 0 - в противном случае.

Алгоритм для вычисления предиката $r(o^u, o^v)$ описан далее (3.1).

Algorithm 3.1 Вычисление предиката компаньонства — TestPair

```

1: function TESTPAIR(  $o^u, o^v$  )
2:   Дано: пара траекторий  $(o^u, o^v)$ 
3:   Найти: значение предиката  $r(o^u, o^v)$ 
4:   флаг нахождения внутри совместного участка
5:    $flag \leftarrow 0$ 
6:   for  $j=1, \dots, m$  do
7:     if  $(flag = 0) \& (d(p_j^u, p_j^v) < d_{min})$  then
8:       совместный участок начался
9:        $flag \leftarrow 1$ 
10:    end if
11:    if  $flag = 1$  then
12:      if  $(d(p_j^u, p_j^v) < d_{min}) \& (d(p_{j+1}^u, p_{j+1}^v) < d_{min})$  then
13:        совместный участок продолжается
14:         $t_{cmn} \leftarrow t_{cmn} + (t_{j+1} - t_j)$ 
15:      end if
16:      if  $(d(p_j^u, p_j^v) < d_{min}) \& (d(p_{j+1}^u, p_{j+1}^v) \geq d_{min})$  then
17:        совместный участок завершен
18:         $flag \leftarrow 0$ 
19:        проверяем, превышает ли длительность заданный порог
20:        if  $t_{cmn} > t_{min}$  then
21:          return 1
22:        end if
23:      end if
24:    end if
25:  end for
26:  return 0
27: end function

```

Обозначим за F — временную сложность вычисления предиката $r(o^i, o^j)$. Из описания алгоритма 3.1 очевидно следует, что $F = O(m)$, где m — число моментов времени, на которых заданы положения объектов.

Тривиальный алгоритм поиска всех пар траекторий-компаньонов на множестве O предполагает перебор всех пар $(o^i, o^j) \in O^2$ и вычисление для каждой пары

предиката r . Очевидно, что временная сложность такого алгоритма равна $C_n^2 \cdot F = O(m \cdot n^2)$. Заметим, что данная оценка справедлива при одновременном стремлении к бесконечности как числа объектов — n , так и числа точек в траекториях — m .

С практической точки зрения интересна задача построения алгоритма, сложность которого относительно n была бы ниже.

В следующих разделах мы построим алгоритм поиска всех пар компаньонов, который при определенных ограничениях на множество траекторий имеет линейную сложность как по m , так и по n .

4. Квантизация траекторий

В данном разделе описан метод квантизации траекторий. Этот метод предназначен для снижения объема траекторной информации и упрощения ее обработки.

Квантизация траекторий заключается в переходе от рассмотрения точек с вещественными координатами к точкам с целочисленными координатами.

Процедура квантизации основана на разбиении пространства и времени на ячейки фиксированного размера. Обозначим $\Delta_s \in \mathbb{R}$ — размер ячейки в пространстве,

$\Delta_t \in \mathbb{R}$ — размер ячейки во времени.

Пусть $p = (x, y, t)$ — некоторая точка в пространстве-времени. Обозначим

$q(p) = (\hat{x}, \hat{y}, \hat{t})$, где

$$\hat{x} = \left\lfloor \frac{x}{\Delta_s} \right\rfloor, \hat{y} = \left\lfloor \frac{y}{\Delta_s} \right\rfloor, \hat{t} = \left\lfloor \frac{t}{\Delta_t} \right\rfloor, \hat{x}, \hat{y}, \hat{t} \in N.$$

Преобразование $q(p) = (\hat{x}, \hat{y}, \hat{t})$ будем называть (Δ_s, Δ_t) -квантизацией. Значения $(\hat{x}, \hat{y}, \hat{t})$ представляют собой индексы ячеек пространства-времени по соответствующим координатам.

Переход от вещественных координат к целочисленным индексам ячеек позволяет значительно упростить процедуру сравнения траекторий. Однако, такая процедура приводит к снижению точности описания траектории.

Также при квантизации может существенно сократиться объем памяти, необходимый для хранения траекторных данных.

Если мы обозначим максимальные по модулю значения индексов ячеек по пространству и времени как N_s и N_t , то мы можем совместить все три индекса (два индекса пространства и один индекс времени) в один обобщенный индекс ячейки следующим образом

$$\hat{c} = w(\hat{x}, \hat{y}, \hat{t}) = \hat{t} \cdot N_s^2 + \hat{y} \cdot N_s + \hat{x}.$$

Таким образом, последовательное применение функций q и w ставит в соответствие каждой точке p ячейку пространства-времени с индексом $\hat{c} = w(q(p))$. В дальнейшем, для краткости будем использовать следующие обозначения:

$$\begin{aligned}\hat{c}(p) &= w(q(p)), \\ p' &= (x', y', t') = z(\hat{c}), \\ x' &= x(\hat{c}), y' = y(\hat{c}), t' = t(\hat{c}).\end{aligned}$$

Здесь функция \hat{c} ставит в соответствие каждой точке пространства-времени

$p = (x, y, t)$ целочисленный обобщенный индекс ячейки, а функция z каждой ячейке с индексом \hat{c} ставит в соответствие точку $p' = (x', y', t')$, являющуюся геометрическим центром ячейки \hat{c} . Функции $x(\hat{c}), y(\hat{c}), t(\hat{c})$

по индексу ячейки возвращают соответствующие координаты ее геометрического центра.

Отметим также, что переход от индексов по пространству и времени $\hat{x}, \hat{y}, \hat{t}$ к обобщенному индексу \hat{c} полностью обратим:

$$\begin{aligned}\hat{t}(\hat{c}) &= \hat{c} \operatorname{div} N_s^2; \\ \hat{y}(\hat{c}) &= (\hat{c} \operatorname{mod} N_s^2) \operatorname{div} N_s; \\ \hat{x}(\hat{c}) &= \hat{c} \operatorname{mod} N_s.\end{aligned}$$

В результате процедуры квантизации, каждой траектории o^i может быть поставлена в соответствие последовательность обобщенных индексов ячеек — $o^i \sim (\hat{c}_1^i, \dots, \hat{c}_m^i)$.

Квантизацию траектории o^i будем обозначать так: $\hat{o}^i = Q(o^i)$.

Отметим, что чем меньше значения параметров Δ_s, Δ_t , тем ближе будет квантизованная траектория к исходной.

Пусть \hat{c} — обобщенный индекс некоторой ячейки. Обозначим

$$D(\hat{c}) = \{w(\hat{x}(\hat{c}) + dx, \hat{y}(\hat{c}) + dy, \hat{t}(\hat{c}) + dt) \mid dx, dy, dt \in \{-2, 0, 2\}\}$$

множество ячеек, соседних с ячейкой \hat{c} .
Заметим, что если

$$\Delta_s = \frac{d_{\min}}{\sqrt{2}},$$

где d_{\min} — максимально допустимое расстояние между компаньонами, то всякие две точки, которые находятся на расстоянии, не превышающем d_{\min} , в результате процедуры квантизации гарантированно попадут в соседние ячейки. При этом под соседними ячейками понимаются такие ячейки, у которых индексы по каждой из координат отличаются не более, чем на 2. Далее, будем полагать, что данное ограничение на Δ_s выполнено.

Также будем полагать, что $\Delta_t \ll t_{\min}$.

5. Алгоритм построения клеточного индекса

В данном разделе вводится специальная структура данных — *клеточный индекс*, и описывается алгоритм для его построения.

Клеточный индекс для каждой ячейки пространства-времени содержит список объектов, которые находятся в данной ячейке или соседней с ней.

Иными словами, клеточный индекс реализует функцию

$$K(\hat{c}) = \{i | \exists j \in \{1, \dots, m\}, \hat{c} \in D(\hat{c}_j^i)\}$$

Отметим, что на практике данные обычно являются разреженными, т.е. существует значительное количество пространственно-временных ячеек, в которых отсутствуют точки траекторий. В связи с этим, для реализации клеточного индекса целесообразно использовать контейнер типа Ключ-Значение. Ключом поиска в этом массиве будет обобщенный индекс ячейки, а значением — множество идентификаторов объектов, которые посещали данную ячейку или соседние с ней.

Ниже приведено описание алгоритма построения клеточного индекса (5.1).

6. Похожие траектории и алгоритм их поиска

Положим $\sigma^i, \sigma^j \in O$ — две траектории. Далее, рассмотрим последовательности ячеек, полученные в результате квантизации:

$$\hat{\sigma}^i = (\hat{c}_1^i, \dots, \hat{c}_m^i),$$

Algorithm 5.1 Построение клеточного индекса — BuildIndex

```
1: function BUILDINDEX(  $\hat{O} = \{\hat{o}^1, \hat{o}^2, \dots, \hat{o}^n\}$  )
2:   Дано: множество квантизованных траекторий  $\hat{O}$ 
3:   Найти: контейнер типа Ключ-Значение,
4:   реализующий отображение  $K(\hat{c})$ 
5:   Создаем пустой контейнер типа Ключ-Значение
6:    $K \leftarrow []$ 
7:   пробегаем по всем траекториям
8:   for  $i=1, \dots, n$  do
9:     пробегаем по точкам траектории
10:    for  $j=1, \dots, m$  do
11:      добавляем номер объекта  $o^i$  во все ячейки из  $D(\hat{c}_j^i)$ 
12:      for  $dx=-2,0,2$  do
13:        for  $dy=-2,0,2$  do
14:          вычисляем обобщенный индекс соседней ячейки
15:           $\hat{c}_0 \leftarrow w(\hat{x}(\hat{c}_j^i) + dx, \hat{y}(\hat{c}_j^i) + dy, \hat{t}(\hat{c}_j^i) + dt)$ 
16:          добавляем  $o_i$  в множество объектов ячейки  $\hat{c}_0$ 
17:           $K[\hat{c}_0].insert(i)$ 
18:        end for
19:      end for
20:    end for
21:  end for
22:  return res;
23: end function
```

$$\hat{o}^j = (\hat{c}_1^j, \dots, \hat{c}_m^j).$$

Будем называть пару траекторий (o^i, o^j) *похожими*, если существуют номера $a, b \in \{1, \dots, m\}$, $a < b$, такие что

$$t(\hat{c}_b^i) - t(\hat{c}_a^i) \geq (t_{\min} - \Delta_t),$$

и для любого $k \in \{a, \dots, b\}$ имеет место $\hat{c}_k^i \in D(\hat{c}_k^j)$, где $D(\hat{c}_k^j)$ - множество ячеек, соседних с \hat{c}_k^j .

Пусть (o^i, o^j) - траектории-компаньоны. Следовательно, на всяком совместном участке $[t_u, t_v]$ в любой момент времени $t_r \in [t_u, t_v]$ имеет место:

$$d(p_r^i, p_r^j) \leq d_{\min}.$$

Ранее мы наложили ограничение на параметр квантизации:

$$\Delta_s = \frac{d_{\min}}{\sqrt{2}}.$$

Следовательно, $\hat{c}_r^i \in D(\hat{c}_r^j)$.

А так как продолжительность совместного участка у траекторий-компаньонов равна не менее, чем t_{\min} , то траектории (o^i, o^j) также будут являться похожими. Следовательно, всякие траектории-компаньоны являются похожими. Обратное в общем случае не верно.

Обозначим $S(\hat{o})$ — множество траекторий, похожих на \hat{o} .

Описанный ранее клеточный индекс позволяет построить эффективный алгоритм, который для заданной траектории находит множество всех похожих на нее траекторий.

На верхнем уровне данный алгоритм можно описать так: пробегаем скользящим окном (отрезком) $[a, b]$, таким что $(t(\hat{c}_b) - t(\hat{c}_a)) \geq t_{\min}$, по ячейкам квантизованной траектории $\hat{o} = (\hat{c}_1, \dots, \hat{c}_m)$. Для каждого положения отрезка $[a, b]$ с помощью клеточного индекса находим все объекты, которые присутствуют во всех ячейках $\hat{c}_a, \dots, \hat{c}_b$ и добавляем полученное множество в общий результат.

Заметим, что так как при построении клеточного индекса мы добавляли каждый объект не только в «свою» ячейку, но и во все соседние с ней, то все траектории, похожие на \hat{o} , будут включены в ответ.

Ниже приведено описание этого алгоритма (6.1).

Введем некоторые ограничения на множество траекторий и с их учетом дадим оценки сложности алгоритма поиска похожих траекторий.

Пусть $P, L \in \mathbb{R}$, а $p_0 = (x_0, y_0, t_0)$ — некоторая точка в пространстве-времени. Обозначим $U(p_0, P, L)$ — окрестность точки p_0 :

Algorithm 6.1 Поиск траекторий, похожих на заданную

```
1: function GETSIMILARТРАЈЕКТОРИЕС(  $\hat{o}, K$  )
2:   Дано: квантизованная траектория  $\hat{o} = (\hat{c}_1, \dots, \hat{c}_m)$ ,
3:   клеточный индекс  $K$ 
4:   Найти:  $S$  — множество траекторий, похожих на  $\hat{o}$ 
5:   Создаем пустое множество траекторий
6:    $S \leftarrow \emptyset, a \leftarrow 1, t_{sum} \leftarrow 0$ 
7:   указатель  $b$  (правый край отрезка) пробегает по всем точкам
8:   траектории  $\hat{o}$ 
9:   for  $b = 1, \dots, m$  do
10:      $t_{sum} \leftarrow t_{sum} + (t(\hat{c}_b) - t(\hat{c}_{b-1}))$ 
11:     нашли совместный участок достаточной продолжительности
12:     if  $t_{sum} \geq t_{min} - \Delta_t$  then
13:       пробуем переместить указатель  $a$  вперед
14:        $flag \leftarrow 1$ 
15:       while  $flag \& (t(\hat{c}_a) < t(\hat{c}_b))$  do
16:          $flag \leftarrow 0$ 
17:         if  $t_{sum} - (t(\hat{c}_{a+1}) - t(\hat{c}_a)) \geq t_{min} - \Delta_t$  then
18:            $t_{sum} \leftarrow t_{sum} - (t(\hat{c}_{a+1}) - t(\hat{c}_a))$ 
19:            $a \leftarrow a + 1$ 
20:            $flag \leftarrow 1$ 
21:         end if
22:       end while
23:       if  $a = b$  then
24:         break
25:       end if
26:       добавляем в  $S$  множество всех объектов,
27:       которые побывали во всех ячейках на отрезке  $[a, b]$ 
28:        $S \leftarrow S \cup \bigcap_{j=a}^b K(\hat{c}_j)$ 
29:     end if
30:   end for
31:   return  $S$ 
32: end function
```

$$U(p_0, P, L) = \{p = (x, y, t) \mid |x - x_0| \leq P, |y - y_0| \leq P, |t - t_0| \leq L\}.$$

Пусть $D \in \mathbb{R}$, $C \in N$. Будем говорить, что множество траекторий $O = \{o^1, o^2, \dots, o^n\}$ обладает (P, L, D, C) -свойством, если для всякой траектории $o^i = \langle p_1^i, \dots, p_m^i \rangle$ и для любых $a, b \in \{1, \dots, m\}$, таких что $(t_b^i - t_a^i) \geq D$ среди точек p_a^i, \dots, p_b^i найдется такая точка p_j^i , что суммарное число траекторий других объектов из O , точки которых содержатся в $U(p_j^i, P, L)$, не превосходит величину C .

Содержательно (P, L, D, C) -свойство означает, что для всякой траектории из O на всяком отрезке достаточной длины найдется точка траектории, в (P, L) -окрестности которой общее число различных объектов не превышает заданный порог C .

Заметим, что реальные данные (траектории автомобилей, пешеходов и др.) обладают (P, L, D, C) -свойством при определенных значениях параметров P, L, D, C .

Рассмотрим последовательность множеств траекторий O_1, \dots, O_k, \dots , каждое из которых обладает (P, L, D, C) -свойством. Положим, что при стремлении k к бесконечности в множествах O_k число траекторий n и число точек в траекториях m также стремятся к бесконечности. Имеет место следующее утверждение.

Теорема 1. *Если для параметров квантизации Δ_s, Δ_t выполнено: $\Delta_s < P, \Delta_t < L$ и $t_{min} \geq D$, то при стремлении k к бесконечности время работы алгоритма 6.1, примененного к произвольной квантизованной траектории из множества \hat{O}_k , не превосходит величину $O(m^2)$ в среднем.*

Доказательство. Заметим, что сложность алгоритма 6.1 не превосходит величину $O(m) \cdot (A + B)$. Здесь первый множитель $O(m)$ соответствует числу возможных различных положений скользящего окна $[a, b]$, величина A — сложность вычисления операции пересечения множеств $\bigcap_{j=a}^b K(\hat{c}_j)$, а величина B — сложность вычисления операции объединения.

Из наличия (P, L, D, C) -свойства у множества траекторий O_k следует, что для всякой траектории o^i и положения скользящего окна $[a, b]$ среди точек траектории p_a^i, \dots, p_b^i найдется точка p_j^i , в окрестности которой — $U(p_j^i, P, L)$ общее число различных объектов не превышает C . Из определения операции квантизации следует, что точка p_j^i будет преобразована в некоторую ячейку \hat{c}_j^i с размерами Δ_s, Δ_t . При этом сама точка p_j^i будет содержаться в ячейке \hat{c}_j^i . Так как $\Delta_s < P, \Delta_t < L$, следовательно ячейка \hat{c}_j^i целиком содержится в окрестности $U(p_j^i, P, L)$. А значит

и общее число различных объектов, содержащихся в ячейке \hat{c}_j^i не превосходит величину C . Далее, будем полагать, что клеточный индекс K представляет собой хеш-таблицу, элементы которой также представляют собой хеш-таблицы. Из этого следует, что $A = C \cdot (b - a) \cdot O(1) = O(m)$ в среднем. Здесь мы предполагаем что $b - a < m$ и для вычисления пересечения множеств достаточно проверить содержание элементов из множества $K(\hat{c}_j^i)$ во всех остальных множествах.

Сложность B , как легко видеть, линейным образом зависит от числа объектов в пересечении $\bigcap_{j=a}^b K(\hat{c}_j)$. Следовательно $B = O(m)$ в среднем.

В итоге получаем утверждение теоремы.

Перейдем к построению локального алгоритма поиска траекторий-компаньонов.

7. Локальный алгоритм поиска компаньонов

Ниже приведено описание локального алгоритма поиска траекторий-компаньонов. (7.1).

Необходимо отметить, что данный алгоритм осуществляет поиск точного решения задачи о компаньонах. Для каждой траектории сначала с помощью алгоритма 6.1 ищется множество всех похожих на нее траекторий, а затем с помощью последующей проверки предиката компаньонства 3.1 из найденного множества исключаются все траектории, которые не являются компаньонами.

Для последовательности множеств траекторий O_1, \dots, O_k, \dots имеет место следующее утверждение.

Теорема 2. *Если для параметров квантизации Δ_s, Δ_t выполнено: $\Delta_s < P, \Delta_t < L$ и $t_{min} \geq D$, то при стремлении k к бесконечности время работы алгоритма 7.1, примененного к множеству O_k , обладающему (P, L, D, C) -свойством, не превосходит величину $O(n \cdot m^2)$ в среднем.*

Здесь, как и ранее, n — число траекторий в O_k , m — число точек в каждой траектории. При стремлении k к бесконечности параметры n и m также стремятся к бесконечности.

Доказательство. Из описания алгоритма 7.1 видно, что его временная сложность равна $O(n \cdot (V + W \cdot F))$, где V — сложность работы алгоритма 6.1, а W — максимальное число траекторий, похожих на заданную, F — сложность вычисления предиката компаньонства.

Из теоремы [?] следует, что $V = O(m^2)$ в среднем.

Далее заметим, что из наличия у множества траекторий (P, L, D, C) -свойства следует, что общее число траекторий, похожих на заданную, не

Algorithm 7.1 Локальный алгоритм поиска траекторий-компаньонов

```
1: function LOCALSEARCH(  $O$  )
2:   Дано: множество траекторий  $O$ 
3:   Найти:  $W$  — множество пар-компаньонов
4:   строим множество квантизованных траекторий
5:    $\hat{O} \leftarrow Q(O)$ 
6:   строим клеточный индекс
7:    $K = BuildIndex(\hat{O})$ 
8:   создаем пустое множество для сохранения результата
9:    $W \leftarrow []$ 
10:  пробегаем по всем траекториям
11:  for  $i=1, \dots, n$  do
12:    ищем траектории, похожие на  $\hat{o}^i$ 
13:     $S^i \leftarrow GetSimilarTrajectories(\hat{o}^i, K)$ 
14:    for  $k = 1, \dots, |S^i|$  do
15:      проверяем, являются ли траектории компаньонами
16:       $g = TestPair(o^i, o^k)$ 
17:      if  $g=1$  then
18:        помещаем пару траекторий в результат
19:         $W.insert(o^i, o^k)$ 
20:      end if
21:    end for
22:  end for
23:  return  $W$ 
24: end function
```

может быть более, чем $C \cdot O(m)$, так как для всякого положения скользящего окна $[a, b]$ найдется ячейка, в которой содержится не более C объектов (см. алгоритм 6.1). Следовательно, $W = C \cdot O(m)$. Ранее (3) было показано, что $F = O(m)$.

Следовательно, сложность локального алгоритма поиска равна $O(m^2 \cdot n)$ в среднем. Теорема доказана.

Таким образом, алгоритм 7.1 имеет линейную сложность относительно числа объектов, что делает его ценным с практической точки зрения, так как обычно на практике число объектов n значительно превышает число точек в траектории — m .

8. Распределенный алгоритм поиска

В данном разделе описан распределенный алгоритм поиска пар траекторий-компаньонов. Описание алгоритма использует стандартную терминологию (RDD, map, reduce, экзекьютор и др.), широко применяемую в распределенных вычислительных системах [4], [5], [6].

Ранее, в разделе 4, была введена функция квантизации \hat{c} , которая ставит в соответствие каждой точке пространства-времени p целочисленный индекс ячейки. При этом, все ячейки имеют одинаковые размеры: Δ_s — по широте и долготе и Δ_t — по времени.

Можно считать, что функция \hat{c} задается параметрами Δ_s и Δ_t . То есть $\hat{c}(p) = \hat{c}(\Delta_s, \Delta_t, p)$.

Рассмотрим функцию $\hat{Z}(p) = \hat{c}(D_s, D_t, p)$, где параметры D_s и D_t принимают значения, значительно большие, чем Δ_s и Δ_t . Функция \hat{Z} разбивает пространство-время на макро-ячейки значительно большего размера. Данные макро-ячейки будем называть *блоками*.

Распределенный алгоритм основан на разбиении пространства-времени на блоки. Размеры блоков D_s и D_t являются конфигурируемыми и должны подбираться так, чтобы общий объем точек, попадающих в блок, был достаточно мал, чтобы полностью поместиться в оперативную память одного экзекьютора.

Распределенный алгоритм состоит из трех этапов:

- 1) разбиение исходных данных по блокам;
- 2) параллельная работа локальных алгоритмов поиска (7.1) на блоках;
- 3) сбор результатов работы локальных алгоритмов на центральной машине кластера.

Заметим, что при разбиении пространства-времени на блоки некоторые траектории-компаньоны могут оказаться потерянными. Это может

произойти, если, например, части совместного участка компаньонства располагаются в соседних блоках. При этом продолжительность каждой из этих частей не превышает параметр t_{min} . Данный недостаток может быть легко устранен за счет использования перекрывающихся блоков. При этом размер перекрытия по времени должен быть не менее, чем t_{min} , а размер перекрытия по каждой из координат должен быть не менее, чем $t_{min} \cdot V_{max}$, где V_{max} - максимально возможная скорость движения объекта. Далее, для простоты мы будем полагать, что блоки не перекрываются.

Отметим, что при фиксированном числе экзекьюторов временная сложность распределенного алгоритма асимптотически равна сложности локального алгоритма — $O(n \cdot m^2)$, где число траекторий — n и число точек в траекториях — m одновременно стремятся к бесконечности (см. раздел 7.1).

Ниже приведено детальное описание распределенного алгоритма (8.1).

Algorithm 8.1 Распределенный алгоритм поиска траекторий компаньонов

```

1: function DISTRIBUTEDSEARCH(  $O$  )
2:   Дано: распределенное множество точек траекторий  $O = RDD[p]$ 
3:   Найти:  $W$  — множество пар траекторий-компаньонов
4:   Каждой точке  $p$  ставим в соответствие пару  $(\hat{Z}(p), p)$ ,
5:   где  $\hat{Z}(p)$  - номер блока  $b$ , в котором находится точка  $p$ .
6:    $RDD[b, p] G \leftarrow O.map(p \Rightarrow (\hat{Z}(p), p))$ 
7:   Группируем точки по номерам блоков
8:    $RDD[b, O_b] B \leftarrow G.groupByKey$ 
9:   Здесь  $B$  представляет собой RDD-контейнер типа
10:  Ключ-Значение, где ключом выступает номер блока  $b$ ,
11:  а значением — множество точек, которые принадлежат
12:  блоку —  $O_b$ .
13:  Заметим, что после операции groupByKey каждое множество  $O_b$ 
14:  будет гарантированно размещено целиком на одном экзекьюторе.
15:  Далее, для каждого блока запускаем локальный алгоритм (7.1)
16:   $RDD[(o^i, o^j)] H \leftarrow B.map((b, O_b) \Rightarrow LocalSearch(O_b))$ 
17:  Устраняем возможные повторения и собираем результат
18:  на управляющей машине кластера
19:   $W \leftarrow H.distinct.collect$ 
20:  return  $W$ 
21: end function

```

Список литературы

- [1] K. Zheng, Y. Zheng, N. Yuan, S. Shang and X. Zhou, “Online Discovery of Gathering Patterns over Trajectories”, *IEEE Transactions on Knowledge and Data Engineering*, **26**:8 (2014), 1974–1988.
- [2] L. Tang, Y. Zheng, J. Yuan, J. Han, A. Leung, C. Hung and W. Peng, “On Discovery of Traveling Companions from Streaming Trajectories”, IEEE 28th International Conference on Data Engineering, 2012, 186–197.
- [3] Z. Li, B. Ding, J. Han and R. Kays, “Swarm: Mining Relaxed Temporal Moving Object Clusters”, *PVLDB*, **3**:1 (2010), 723–734.
- [4] “Programming model "map-reduce"”, <https://en.wikipedia.org/wiki/MapReduce>.
- [5] “Apache Spark Framework”, <https://spark.apache.org/>.
- [6] B.Chambers, M.Zaharia, *Spark: The Definitive Guide: Big Data Processing Made Simple*, 2018.
- [7] “Equirectangular Projection”, https://en.wikipedia.org/wiki/Equirectangular_projection.

References

- [1] K. Zheng, Y. Zheng, N. Yuan, S. Shang and X. Zhou, “Online Discovery of Gathering Patterns over Trajectories”, *IEEE Transactions on Knowledge and Data Engineering*, **26**:8 (2014), 1974–1988.
- [2] L. Tang, Y. Zheng, J. Yuan, J. Han, A. Leung, C. Hung and W. Peng, “On Discovery of Traveling Companions from Streaming Trajectories”, IEEE 28th International Conference on Data Engineering, 2012, 186–197.
- [3] Z. Li, B. Ding, J. Han and R. Kays, “Swarm: Mining Relaxed Temporal Moving Object Clusters”, *PVLDB*, **3**:1 (2010), 723–734.
- [4] “Programming model "map-reduce"”, <https://en.wikipedia.org/wiki/MapReduce>.
- [5] “Apache Spark Framework”, <https://spark.apache.org/>.
- [6] B.Chambers, M.Zaharia, *Spark: The Definitive Guide: Big Data Processing Made Simple*, 2018.
- [7] “Equirectangular Projection”, https://en.wikipedia.org/wiki/Equirectangular_projection.

Distributed companion trajectory search algorithm

Sokolov A.P., Aliseychik P.A., Moiseev S.V.

The work is focused on development of the distributed algorithm for search of companion-trajectories. We call two trajectories as companion-trajectories if they both contain section of required duration on which at any time moment distance between two objects is no larger than specified threshold. It’s assumed that search is performed on database that is so big that it cannot be stored and processed on a single work station. Such tasks are usually called big-data tasks. We define trajectory quantization procedure, introduce cell index data structure, define similar trajectories. Then we build effective algorithm for search of similar trajectories. Based on this algorithm we develop effective

local and distributed algorithms for search of companion trajectories. Asymptotic complexity of these algorithms is estimated.

Keywords: big-data, geo-spatial data analysis, trajectory analysis, companion-trajectory search, similar trajectory search, traffic analysis.

Часть 3.
Математические модели

Локально восстанавливаемые коды на графах

Валинуров Д.Ю.¹

Локально восстанавливаемые коды (LRC коды) это линейные коды с представляющим большой интерес для приложений свойством, что каждый символ кодового слова можно восстановить по небольшому множеству других символов. Символы кода можно интерпретировать как серверы с некоторой информацией. Тогда становится естественным задание топологии этих серверов в виде графа, в котором для каждого сервера указаны сервера, по данным из которых можно однозначно восстановить его данные. В статье рассматриваются LRC коды для некоторых типов топологий и приводятся оценки на скорость таких кодов.

Ключевые слова: коды исправляющие ошибки, локально восстанавливаемые коды, коды на графах.

1. Введение

С распространением систем хранения информации появилась необходимость в восстановлении стёртой информации с затратой минимального количества ресурсов. Одним из направлений исследований стали LRC коды, в которых стёртый символ может восстанавливаться по небольшому количеству других символов. Если представить систему хранения информации как множество датацентров, объединённых в некоторую топологию, то естественной становится задача восстановления потерянной информации с учётом этой топологии. В статье представлены оценки на скорость кода в произвольной топологии, заданной графом, и конструкция кода с субпакетизацией. Рассмотрена достижимость построенным кодом оценок на скорость кода в простейших топологиях — цикле и торе.

Обозначим через \mathbb{F}_q конечное поле из q элементов. Назовём (n, k) кодом над алфавитом A подмножество $C \subseteq A^n$ мощности $|C| = A^k$. Далее

¹Валинуров Денис Юрьевич — аспирант каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: denis.valinurov@yandex.ru.

Valinurov Denis Yurevich — graduate student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

будем рассматривать *линейные* $[n, k]$ коды над конечным полем \mathbb{F}_q — k -мерное линейное подпространство \mathbb{F}_q^n , где k является размерностью кода $\dim(C)$. *Скоростью* (n, k) кода назовём величину $R = \frac{k}{n}$.

Определение 1. Минимальным расстоянием кода C называется величина $d = \min_{x, y \in C, x \neq y} h(x, y)$, где h — расстояние Хэмминга, то есть количество компонент, в которых векторы не равны. Нетрудно показать, что для линейных кодов $d = \min_{x \in C, x \neq 0} w(x)$, где $w(x)$ — вес слова x , то есть количество ненулевых компонент.

Из минимального расстояния получается такая характеристика кода, как корректирующая способность. Код с минимальным расстоянием d может исправить $d - 1$ стираний.

Для линейных кодов известна оценка Синглтона $d \leq n - k + 1$. Коды, на которых достигается равенство в этой оценке, называются *разделимыми кодами с максимальным расстоянием* или *МДР кодами*. Наиболее известным примером МДР кодов являются коды Рида-Соломона [2].

Обозначим за $[n]$ множество $\{1, 2, \dots, n\}$. Ограничением $C|_A$ кода C на множество координат $A \subseteq [n]$ назовём код, полученный из C удалением всех координат не из A .

Определение 2. Говорим, что (n, k) код обладает свойством r -локальности, если выполняется следующее: для любого $i \in [n]$ существует подмножество $R_i \subseteq [n] \setminus i$, $|R_i| \leq r$ такое, что ограничения множества $C(i, a) = \{x \in C : x_i = a\}$ на R_i имеют пустое пересечение для $a \neq a'$, то есть $C|_{R_i}(i, a) \cap C|_{R_i}(i, a') = \emptyset$.

Линейный код с таким свойством называется LRC $[n, k, r]$ кодом (locally recoverable code). Множества R_i будем называть *локальностями*. Из определения видно, что i -ый символ не может принимать разные значения при одинаковых значениях символов из R_i . Поэтому можно говорить, что символ i однозначно восстанавливается по множеству R_i и является функцией символов $\alpha_1, \alpha_2, \dots, \alpha_r \in R_i$. Поэтому локальные коды обладают важным свойством, что каждый символ можно восстановить по небольшому количеству других символов, что позволяет избежать передач больших объемов данных для восстановления.

В статье [3] доказаны основные оценки на скорость R и минимальное расстояние d для LRC кода: $d \leq n - k - \lceil \frac{k}{r} \rceil + 2$, $R \leq \frac{r}{r+1}$. LRC код называется *оптимальным*, если он достигает равенства в оценке на минимальное расстояние: $d = n - k - \lceil \frac{k}{r} \rceil + 2$.

Обычно сервера связаны в сеть, называемую топологией и задаваемую некоторым графом. Обозначим $G(V, E)$ — граф с множеством вер-

шин V и множеством рёбер E . Для произвольного графа $G(V, E)$ и произвольной вершины v в этом графе обозначим

$$N(v) = \{x : x \in V, (v, x) \in E\}.$$

Если S — множество, то примем $N(S) = \bigcup_{s \in S} N(s)$.

Определение 3. Пусть G — произвольный граф без петель и кратных рёбер с занумерованным множеством вершин $V = \{1, \dots, n\}$. Говорим, что $[n, k, r]$ LRC код C является кодом для топологии задаваемой графом G , если для каждого i выполнено $N(i) = R_i$, то есть соседние вершины в графе G являются одновременно локальностью для этой вершины в коде C (для неориентированных графов считаем, что ребро идёт в обе стороны).

Коды над топологиями слабо изучены в литературе, но многие оценки на эти коды следуют из оценок для произвольных LRC кодов.

В статье [6] представлена торическая топология локальных кодов. Кодовым словом является матрица $m \times n$. Символы можно восстанавливать локально по строке и по столбцу, то есть если в строке происходит не более a стираний, то их можно восстановить используя только символы этой строки с помощью $[n, n - a]$ кода C_{row} . Аналогично, если в столбце происходит не более b стираний, то их можно восстановить локально с помощью $[m, m - b]$ кода C_{col} .

В статье [8] представлены конструкции кодов для произвольных графов. Для неориентированных графов находится максимальное паросочетание, и информация в вершинах на концах одного ребра этого паросочетания дублируется. Для ориентированных графов дублируется информация во всех вершинах каждого цикла.

Определение 4. Говорим, что код C для топологии $G(V = \{1, \dots, n\}, E)$ исправляет локально e стираний, если для $x = (x_1, \dots, x_n) \in C$ и произвольного $U \subset V, |U| \leq e$ выполнено следующее условие: для любого $i \in U$ существует функция f_i^U такая что $x_i = f_i^U(\{x_j : j \in \bigcup_{l \in U} N(l) \setminus U\})$.

LRC коды с несколькими стираниями в литературе называются ME-LRC кодами, оценки на скорость и расстояние таких кодов изучаются в [7]. Также в литературе встречаются t-LRC коды, где для каждого символа существует t непересекающихся локальностей, по которым этот символ может быть независимо восстановлен. Ясно, что t-LRC коды также могут локально восстанавливать любые t стираний. Оценки на t-LRC коды были изучены в [4].

В дальнейшем помимо линейных кодов будем рассматривать коды над алфавитом $A = \mathbb{F}_q^w$. Каждый символ такого кода будем интерпретировать как *сервер*, хранящий вектор из w значений из \mathbb{F}_q . В литературе параметр w называется субпакетизацией. Обозначим общую длину такого кода как $N = nw$, размерность – $K = kw$. Такие коды конкатенацией векторов для каждого сервера могут быть сведены к линейным кодам. Если на множестве серверов такого кода задана локальность, то соответствующий r -локальный код будем называть $[n, k, r, w]$ кодом.

Определение 5. Введём расстояние между вершинами графа: $\text{dist}(v_1, v_2)$, как минимальное количество рёбер в пути между вершинами v_1 и v_2 . Назовём r -локальностью вершины v в графе G с функцией расстояния dist множество $N(v, r) = \{v' : \text{dist}(v, v') \leq r\}$.

Таким образом, далее будем считать, что каждая вершина графа по умолчанию хранит символ из \mathbb{F}_q , если же задан параметр w , то хранит вектор из \mathbb{F}_q^w . А рёбра графа задают локальности, то есть информация в вершине может быть однозначно восстановлена по информации из смежных вершин.

2. Оценки на код для произвольных топологий

В этом разделе приведём утверждения, верные для локальных кодов над произвольными графами. Как следствие теоремы 2.1.1 из статьи [3] получим следующее утверждение.

Утверждение 1. Пусть $G(V, E)$ – ориентированный граф, C – код для топологии G , $S \subseteq V$ – подмножество вершин G такое, что подграф, натянутый на вершины $V \setminus S$, является ациклическим. Тогда имеем неравенство на размерность кода: $\dim C \leq |S|$.

Доказательство. Рассмотрим подграф, натянутый на вершины $V \setminus S$. Из ацикличности следует, что в подграфе существует вершина v без исходящих рёбер. То есть локальность этой вершины целиком лежит в S , поэтому данные вершины v однозначно восстанавливаются по данным вершин из множества S .

Подграф, натянутый на вершины $V \setminus (S \cup \{v\})$, также является ациклическим, поэтому далее действуем аналогично. В итоге получается, что все вершины из $V \setminus S$ однозначно восстанавливаются по вершинам S , откуда получаем $\dim C \leq |S|$. \square

Определение 6. Разбиением неориентированного графа G на непересекающиеся клики назовём разбиение множества вершин V на непе-

ресекающиеся подмножества A_1, \dots, A_l такие, что $\bigcup_{i \in [l]} A_i = V$ и для каждого i подграф, натянутый на вершины A_i , является полным.

Лемма 1. Пусть $G(V, E)$ — неориентированный граф, A_1, \dots, A_l — его разбиение на непересекающиеся клики. Тогда существует код C для топологии G , исправляющий локально e стираний, для которого выполнено $\dim C \geq |V| - e \cdot l$

Доказательство. В клике любые две вершины соединены ребром, поэтому на вершинах клики A_i мы можем построить $[|A_i|, |A_i| - e]$ МДР код C_i , который будет исправлять любые e стираний. Таким образом, если e стираний попадают в одну клику, то мы можем локально восстановить все стирания, используя код C_i . Если e стираний попадают в разные клики, то мы можем в каждой клике локально восстановить попавшие в эту клику стирания с помощью соответствующего кода.

Объединим все проверочные соотношения кодов C_i и получим проверочные соотношения кода C . Для кода C будет выполнено $\dim C = \sum_{i \in [l]} \dim C_i = \sum_{i \in [l]} (|A_i| - e) = |V| - e \cdot l$ \square

Определение 7. Набором независимых подмножеств I множества вершин V графа G назовём множества $A_1, \dots, A_l \subseteq V$ такие, что для любых $i \neq j$ и любых $x \in A_i, y \in A_j$: x и y не соединены ребром. Мощностью набора назовём число $|I| = \bigcup_{i \in [l]} |A_i|$. Множеством соседних вершин набора назовём множество $N(I) = \bigcup_{i \in [l]} N(A_i) \setminus \bigcup_{i \in [l]} A_i$. Расстоянием набора назовём число $D(I) = \max_{i \in [l]} |A_i|$.

Лемма 2. Пусть $G(V, E)$ — неориентированный граф, C — код для топологии G , исправляющий локально e стираний. Рассмотрим I — набор независимых подмножеств графа G с $D(I) \leq e$. Тогда $\dim C \leq |V| - |I|$.

Доказательство. Для всех i выполнено $(N(A_i) \setminus A_i) \subseteq (V \setminus \bigcup_{i \in [l]} A_i)$. Поэтому из свойства восстановления e стираний по локальным соотношениям получается, что все вершины из $\bigcup_{i \in [l]} A_i$ можно восстановить по вершинам из $V \setminus \bigcup_{i \in [l]} A_i$, откуда следует искомое неравенство. \square

Замечание 1. Для более точной оценки в лемме 3 необходимо находить наборы подмножеств с максимальной суммой $\sum_{i=1}^l |A_i|$. В случае $e = 1$ задача состоит в нахождении максимального независимого множества, что эквивалентно нахождению минимального покрытия вершин и является NP-полной задачей.

Как обобщение теоремы 14 из [8] и как следствие теоремы 2.1.2 из [3] может быть получено следующее утверждение.

Утверждение 2. Пусть $C = [n, k, d]$ код для топологии $G(V, E)$, исправляющий локально e стираний. I – набор независимых подмножеств $G(V, E)$ такой, что $|N(I)| \leq k - 1$ и $D(I) \leq e$. Тогда $d \leq n - k + 1 - |I|$.

Доказательство. Пусть $R = (\bigcup_{A \in I} A) \cup N(I) \cup R'$, где R' – множество $k - 1 - |N(I)|$ произвольных вершин из $V \setminus N(I)$. Рассмотрим код $C|_R$, свойство восстанавливать локально e стираний у него остаётся, так как можно подставить в выколотые координаты кода C нули. Отсюда по лемме 3 получаем, что $\dim C|_R \leq k - 1$. То есть $\dim C|_R \neq \dim C$ и код $C|_R$ содержит нулевой вектор, поэтому $d \leq n - |R| \leq n - k + 1 - |I|$. \square

3. Конструкция для графов с функцией расстояния

В этом разделе приведём конструкцию исправляющего локально e стираний $[n, k, r, w]$ кода в виде композиции МДР кодов для произвольного неориентированного графа с заданной функцией расстояния. Под стиранием понимаем стирание всех w символов из одной вершины.

Определение 8. Граф $G(V, E)$ называется (вершинно) транзитивным, если для любых двух вершин $v_1, v_2 \in V$ существует сохраняющее смежность биективное отображение множества вершин на себя $f: V \rightarrow V$ такое, что $f(v_1) = v_2$.

У транзитивного графа окрестности каждой вершины устроены одинаково, и поэтому $|N(v_1, s)| = |N(v_2, s)|$ для любых вершин v_1, v_2 и расстояний s . Приведём конструкцию для случая транзитивного графа $G(V = \{1, \dots, n\}, E)$ и $T|w$, где $T = |N(1, \frac{r}{2})|$. Но конструкцию можно рассматривать и для произвольного графа, что будет показано далее.

Построение 1. Будем строить код C длины $N = nw$. Обозначим $\omega = \frac{w}{T}$. Для каждой вершины разобьём множество из w символов, входящих в эту вершину, на T частей, в каждой из которых ω символов. Таким образом, кодовое слово нашего кода C будет состоять из nT частей (каждая по ω символов), которые мы проиндексирuem множеством $[n] \times [T]$.

Зафиксируем некоторый $[w, w - ew]$ МДР код \mathfrak{C} , исправляющий ew стираний. Далее для каждого $i \in [n]$ выберем из всех вершин из $N(i, \frac{r}{2})$ по одной части, не выбранной ранее, и обозначим через $V_i \subseteq [nw]$ – соответствующее выбранным частям подмножество символов.

Добавим к коду C проверочные соотношения кода \mathfrak{C} , построенного на выбранных символах, т.е. потребуем чтобы каждое слово $c \in C$ удовлетворяло условию $c|_{V_i} \in \mathfrak{C}$. Для каждого $i \in [n]$ будет добавлено $e\omega$ проверочных соотношений, и все эти проверочные соотношения будут линейно независимы, так как будут являться проверочными соотношениями МДР кода, построенного на непересекающихся множествах символов. Поэтому код C будет иметь размерность $K = n(w - e\omega)$.

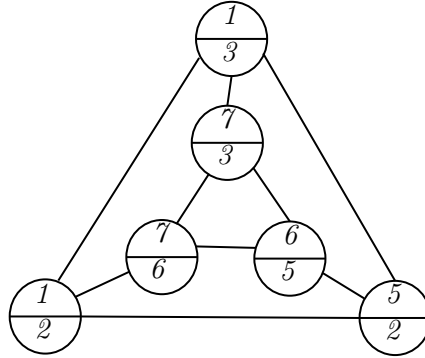
Теорема 1. Код C из построения 1 является $[n, k, r, w]$ кодом для транзитивного графа $G(V, E)$ с заданной функцией расстояния, исправляющего локально e стираний, и его скорость $R = 1 - e/T$, где $T = |N(v, \frac{r}{2})|, v \in V$.

Доказательство. Код \mathfrak{C} имеет размерность $K = n(w - e\omega)$, откуда следует оценка на скорость $R = K/N = 1 - e/T$.

Нетрудно видеть, что вершины на расстоянии более $\frac{r}{2} + \frac{r}{2} = r$ не участвуют в одном проверочном соотношении, из чего следует r -локальность.

Выберем произвольные e вершин, без ограничения общности пусть это вершины $\{1, \dots, e\}$, и докажем, что их можно восстановить по локальностям. Рассмотрим множество $S = \{V(i) : \exists v \in [e], i \in N(v, \frac{r}{2})\}$. Каждая часть любой из e стёртых вершин входит в некоторый $V(j) \in S$, поэтому эта часть может быть восстановлена по соответствующим проверочным соотношениям $C|_{V(j)}$, так как \mathfrak{C} является $[w, w - e\omega]$ МДР кодом и может исправить стирание вплоть до e частей по ω символов в каждой. \square

Замечание 2. Конструкция лучше работает для чётных r , например, для $r = 1$ конструкция вообще не работает. В общем случае для построения V_i не обязательно выбирать вершины из множества $N(i, \frac{r}{2})$. Достаточно требовать, чтобы расстояние между любыми двумя выбранными вершинами не превосходило r . Например, для $r = 1$ нетрудно построить модифицированную конструкцию, которую покажем на следующем примере.



Над этим графом строим код с параметрами $n = 6, r = 1, w = 2$. Каждую вершину содержит два символа, и в двух соседних вершинах два символа являются копией друг друга. Одинаковыми цифрами обозначены символы, которые являются копией друг друга. Скорость полученного кода равна $1/2$.

Замечание 3. Также конструкцию можно обобщить на случай нетранзитивных графов. Для таких графов будем делить множество символов в каждой вершине на разное количество частей, а именно вершину i поделим на $|N(i, \frac{r}{2})|$ частей. Части, на которые делится множество символов в вершине, также будем считать различными по размеру. Проиндексируем полученные части как ранее. Пусть множество $K(i, j)$ будет содержать те символы кода, которые входят в часть j вершины $i \in [n]$, где $1 \leq j \leq |N(i, \frac{r}{2})|$.

Тогда для $K(i, j)$ будут выполняться следующие равенства:

$$\bigcup_{1 \leq j \leq |N(i, \frac{r}{2})|} K(i, j) = \{iw + 1, \dots, (i + 1)w\} \text{ и } K(i_1, j_1) \cap K(i_2, j_2) = \emptyset \text{ для}$$

всех $i, i_1, i_2 \in [n]$ и $(i_1, j_1) \neq (i_2, j_2)$. То есть все части не пересекаются, и объединение всех частей одной вершины составляет в точности множество символов, содержащихся в этой вершине.

В конструкции для нетранзитивных графов вместо одного МДР кода \mathfrak{C} будем рассматривать разные МДР коды \mathfrak{C}_i , каждый из которых будет построен на координатах $V(i)$ в обозначениях выше. Будем требовать, чтобы каждое слово $s \in \mathfrak{C}$ удовлетворяло условию $s|_{V_i} \in \mathfrak{C}_i$.

Обозначим

$$n(i) = |V(i)|,$$

$$h(i) = \max_{\substack{(v_1, j_1), (v_2, j_2), \dots, (v_e, j_e), \\ \forall t \in [e]: K(v_t, j_t) \subset V(i)}} \sum_{t \in [e]} |K(v_t, j_t)|.$$

То есть величина $h(i)$ равна максимальному размеру объединения произвольных e частей, входящих в $V(i)$. Тогда \mathfrak{C}_i будет строиться как

$[n(i), n(i) - h(i)]$ МДР код на координатах $V(i)$ и будет исправлять стирание любого множества e входящих в него частей серверов, так как максимальный размер такого множества равен $h(i)$. Поэтому теорема 1 также будет верна, но для меньшего значения скорости, а именно $R = \sum_{i \in [n]} \frac{n(i) - h(i)}{nw}$.

Для уменьшения избыточности можно требовать, например, чтобы значение $h(i)/n(i)$ было в точности равно $e/|N(i, \frac{r}{2})|$. В таком случае все части, входящие в $V(i)$, должны иметь одинаковый размер. Такое разбиение на части не всегда существует, что будет видно далее. Допустим, что для кода \mathfrak{C}_i из каждой вершины $v \in N(i, \frac{r}{2})$ выбрали части с одинаковыми размерами ω_i (ω_i могут быть различны для $i \in [n]$). Тогда должно выполняться: $\sum_{j \in N(i, \frac{r}{2})} \omega_j = w$.

В итоге, для нахождения размерности кода получаем следующую задачу целочисленного линейного программирования, где максимизируемым значением является размерность искомого кода:

$$\begin{cases} nw - \sum_{i \in [n]} \omega_i e \rightarrow \max; \\ \sum_{j \in N(i, \frac{r}{2})} \omega_j = w, \quad \forall i \in [n]; \\ \omega_i \geq 0, \quad \forall i \in [n]. \end{cases}$$

4. Примеры LRC на графах

В этом разделе рассмотрим простейшие топологии: цикл и тор с заданными на них функциями расстояний, то есть локальностями вершины v будут множества $N(v, r)$. Для этих топологий построим коды с исправлением e стираний. Будем сравнивать параметры кодов для различных w с оценками для LRC с $w = 1$, что в общем случае является неверным, так как используя субпакетизацию также можно улучшать различные параметры. Построенная ранее конструкция достигает простейшую нижнюю оценку на скорость кода в рассматриваемых топологиях, но применима и для произвольного графа.

Определение 9. Предельной скоростью семейства $[n, k, r, w]$ кодов над семейством графов $\{G_n\}_{n=1}^{\infty}$ назовём величину $\hat{R}(r, e, w) = \sup_{n \rightarrow \infty} R(G_n, e)$, где $R(G_n, e)$ — наибольшая скорость $[n, k, r, w]$ кода над графом G_n , который может исправлять стирания e серверов.

Предельной скоростью кода конструкции 1 назовём величину $\hat{R}'(r, e, n) = \sup_{w \rightarrow \infty} R'(G_n, w, e)$, где $R'(G_n, w, e)$ — скорость кода конструкции 1, исправляющего e стираний, над графом G_n с субпакетизацией w .

4.1. Циклический граф

Циклическим графом Z_n назовём граф с n вершинами на рисунке 3. Зададим функцию расстояния и r -локальность согласно определению 5.

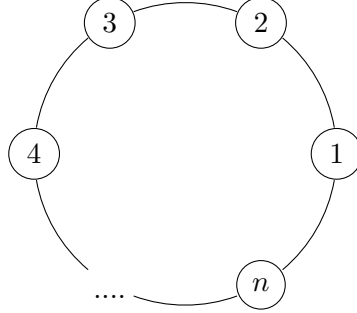


Рис. 1. Циклический граф локальности.

Для LRC с одним стиранием есть общая оценка на скорость кода из статьи [3]: $R \leq 1 - \frac{1}{2r+1}$, так как локальность в обычном смысле равна $2r$. Далее покажем, что оптимальная скорость для циклического графа меньше этого значения.

Утверждение 3. Для циклического графа Z_n наибольшая скорость LRC кода, исправляющего e ошибок, удовлетворяет следующим неравенствам:

$$1 - e \times \lceil \frac{n}{r+1} \rceil / n \leq R(Z_n, e) \leq 1 - e \times \lfloor \frac{n}{r+e} \rfloor / n.$$

Доказательство. Построим граф G , вершинами которого будут являться вершины Z_n , а рёбрами будут соединены все пары вершин, расстояние между которыми не превышает r .

Выберем в качестве набора независимых подмножеств графа G для леммы 2 множество $M = \{(r+e)i+j : 1 \leq j \leq e\} : 0 \leq i \leq \lfloor \frac{n}{r+e} \rfloor\}$. Тогда по лемме 2: $R \leq 1 - |M|/n = 1 - e \times \lfloor \frac{n}{r+e} \rfloor / n$.

Подряд идущие $r+1$ вершины графа G образуют клику, поэтому из леммы 1 получаем $R \geq 1 - e \times \lceil \frac{n}{r+1} \rceil / n$. \square

Переходя в утверждении 3 к пределу при $n \rightarrow \infty$ получим следующую теорему.

Теорема 2. Предельная скорость $[n, k, r, w]$ кода над семейством $\{Z_n\}$ удовлетворяет следующим неравенствам: $1 - \frac{e}{r+1} \leq \tilde{R}(r, e, w) \leq 1 - \frac{e}{r+e}$.

Доказательство. Переход к $n \rightarrow \infty$ в утверждении 3. \square

Естественно требовать, чтобы размер локальности каждой вершины не превышал общего количества вершин. Следующее утверждение верно для $2r+1 \leq n$.

Утверждение 4. *Предельная скорость кода конструкции 1 для циклического графа равна $\hat{R}(r, 1, n) = \hat{R}(r, 1, w) = 1 - \frac{1}{r+1}$.*

Доказательство. Следуя построению 1 и замечанию 2, для каждой вершины $i \in [n]$ выберем множество из $r + 1$ последовательных вершин $\{i - \lfloor \frac{r}{2} \rfloor, \dots, i + \lceil \frac{r}{2} \rceil\}$ (считается, что вершины зациклены). Если $r + 1 | w$, то разобьём выбранные вершины на части размера $\omega_i = \frac{w}{r+1}$, иначе по замечанию 3 размеры частей можно выбирать равными $\omega_i = \lfloor \frac{w}{r+1} \rfloor$ и $\omega_i = \lceil \frac{w}{r+1} \rceil$. Далее возьмём часть размера ω_i с каждой выбранной вершины и построим на этих частях проверочное соотношение. Возьмём все проверочные соотношения для каждого $i \in [n]$. Размерность итогового кода $K = nw - \lceil \frac{w}{r+1} \rceil n$, откуда $R = 1 - \lceil \frac{w}{r+1} \rceil / w$. При $w \rightarrow \infty$ получаем искомое. \square

Замечание 4. *Следующий тривиальный пример удовлетворяющий оценке леммы 2 показывает, что скорость может быть больше предельной скорости. Рассмотрим следующий код, заданный проверочной матрицей:*

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Он имеет параметры $n = 3, w = 2, r = 1$, и его скорость $R = 2/3$, тогда как предельная скорость равняется $1/2$.

4.2. Торическая топология

В этом разделе рассмотрим коды для торических топологий с $e = 1$.

Будем называть l -мерным тором решётку $n_1 \times n_2 \times \dots \times n_l$ с зацикленными краями. Код C для двумерного тора можно получить как прямое произведение 2 кодов для циклических графов с длинами n_1, n_2 . Если рассмотреть кодовые слова в виде матрицы $n_1 \times n_2$, каждая ячейка матрицы имеет две локальности – по строке и по столбцу, так как каждая строка и каждый столбец является кодовым словом кода над циклическим графом. Таким образом, скорость полученного кода равна $(\frac{r}{r+1})^2$. Далее в этом разделе будем рассматривать коды для l -мерного тора, в которых локальность задаётся множеством $N(v, r)$ из определения 5.

Лемма 3. *На l -мерном торе в шар радиуса r с центром в одной из вершин v входит $S(l, r) \sim \frac{2^l r^l}{l!}$ вершин при $r \rightarrow \infty$.*

Доказательство. Для $l = 1$ утверждение очевидно. Докажем по индукции для произвольного l .

Пусть x_1, \dots, x_l - измерения l -мерной решётки, центр шара расположим в точке $(0, \dots, 0)$. Нужно оценить количество целочисленных решений неравенства $|x_1| + \dots + |x_l| \leq r$. Неравенство $x_1 + \dots + x_l \leq r$ по правилам комбинаторики имеет C_{r+l}^r неотрицательных целочисленных решений. Ровно столько вершин шара с неотрицательными координатами. Аналогичное количество получится если какие-то x_i будут неположительны. Всего есть 2^l комбинаций выбора какие координаты будут неположительны, а какие неотрицательны. Тогда если просуммировать C_{r+l}^r для каждой комбинации, то вершины, у которых некоторые координаты равны нулю, будут учтены дважды, а значит $S(l, r) < 2^l C_{r+l}^r$.

С другой стороны, по формуле включений-исключений с округлением до второго слагаемого при $r \rightarrow \infty$ получаем: $S(l, r) > 2^l C_{r+l}^r - lS(l-1, r) = 2^l C_{r+l}^r + o(r^l) = \frac{2^l r^l}{l!} + o(r^l)$, так как $S(l-1, r) = O(r^{l-1})$ по индукции. \square

Лемма 4. На l -мерном торе при $n_1, \dots, n_l, r \rightarrow \infty$:

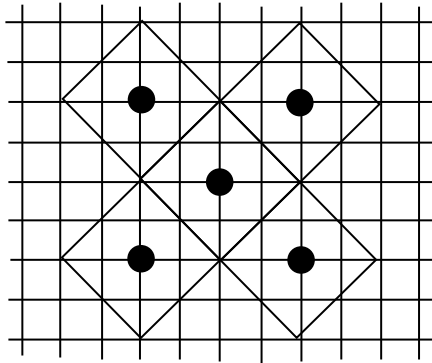
1) Существует множество вершин M_1 такое, что расстояние между любыми двумя вершинами этого множества превышает r и

$$|M_1| = l! \prod_{i=1}^l n_i / (r^l (1 + o(1))) + o\left(\prod_{i=1}^l n_i\right).$$

2) Существует разбиение всех вершин тора на множество M_2 подмножеств таких, что расстояние между любыми двумя вершинами

одного подмножества не превышает r и $|M_2| = l! \prod_{i=1}^l n_i / (r^l (1 + o(1))) + o\left(\prod_{i=1}^l n_i\right)$.

Доказательство. Пусть n_1, \dots, n_l достаточно большие. Отступим от границ тора на расстояние r и будем покрывать оставшуюся область шарами в метрике Манхэттена такими, что соседние шары имеют общую грань. Ниже показан пример таких шаров радиуса 2 для $l = 2$.



Отношение области, на которую мы отступили от краёв и которую мы не замостим полностью, к $\prod_{i=1}^l n_i$ стремится к нулю, поэтому размер этой области $o(\prod_{i=1}^l n_i)$.

1) Если выбрать радиусы шаров для замощения $\lceil \frac{r+1}{2} \rceil$, то расстояние между центрами любых шаров будет превосходить r . По лемме 3 размер шара радиуса $\lceil \frac{r+1}{2} \rceil$ равен $2^l \lceil \frac{r+1}{2} \rceil^l / l! + o(r^l) = \frac{r^l}{l!} + o(r^l)$. Значит, всего шаров в покрытии будет не менее $l! \prod_{i=1}^l n_i / (r^l + o(r^l)) + o(\prod_{i=1}^l n_i)$, так как шары пересекаются. Таким образом, центры шаров являются множеством M_1 из первого утверждения леммы.

2) Если выбрать радиусы шаров для замощения $\lfloor \frac{r}{2} \rfloor$, то все точки внутри одного шара будут находиться на расстоянии не превышающем r друг от друга. По лемме 3 размер шара также будет равен $2^l \lfloor \frac{r}{2} \rfloor^l / l! + o(r^l) = \frac{r^l}{l!} + o(r^l)$. Учитывая, что размер пересечения шаров равняется $o(r^l)$ при $r \rightarrow \infty$, получаем, что шаров не более $l! \prod_{i=1}^l n_i / (r^l + o(r^l)) + o(\prod_{i=1}^l n_i)$. Примем как подмножества из второго утверждения леммы рассмотренные шары и одноэлементные множества в области на краях тора. Точки в пересечении двух шаров отнесём к любому из этих шаров. Тогда получим множество M_2 из второго утверждения леммы. \square

Теорема 3. Для предельной скорости $[n_1 \times \dots \times n_l, k, r, w]$ LRC кода над семейством l -мерных торов при $r \rightarrow \infty$ верно:

$$\tilde{R}(r, 1, w) = 1 - l! / (r^l (1 + o(1))).$$

Доказательство. Построим граф G , вершинами которого будут являться вершины тора, а рёбрами будут соединены все пары вершин, расстояние между которыми не превышает r .

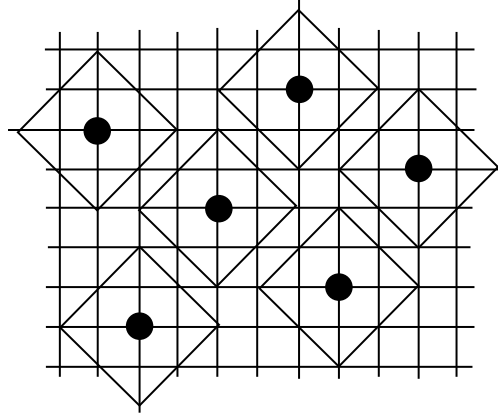
Из первого утверждения леммы 4 и леммы 2 для построенного графа G при $n_1, \dots, n_l, r \rightarrow \infty$ получаем $\tilde{R}(r, 1, w) \leq 1 - l! / (r^l (1 + o(1)))$. Из второго утверждения леммы 4 и леммы 1 для графа G получаем $\tilde{R}(r, 1, w) \geq 1 - l! / (r^l (1 + o(1)))$. \square

В утверждении ниже как и ранее требуем, чтобы размер локальности каждой вершины не превышал общего количества вершин.

Утверждение 5. Для $l = 2$ и r кратных 2 выполнено: $\hat{R}(r, 1, n) = \tilde{R}(r, 1, w) = 1 - \frac{2}{r^2 + 2r + 2}$.

Доказательство. Рассмотрим следующее покрытие тора шарами радиуса $\frac{r}{2}$. Разместим шар с центром в точке (x, y) тора. Разместим четыре

шара с центрами в точках $(x + \frac{r}{2}, y - \frac{r}{2} - 1)$, $(x + \frac{r}{2} + 1, y + \frac{r}{2})$, $(x - \frac{r}{2}, y + \frac{r}{2} + 1)$, $(x - \frac{r}{2} - 1, y - \frac{r}{2})$. Далее будем продолжать так же для уже построенных шаров. Ниже показан пример такого покрытия для $l = 2$ и $r = 4$.



Построим граф G , вершинами которого будут являться вершины тора, а рёбрами будут соединены все пары вершин, расстояние между которыми не превышает r . В шар радиуса $\frac{r}{2}$ входит $\frac{r^2+2r+2}{2}$ вершин тора. Шары из рассмотренного выше замощения покрывают все вершины тора без пересечений, за исключением вершин на краях. Расстояние между центрами шаров превышает r , поэтому можно рассмотреть центры шаров покрытия как множество независимых вершин графа G для леммы 2, откуда получаем $\tilde{R}(r, 1, w) \leq 1 - \frac{2}{r^2+2r+2}$. С другой стороны, сами шары образуют разбиение на клики диаметра r графа G , за исключением вершин на краях. Поэтому из леммы 1 получаем $\tilde{R}(r, 1, w) \geq 1 - \frac{2}{r^2+2r+2}$, так как количество непокрытых вершин на краях равно $o(n_1 n_2)$.

Построим конструкцию 1 над тором. Тор является транзитивным графом, поэтому из теоремы 1 получаем, что $\hat{R}(r, 1, n) = 1 - \frac{2}{r^2+2r+2}$. \square

5. Заключение

В разделе 2 были получены оценки на скорость и минимальное расстояние для произвольных графов. В разделе 3 представлена конструкция для произвольных графов с субпакетизацией. В разделе 4 были рассмотрены коды на простейших графах – цикле и торе. Были получены оценки на скорость полученной конструкции и показана оптимальность этой конструкции для некоторых типов графов. Большой интерес в литературе представляет минимальное расстояние кодов, в статье этому уделено немного внимания. В представленной конструкции субпакетизация используется для построения локального кода, в общем случае интерес

представляет также вопрос насколько субпакетизация может улучшить параметры r -локального кода.

Список литературы

- [1] Ф.Дж.Мак-Вильямс, Н.Дж.А.Слоэн, *Теория кодов исправляющих ошибки*, «Связь», Москва, 1979, 744 с.
- [2] S. Reed, G. Solomon, “Polynomial Codes Over Certain Finite Fields”, *Journal of the Society for Industrial and Applied Mathematics*, **8** (1960), 300-304
- [3] I. Tamo, A. Barg, “A family of optimal locally recoverable codes”, *IEEE Transactions on Information Theory*, **80**:8 (2014), 4661-4676
- [4] I. Tamo, A. Barg, A. Frolov, “Bounds on the Parameters of Locally Recoverable Codes”, *IEEE Transactions on Information Theory*, **62**:6 (2016), 3070-3083
- [5] I. Tamo, A. Barg, S. Vladut, “Locally recoverable codes on algebraic curves”, *IEEE Transactions on Information Theory*, **63**:8 (2017), 4928-4939
- [6] P. Gopalan, G. Hu, S. Kopparty, S. Saraf, C. Wang, S. Yekhanin, “Maximally Recoverable Codes for Grid-like Topologies”, *Proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2017, 2092-2108
- [7] P. Huang, E. Yaakobi, P. H. Siegel, “Multi-Erasure Locally Recoverable Codes Over Small Fields: A Tensor Product Approach”, *IEEE Transactions on Information Theory*, **66**:5 (2020), 2609 - 2624
- [8] A. Mazumdar, “Storage Capacity of Repairable Networks”, *IEEE Transactions on Information Theory*, **61**:11 (2015), 5810 - 5821

References

- [1] F. MacWilliams, N. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam, 1977, 762 с.
- [2] S. Reed, G. Solomon, “Polynomial Codes Over Certain Finite Fields”, *Journal of the Society for Industrial and Applied Mathematics*, **8** (1960), 300-304
- [3] I. Tamo, A. Barg, “A family of optimal locally recoverable codes”, *IEEE Transactions on Information Theory*, **80**:8 (2014), 4661-4676
- [4] I. Tamo, A. Barg, A. Frolov, “Bounds on the Parameters of Locally Recoverable Codes”, *IEEE Transactions on Information Theory*, **62**:6 (2016), 3070-3083
- [5] I. Tamo, A. Barg, S. Vladut, “Locally recoverable codes on algebraic curves”, *IEEE Transactions on Information Theory*, **63**:8 (2017), 4928-4939
- [6] P. Gopalan, G. Hu, S. Kopparty, S. Saraf, C. Wang, S. Yekhanin, “Maximally Recoverable Codes for Grid-like Topologies”, *Proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2017, 2092-2108
- [7] P. Huang, E. Yaakobi, P. H. Siegel, “Multi-Erasure Locally Recoverable Codes Over Small Fields: A Tensor Product Approach”, *IEEE Transactions on Information Theory*, **66**:5 (2020), 2609 - 2624
- [8] A. Mazumdar, “Storage Capacity of Repairable Networks”, *IEEE Transactions on Information Theory*, **61**:11 (2015), 5810 - 5821

Locally recoverable codes on graphs

Valinurov D.Y.

The locally recoverable codes (LRC codes) are linear codes with an important for applications property that every symbol of a codeword can be recovered from a small set of other symbols. Codeword symbols can be interpreted as servers with information. It is natural to define a topology of these servers as a graph so that every server can be recovered using neighborhood servers in this graph. The paper provides LRC code constructions for some topologies and bounds on the rate of such codes.

Keywords: erasure coding, locally recoverable codes, codes on graphs.

Λ-выражения для примитивно-рекурсивных функций в иерархии Гжегорчика

Коновалов А.Ю.¹

В данной работе рассматриваются Λ-выражения, построенные на основе универсальных функций для классов примитивно-рекурсивных функций иерархии Гжегорчика. Найдено достаточное условие на вид Λ-выражения, при котором это Λ-выражение определяет примитивно-рекурсивную функцию из заданного класса иерархии Гжегорчика.

Ключевые слова: иерархия Гжегорчика, примитивно-рекурсивные функции, строгая примитивно-рекурсивная реализуемость.

1. Введение

В [1] А. Гжегорчиком была разработана иерархия примитивно-рекурсивных функций, которая в дальнейшем была улучшенная П. Акстом [2], и на основе которой З. Дамнянович определил понятие строгой примитивно-рекурсивной реализуемости [3]. При исследовании данного вида реализуемости естественным образом возникают термы (далее — φ -термы), строящиеся из выражений вида $\varphi_z^i(x_1, \dots, x_n)$, где φ_z^i — обозначение для примитивно-рекурсивной функции с номером z в нумерации П. Акста для i -го уровня иерархии примитивно-рекурсивных функций. Одной из особенностей φ -термов, которая затрудняет работу с ними, является их «многоэтажность». В общем случае она не может быть устранена в силу того факта, что для класса всех примитивно-рекурсивных функций, находящихся на i -ом уровне иерархии Акста, отсутствует принадлежащая этому классу универсальная функция. В данной работе разработана

¹Коновалов Александр Юрьевич — канд. физ.-мат. наук, мл. науч. сотр. лаб. математических проблем искусственного интеллекта каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: konoval@yopmail.com.

Konovalov Aleksandr Yurevich — Candidate of Physico-Mathematical Sciences, Junior Researcher, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems, Laboratory of Mathematical Problem of Artificial Intelligence.

техника, которая позволяет осуществлять эквивалентные преобразования φ -термов, решая нетривиальную задачу сведения одних φ -термов к другим. Автор полагает, что полученные результаты найдут применение в дальнейших исследованиях понятия строгой примитивно-рекурсивной реализуемости.

Работа выполнена при финансовой поддержке РФФИ, проект №20-01-00670.

2. Иерархия примитивно-рекурсивных функций

Будем использовать следующую общеупотребительную символику для замены слов и словосочетаний естественного языка: $\hat{=}$ — «есть по определению»; \forall — «для всех»; \exists — «существует»; \iff — «тогда и только тогда, когда»; \implies — «если ..., то ...»; \neg — «неверно, что». Символ \square означает конец доказательства. Посредством \mathbb{N} обозначаем множество натуральных чисел $\{0, 1, 2, \dots\}$.

Примитивно-рекурсивными называют функции натурального аргумента, которые могут быть получены с помощью операций подстановки и рекурсии из следующих исходных функций: функции-константы $O(x) = 0$, функции прибавления единицы $S(x) = x + 1$ и семейства функций проекции $I_i^n(x_1, \dots, x_n) = x_i$ ($n = 1, 2, \dots, 1 \leq i \leq n$).

Пусть p_1, p_2, \dots — последовательность всех простых чисел. Посредством $\langle a_1, \dots, a_n \rangle$ будем обозначать натуральное число $p_1^{a_1} \cdot \dots \cdot p_n^{a_n}$, а через $\mathfrak{p}_i a$ — степень простого числа p_i при разложении натурального числа $a \geq 1$ на простые множители. Выражение $\mathfrak{p}_i 0$ будем считать равным 0 для любого i .

Будем использовать следующее обозначение для функций-констант: $\mathfrak{c}_a^n(x_1, \dots, x_n) = a$ ($n \geq 1, a \geq 0$). Последовательность функций f_n определяется в [1, §4] следующими соотношениями:

$$\begin{aligned} f_0(x, y) &= y + 1; \\ f_1(x, y) &= x + y; \\ f_2(x, y) &= (x + 1) \cdot (y + 1); \\ \text{для } n \geq 3 \quad f_n(x + 1, y) &= f_n(x, f_n(x, y)), \\ f_n(0, y) &= f_{n-1}(y + 1, y + 1). \end{aligned}$$

Введем обозначения для функций сигнум и симметрической разности:

$$\text{sg}(x) = \begin{cases} 0, & \text{если } x = 0, \\ 1, & \text{если } x > 0; \end{cases} \quad x \ominus y = \begin{cases} x - y, & \text{если } x > y, \\ 0, & \text{если } x \leq y. \end{cases}$$

Следуя [2] и [5], определим иерархию классов $\mathbf{E}_0 \subseteq \mathbf{E}_1 \subseteq \dots$, объединение которых есть в точности множество всех примитивно-рекурсивных

функций. Будем говорить, что функция $\varphi(z, \bar{x})$ получена *ограниченной рекурсией* (по Аксту) из функций $\psi(\bar{x})$, $\chi(z, y, \bar{x})$ и $\xi(z, \bar{x})$, если выполняются следующие соотношения:

$$\begin{aligned}\varphi(0, \bar{x}) &= \psi(\bar{x}), \\ \varphi(z+1, \bar{x}) &= \chi(z, \varphi(z, \bar{x}), \bar{x}) \cdot \text{sg}(\varphi(z, \bar{x})) \cdot \text{sg}(\xi(z, \bar{x}) \ominus \varphi(z, \bar{x})).\end{aligned}$$

Пусть $\Theta = \{\theta_1, \dots, \theta_m\}$ — конечное множество функций натурального аргумента. Посредством $\mathbf{E}^4(\Theta)$ будем обозначать наименьший замкнутый относительно подстановки и ограниченной рекурсии класс функций, который содержит все функции из Θ , а также следующие базовые функции: S , sg , \ominus , f_4 , I_i^n , c_a^n ($n \geq 1$, $1 \leq i \leq n$, $a \geq 0$). Определим нумерацию функций из класса $\mathbf{E}^4(\Theta)$ индукцией по построению функций из этого класса:

- $\langle 0, k_i, i \rangle$ — номер функции θ_i , где k_i — валентность θ_i ($1 \leq i \leq m$);
- $\langle 1, 1 \rangle$ — номер функции S ;
- $\langle 2, n, a \rangle$ — номер функции c_a^n ($n \geq 1$, $a \geq 0$);
- $\langle 3, n, i \rangle$ — номер функции I_i^n ($n \geq 1$, $1 \leq i \leq n$);
- $\langle 4, 1 \rangle$ — номер функции sg ;
- $\langle 5, 2 \rangle$ — номер функции \ominus ;
- $\langle 6, 2 \rangle$ — номер функции f_4 ;
- $\langle 7, n, e_0, e_1, \dots, e_k \rangle$ — номер функции $\varphi(\bar{x}) = \psi(\chi_1(\bar{x}), \dots, \chi_k(\bar{x}))$ при условии, что натуральное число e_0 — номер k -местной функции $\psi \in \mathbf{E}^4(\Theta)$, а натуральные числа e_1, \dots, e_k — номера n -местных функций $\chi_1, \dots, \chi_k \in \mathbf{E}^4(\Theta)$ соответственно ($n \geq 1$, $k \geq 1$);
- $\langle 8, n+1, e_0, e', e'' \rangle$ — номер $(n+1)$ -местной функции φ , полученной ограниченной рекурсией из функций ψ , χ и ξ :

$$\begin{aligned}\varphi(0, \bar{x}) &= \psi(\bar{x}), \\ \varphi(z+1, \bar{x}) &= \chi(z, \varphi(z, \bar{x}), \bar{x}) \cdot \text{sg}(\varphi(z, \bar{x})) \cdot \text{sg}(\xi(z, \bar{x}) \ominus \varphi(z, \bar{x}))\end{aligned}$$

при условии, что e_0, e', e'' — номера функций $\psi, \chi, \xi \in \mathbf{E}^4(\Theta)$ соответственно ($n \geq 1$).

Множество всех номеров функций из класса $\mathbf{E}^4(\Theta)$ обозначаем через I^Θ . Если $e \in I^\Theta$, то посредством φ_e^Θ обозначаем pe -местную функцию из $\mathbf{E}^4(\Theta)$ с номером e . Множество всех номеров n -местных функций из

класса $\mathbf{E}^4(\Theta)$ обозначаем через I_n^Θ ($n \geq 1$). Определим 2-местную функцию u^Θ :

$$u^\Theta(e, a) = \begin{cases} \varphi_e^\Theta(p_1 a, \dots, p_{(p_2 e)} a), & \text{если } e \in I^\Theta, \\ 0, & \text{иначе.} \end{cases}$$

Таким образом, функция u^Θ является универсальной для класса $\mathbf{E}^4(\Theta)$ в том смысле, что выполняется соотношение

$$\varphi_e^\Theta(a_1, \dots, a_n) = u^\Theta(e, \langle a_1, \dots, a_n \rangle)$$

для всех натуральных чисел a_1, \dots, a_n и $e \in I_n^\Theta$. Используя диагональный метод (рассмотрев функцию $u^\Theta(x, \langle x \rangle) + 1$) можно показать, что функция u^Θ не принадлежит классу $\mathbf{E}^4(\Theta)$.

Определим иерархию классов $\mathbf{E}^4(\Theta_0) \subseteq \mathbf{E}^4(\Theta_1) \subseteq \dots$, задав последовательность множеств $\Theta_0 \subseteq \Theta_1 \subseteq \dots$ следующим образом:

$$\Theta_0 = \emptyset, \quad \Theta_1 = \{u^{\Theta_0}\}, \quad \dots, \quad \Theta_i = \Theta_{i-1} \cup \{u^{\Theta_{i-1}}\}, \quad \dots$$

Будем использовать следующие сокращения:

$$\mathbf{E}_i = \mathbf{E}^4(\Theta_i), \quad u_i = u^{\Theta_i}, \quad I^i = I^{\Theta_i}, \quad I_n^i = I_n^{\Theta_i}, \quad \varphi_e^i = \varphi_e^{\Theta_i} \quad (i \geq 0, n \geq 1, e \geq 1).$$

Предложение 1. *Объединение $\bigcup_{i=0}^{\infty} \mathbf{E}_i$ представляет собой множество всех примитивно-рекурсивных функций.*

Доказательство. См. [1, Теорема 4.13], [2, Теорема на стр. 58]. \square

Введем обозначения для следующих функций:

$$\begin{aligned} t^n(x_1, \dots, x_n) &= \langle x_1, \dots, x_n \rangle; \\ \varepsilon(x, y) &= \begin{cases} 1, & \text{если } x = y, \\ 0, & \text{если } x \neq y; \end{cases} & \text{if}(z, x, y) &= \begin{cases} x, & \text{если } z = 0, \\ y, & \text{иначе;} \end{cases} \\ \text{in}(z, i) &= \begin{cases} 1, & \text{если } z \in I^i, \\ 0, & \text{иначе;} \end{cases} & \text{in}'(z, i, n) &= \begin{cases} 1, & \text{если } z \in I_n^i, \\ 0, & \text{иначе.} \end{cases} \end{aligned}$$

Предложение 2. *Функции сложения, умножения, возведения в степень и функции $\varepsilon, \text{if}, \text{in}, \text{in}', p_n, t^n$ ($n \geq 1$) принадлежат классу \mathbf{E}_0 .*

Доказательство. Функции сложения, умножения и возведения в степень являются элементарными по Кальмару [1, стр. 8, 9]. Из [1, Теоремы 4.4, 4.7] и [2, Теорема на стр. 58] следует, что все элементарные по Кальмару функции принадлежат \mathbf{E}_0 . Функция in принадлежит классу \mathbf{E}_0 в силу [2, §2], [5, §46]. Остальные функции принадлежат классу \mathbf{E}_0 , так как могут быть получены при помощи следующих подстановок:

- $\varepsilon(x, y) = (1 \ominus \mathbf{sg}(x \ominus y)) \cdot (1 \ominus \mathbf{sg}(y \ominus x))$
- $\mathbf{if}(z, x, y) = (1 \ominus z) \cdot x + \mathbf{sg}(z) \cdot y$
- $t^n(x_1, \dots, x_n) = p_1^{x_1} \cdot \dots \cdot p_n^{x_n}$
- $\mathbf{p}_n(x) = \mathbf{exp}(x, p_n)$, где \mathbf{exp} — элементарная по Кальмару функция, определяемая на стр. 13 в [1].
- $\mathbf{in}'(z, i, n) = \mathbf{in}(z, i) \cdot \varepsilon(\mathbf{p}_2 z, n)$

□

Для всякой функции $g \in \mathbf{E}_0$ фиксируем некоторый ее номер в классе \mathbf{E}_0 , который будем обозначать $\Lambda[g]$. Без ограничения общности будем считать, что имеет место $1 \leq \mathbf{p}_1 \Lambda[g] \leq 6$, если g — базовая функция, т.е. для базовых функций фиксирован именно тот номер, который указан в определении нумерации класса $\mathbf{E}^4(\Theta)$. Множество всех n -местных функций из класса \mathbf{E}_0 обозначаем \mathbf{E}_0^n ($n \geq 1$).

3. φ -термы

Назовем φ -атомами предметные переменные и натуральные числа. Будем говорить, что выражение t есть φ -терм, если выполняется одно из следующих условий:

- t — φ -атом;
- t имеет вид $g(t_1, \dots, t_n)$, где $g \in \mathbf{E}_0^n$, и t_1, \dots, t_n — φ -термы ($n \geq 1$);
- t имеет вид $\varphi_s^v(t_1, \dots, t_n)$, где s, t_1, \dots, t_n — φ -термы, v — φ -атом ($n \geq 1$).

Через $Var(t)$ обозначаем множество всех предметных переменных, входящих в φ -терм t . Будем называть *замкнутыми* те φ -термы, которые не содержат переменных.

Для каждого замкнутого φ -терма t индуктивно определим его значение $v(t) \in \mathbb{N} \cup \{-1\}$. Будем писать $!t$, если $v(t) \neq -1$.

- $v(a) \Rightarrow a$, если a — натуральное число;
- $v(g(t_1, \dots, t_n)) \Rightarrow \begin{cases} g(v(t_1), \dots, v(t_n)), & \text{если } !t_1, \dots, !t_n; \\ -1, & \text{иначе} \end{cases}$, где $g \in \mathbf{E}_0^n$;

$$\begin{aligned}
& \bullet v(\varphi_s^i(t_1, \dots, t_n)) \Leftrightarrow \\
& \Leftrightarrow \begin{cases} \varphi_{v(s)}^i(v(t_1), \dots, v(t_n)), & \text{если } !s, !t_1, \dots, !t_n \text{ и } v(s) \in I_n^i; \\ -1, & \text{иначе.} \end{cases}
\end{aligned}$$

Через $[k_1, \dots, k_n/x_1, \dots, x_n]t$ будем обозначать результат подстановки в φ -терм t натуральных чисел k_1, \dots, k_n вместо всех вхождений переменных x_1, \dots, x_n соответственно. Пусть t_0, t_1, t_2 — φ -термы, все переменные которых содержатся среди списка переменных x_1, \dots, x_n . Будем говорить, что φ -термы t_1 и t_2 *условно равны*, и писать $t_1 \simeq t_2$, если имеет место

$$v([k_1, \dots, k_n/x_1, \dots, x_n]t_1) = v([k_1, \dots, k_n/x_1, \dots, x_n]t_2) \quad (1)$$

для всех натуральных чисел k_1, \dots, k_n . Будем говорить, что φ -терм t_2 является *доопределением* φ -терма t_1 , и писать $t_1 \preceq t_2$, если имеет место соотношение (1) для всех натуральных чисел k_1, \dots, k_n , для которых верно $![k_1, \dots, k_n/x_1, \dots, x_n]t_1$. Назовем φ -терм t_0 *всюду определенным*, если выполняется $![k_1, \dots, k_n/x_1, \dots, x_n]t_0$ для всех натуральных чисел k_1, \dots, k_n . Нетрудно показать, что отношения \simeq и \preceq , а также понятие всюду определенного терма не зависят от выбора списка переменных x_1, \dots, x_n .

Определим отношение $u \in_{\bar{a}} t$, где u — φ -атом, t — φ -терм, \bar{a} — конечная последовательность элементов множества $\mathbb{N} \cup \{\uparrow, \downarrow\}$, индукцией по построению φ -терма t :

- $u \in_0 u$;
- $u \in_{i, \bar{q}} g(t_1, \dots, t_n)$, если $u \in_{\bar{q}} t_i$ ($1 \leq i \leq n, g \in \mathbf{E}_0^n$);
- $u \in_{\uparrow} \varphi_s^u(t_1, \dots, t_n)$;
- $u \in_{\downarrow, \bar{q}} \varphi_s^u(t_1, \dots, t_n)$, если $u \in_{\bar{q}} s$;
- $u \in_{i, \bar{q}} \varphi_s^u(t_1, \dots, t_n)$, если $u \in_{\bar{q}} t_i$ ($1 \leq i \leq n$).

Произвольную тройку (u, t, \bar{a}) , для которой имеет место $u \in_{\bar{a}} t$, назовем *вхождением* φ -атома u в φ -терм t . Будем говорить, что φ -атом u имеет вхождение в φ -терм t , если верно $u \in_{\bar{a}} t$ для некоторой последовательности \bar{a} . *Уровнем* вхождения (u, t, \bar{a}) назовем число символов \downarrow в последовательности \bar{a} . Будем говорить, что вхождение (u, t, \bar{a}) есть *вхождение в качестве верхнего индекса*, если последовательность \bar{a} оканчивается символом \uparrow . Вхождения (u, t, \bar{a}) , для которых последний символ последовательности \bar{a} есть 0 , назовем *вхождениями в качестве аргумента*.

Нетрудно видеть, что любое вхождение есть либо вхождение в качестве верхнего индекса, либо вхождение в качестве аргумента.

Множество всех φ -атомов, которые имеют вхождения в качестве верхнего индекса в φ -терм t с уровнем вхождения равным i , обозначаем $In^i(t)$. Множество всех переменных, которые имеют вхождения в качестве аргумента в φ -терм t с уровнем вхождения равным i , обозначаем $Arg^i(t)$. *Высотой* φ -терма t назовем уровень вхождения, имеющего наибольший уровень среди всех вхождений φ -атомов в него. Высоту φ -терма t обозначаем через $h(t)$. Следующие леммы непосредственно следуют из выше определенных понятий.

Лемма 1. Пусть s, t_1, \dots, t_n — φ -термы, u — φ -атом, i — натуральное число. Тогда выполняются следующие соотношения

$$\begin{aligned} In^i(g(t_1, \dots, t_n)) &= \bigcup_{k=1}^n In^i(t_k), & Arg^i(g(t_1, \dots, t_n)) &= \bigcup_{k=1}^n Arg^i(t_k), \\ In^0(\varphi_s^u(t_1, \dots, t_n)) &= \{u\} \cup \bigcup_{k=1}^n In^0(t_k), & Arg^0(\varphi_s^u(t_1, \dots, t_n)) &= \bigcup_{k=1}^n Arg^0(t_k), \\ In^{i+1}(\varphi_s^u(t_1, \dots, t_n)) &= In^i(s) \cup \bigcup_{k=1}^n In^{i+1}(t_k), \\ Arg^{i+1}(\varphi_s^u(t_1, \dots, t_n)) &= Arg^i(s) \cup \bigcup_{k=1}^n Arg^{i+1}(t_k). \end{aligned}$$

Лемма 2. Пусть s, t_1, \dots, t_n — φ -термы, u — φ -атом. Тогда выполняются следующие соотношения:

$$\begin{aligned} h(u) &= 0; \\ h(g(t_1, \dots, t_n)) &= \max(h(t_1), \dots, h(t_n)), \text{ где } g \in \mathbf{E}_0^n; \\ h(\varphi_s^u(t_1, \dots, t_n)) &= \max(h(s) + 1, h(t_1), \dots, h(t_n)). \end{aligned}$$

Предложение 3. Пусть t — φ -терм. Тогда для всех натуральных чисел n имеет место

$$h(t) \leq n \iff In^n(t) = \emptyset. \quad (2)$$

Доказательство. Индукция по построению φ -терма t .

Пусть t — φ -атом и n — произвольное натуральное число. Тогда верно $h(t) = 0$ и $In^n(t) = \emptyset$. Следовательно, имеет место (2).

Пусть t имеет вид $g(t_1, \dots, t_m)$ и n — произвольное натуральное число. Тогда, применяя предположение индукции и леммы 1, 2, получаем

$$\begin{aligned} h(t) \leq n &\iff \forall i = 1, \dots, m : h(t_i) \leq n \iff \\ &\iff \forall i = 1, \dots, m : In^n(t_i) = \emptyset \iff In^n(t) = \emptyset. \end{aligned}$$

Пусть t имеет вид $\varphi_s^u(t_1, \dots, t_m)$ и n — произвольное натуральное число. Заметим, что верно $h(t) > 0$ и $u \in In^0(t)$. Таким образом, имеет место (2) при $n = 0$. Если $n > 0$, то, применяя предположение индукции и леммы 1, 2, получаем

$$h(t) \leq n \iff h(s) \leq n - 1 \text{ и } \forall i = 1, \dots, m : h(t_i) \leq n \iff \\ In^{n-1}(s) = \emptyset \text{ и } \forall i = 1, \dots, m : In^n(t_i) = \emptyset \iff In^n(t) = \emptyset.$$

□

4. Λ -выражения

Пусть t — φ -терм, $\bar{x} = x_1, \dots, x_m$ — список предметных переменных. Определим φ -терм $\Lambda\bar{x}.t$ индукцией по построению φ -терма t :

- $\Lambda\bar{x}.a \iff \Lambda[c_a^m]$, где a — натуральное число;
- $\Lambda\bar{x}.y \iff \Lambda[I_k^m]$, если y есть x_k ($1 \leq k \leq m$);
- $\Lambda\bar{x}.y \iff \langle 2, m, y \rangle$, если переменная y не входит в список \bar{x} ;
- $\Lambda\bar{x}.\varphi_s^u(t_1, \dots, t_n) \iff \langle 7, m, s, \Lambda\bar{x}.t_1, \dots, \Lambda\bar{x}.t_n \rangle$;
- $\Lambda\bar{x}.g(t_1, \dots, t_n) \iff \langle 7, m, \Lambda[g], \Lambda\bar{x}.t_1, \dots, \Lambda\bar{x}.t_n \rangle$.

Отметим, что если переменная y не входит в список \bar{x} , то имеет место $v([a/y] \Lambda\bar{x}.y) = \Lambda[c_a^m]$ для всех натуральных чисел a .

Предложение 4. Пусть t — φ -терм, \bar{x} — список предметных переменных. Тогда

$$In^i(\Lambda\bar{x}.t) = In^{i+1}(t) \text{ для всех } i \geq 0; \quad (3)$$

$$Arg^i(\Lambda\bar{x}.t) = Arg^{i+1}(t) \text{ для всех } i > 0; \quad (4)$$

$$Arg^0(\Lambda\bar{x}.t) = Arg^1(t) \cup (Arg^0(t) - \bar{x}). \quad (5)$$

Доказательство. Индукция по построению φ -терма t .

- Докажем соотношение (3).

Пусть t — φ -атом. Тогда $In^i(\Lambda\bar{x}.t) = \emptyset = In^{i+1}(t)$, так как в φ -термах t и $\Lambda\bar{x}.t$ нет верхних индексов.

Пусть t имеет вид $g(t_1, \dots, t_n)$. Тогда, применяя предположение индукции и лемму 1, получаем

$$In^i(\Lambda\bar{x}.t) = In^i(\langle 7, m, \Lambda[g], \Lambda\bar{x}.t_1, \dots, \Lambda\bar{x}.t_n \rangle) = \\ = \bigcup_{k=1}^n In^i(\Lambda\bar{x}.t_k) = \bigcup_{k=1}^n In^{i+1}(t_k) = In^{i+1}(g(t_1, \dots, t_n)).$$

Пусть t имеет вид $\varphi_s^u(t_1, \dots, t_n)$. Тогда, применяя предположение индукции и лемму 1, получаем

$$\begin{aligned} In^i(\Lambda\bar{x}.t) &= In^i(\langle 7, m, s, \Lambda\bar{x}.t_1, \dots, \Lambda\bar{x}.t_n \rangle) = \\ &= In^i(s) \cup \bigcup_{k=1}^n In^i(\Lambda\bar{x}.t_k) = In^i(s) \cup \bigcup_{k=1}^n In^{i+1}(t_k) = \\ &= In^{i+1}(\varphi_s^u(t_1, \dots, t_n)). \end{aligned}$$

- Докажем соотношение (4).

Пусть t — φ -атом. Тогда $h(t) = h(\Lambda\bar{x}.t) = 0$. Т.к. $i > 0$, получаем $Arg^i(\Lambda\bar{x}.t) = \emptyset = Arg^i(t)$.

В остальных случаях используем предположение индукции и лемму 1 аналогично доказательству соотношения (3).

- Докажем соотношение (5).

Пусть t — натуральное число a . Тогда

$$Arg^0(\Lambda\bar{x}.a) = Arg^0(\Lambda[c_a^m]) = \emptyset = Arg^1(a) \cup (Arg^0(a) - \bar{x}).$$

Пусть t есть x_i ($1 \leq i \leq n$). Тогда

$$Arg^0(\Lambda\bar{x}.x_i) = Arg^0(\Lambda[I_i^m]) = \emptyset = Arg^1(x_i) \cup (Arg^0(x_i) - \bar{x}).$$

Пусть t — предметная переменная z , не входящая в список \bar{x} . Тогда

$$Arg^0(\Lambda\bar{x}.z) = Arg^0(\langle 2, m, z \rangle) = \{z\} = Arg^1(z) \cup (Arg^0(z) - \bar{x}).$$

Пусть t имеет вид $g(t_1, \dots, t_n)$. Тогда, используя предположение индукции и лемму 1, получаем

$$\begin{aligned} Arg^0(\Lambda\bar{x}.t) &= Arg^0(\langle 7, m, \Lambda[g], \Lambda\bar{x}.t_1, \dots, \Lambda\bar{x}.t_n \rangle) = \\ &= \bigcup_{k=1}^n Arg^0(\Lambda\bar{x}.t_k) = \bigcup_{k=1}^n Arg^1(t_k) \cup \bigcup_{k=1}^n (Arg^0(t_k) - \bar{x}) = \\ &= Arg^1(g(t_1, \dots, t_n)) \cup (Arg^0(g(t_1, \dots, t_n)) - \bar{x}). \end{aligned}$$

Пусть t имеет вид $\varphi_s^u(t_1, \dots, t_n)$. Тогда, применяя предположение индукции и лемму 1, получаем

$$\begin{aligned}
Arg^0(\Lambda \bar{x}. t) &= Arg^0(\langle 7, m, s, \Lambda \bar{x}. t_1, \dots, \Lambda \bar{x}. t_n \rangle) = \\
&= Arg^0(s) \cup \bigcup_{k=1}^n Arg^0(\Lambda \bar{x}. t_k) = \\
&= Arg^0(s) \cup \bigcup_{k=1}^n Arg^1(t_k) \cup \bigcup_{k=1}^n (Arg^0(t_k) - \bar{x}) = \\
&= Arg^1(\varphi_s^u(t_1, \dots, t_n)) \cup (Arg^0(\varphi_s^u(t_1, \dots, t_n)) - \bar{x}).
\end{aligned}$$

□

Введем следующие обозначения:

$$In^{\geq i}(t) \equiv \bigcup_{k=i}^{\infty} In^k(t), \quad Arg^{\geq i}(t) \equiv \bigcup_{k=i}^{\infty} Arg^k(t). \quad (i \geq 0)$$

Из соотношений (3), (4) следует

$$In^{\geq i}(\Lambda \bar{x}. t) = In^{\geq i+1}(t) \text{ для всех } i \geq 0, \quad (6)$$

$$Arg^{\geq i}(\Lambda \bar{x}. t) = Arg^{\geq i+1}(t) \text{ для всех } i > 0. \quad (7)$$

Предложение 5. Пусть t — φ -терм, $\bar{x}_1, \dots, \bar{x}_n$ — списки предметных переменных. Тогда выполняются следующие соотношения:

$$In^{\geq i}(\Lambda \bar{x}_n \dots \Lambda \bar{x}_1. t) = In^{\geq i+n}(t) \text{ для всех } i \geq 0; \quad (8)$$

$$Arg^{\geq i}(\Lambda \bar{x}_n \dots \Lambda \bar{x}_1. t) = Arg^{\geq i+n}(t) \text{ для всех } i > 0; \quad (9)$$

$$Arg^0(\Lambda \bar{x}_n \dots \Lambda \bar{x}_1. t) = (\dots ((Arg^0(t) - \bar{x}_1) \cup Arg^1(t)) \dots - \bar{x}_n) \cup Arg^n(t); \quad (10)$$

$$Arg^{\geq 0}(\Lambda \bar{x}_n \dots \Lambda \bar{x}_1. t) = (\dots ((Arg^0(t) - \bar{x}_1) \cup Arg^1(t)) \dots - \bar{x}_n) \cup Arg^{\geq n}(t). \quad (11)$$

Доказательство. Соотношения (8), (9) следуют из (6), (7) соответственно. Применяя (9) при $i = 1$, получаем

$$\begin{aligned}
Arg^{\geq 0}(\Lambda \bar{x}_n \dots \Lambda \bar{x}_1. t) &= Arg^0(\Lambda \bar{x}_n \dots \Lambda \bar{x}_1. t) \cup Arg^{\geq 1}(\Lambda \bar{x}_n \dots \Lambda \bar{x}_1. t) = \\
&= Arg^0(\Lambda \bar{x}_n \dots \Lambda \bar{x}_1. t) \cup Arg^{\geq n+1}(t).
\end{aligned}$$

Таким образом, соотношение (11) сводится к (10). Докажем (10) индукцией по n . При $n = 1$ соотношение (10) справедливо в силу предложения

4. Используя предположение индукции для натурального числа n равного $n - 1$, получаем

$$\begin{aligned} & Arg^0(\Lambda \bar{x}_n \dots \Lambda \bar{x}_1. t) = \\ & = (\dots ((Arg^0(\Lambda \bar{x}_1. t) - \bar{x}_2) \cup Arg^1(\Lambda \bar{x}_1. t)) \dots - \bar{x}_n) \cup Arg^{n-1}(\Lambda \bar{x}_1. t). \end{aligned}$$

Используя тот факт, что для каждого $i = 1, \dots, n-1$ в силу предложения 4 имеет место $Arg^i(\Lambda \bar{x}_1. t) = Arg^{i+1}(t)$, получаем (10). \square

Предложение 6. Пусть t — φ -терм, $\bar{x}_1, \dots, \bar{x}_n$ — списки предметных переменных и множество $Arg^{\geq i}(t)$ не содержит переменных из списка \bar{x}_i для каждого $i = 1, \dots, n$. Тогда имеет место

$$Arg^{\geq 0}(\Lambda \bar{x}_n \dots \Lambda \bar{x}_1. t) = Arg^{\geq 0}(t) - \bar{x}_1 \dots \bar{x}_n.$$

Доказательство. Применяя (11), получаем

$$\begin{aligned} & Arg^{\geq 0}(\Lambda \bar{x}_n \dots \Lambda \bar{x}_1. t) = \\ & = (\dots ((Arg^0(t) - \bar{x}_1) \cup Arg^1(t)) \dots - \bar{x}_n) \cup Arg^{\geq n}(t) = \\ & = \left(\bigcup_{k=0}^{n-1} Arg^k(t) \cup Arg^{\geq n}(t) \right) - \bar{x}_1 \dots \bar{x}_n = Arg^{\geq 0}(t) - \bar{x}_1 \dots \bar{x}_n. \end{aligned}$$

\square

Предложение 7. Пусть t — φ -терм, v — φ -атом, $\bar{x} = x_1, \dots, x_m$ — список предметных переменных, и верно $In^0(t) \subseteq \{v\}$. Тогда имеет место следующее условное равенство:

$$\varphi_{\Lambda \bar{x}. t}^v(\bar{x}) \simeq t. \quad (12)$$

Доказательство. Пусть все переменные, входящие в φ -термы v и t , содержатся среди списка $\bar{y} = y_1, \dots, y_l$. Осуществим подстановку произвольных натуральных чисел $\bar{k} = k_1, \dots, k_l$ в правую и левую части условного равенства (12) вместо всех вхождений переменных y_1, \dots, y_l соответственно, и докажем, что имеет место

$$v([\bar{k}/\bar{y}] \varphi_{\Lambda \bar{x}. t}^v(\bar{x})) = v([\bar{k}/\bar{y}] t)$$

индукцией по построению φ -терма t . Обозначим натуральное число $[\bar{k}/\bar{y}]v$ через j , и список натуральных чисел $[\bar{k}/\bar{y}]\bar{x}$ через \bar{k}' .

- Пусть t есть натуральное число a . Тогда $\Lambda \bar{x}. t$ есть $\Lambda[c_a^m]$ и имеет место:

$$\begin{aligned} v([\bar{k}/\bar{y}] \varphi_{\Lambda \bar{x}. t}^v(\bar{x})) &= v(\varphi_{[\bar{k}/\bar{y}]\Lambda[c_a^m]}^{[\bar{k}/\bar{y}]v}([\bar{k}/\bar{y}]\bar{x})) = v(\varphi_{\Lambda[c_a^m]}^j(\bar{k}')) = \\ &= c_a^m(\bar{k}') = a = v(a) = v([\bar{k}/\bar{y}]t). \end{aligned}$$

- Пусть t есть переменная x_i ($1 \leq i \leq m$). Тогда $\Lambda \bar{x}. t$ есть $\Lambda[I_i^m]$ и имеет место:

$$\begin{aligned} \mathbf{v}([\bar{k}/\bar{y}] \varphi_{\Lambda \bar{x}. t}^v(\bar{x})) &= \mathbf{v}(\varphi_{[\bar{k}/\bar{y}] \Lambda[I_i^m]}^{[\bar{k}/\bar{y}] v}([\bar{k}/\bar{y}] \bar{x})) = \mathbf{v}(\varphi_{\Lambda[I_i^m]}^j(\bar{k}')) = \\ &= I_i^m(\bar{k}') = k'_i = \mathbf{v}(k'_i) = \mathbf{v}([\bar{k}/\bar{y}] x_i) = \mathbf{v}([\bar{k}/\bar{y}] t). \end{aligned}$$

- Пусть t есть переменная z , которая не входит в список \bar{x} . Обозначим $[\bar{k}/\bar{y}] z$ через b . Тогда имеет место $\mathbf{v}([\bar{k}/\bar{y}] \Lambda \bar{x}. t) = \mathbf{v}(\langle 2, m, b \rangle) = \Lambda[\mathbf{c}_b^m]$ и верно:

$$\begin{aligned} \mathbf{v}([\bar{k}/\bar{y}] \varphi_{\Lambda \bar{x}. t}^v(\bar{x})) &= \mathbf{v}(\varphi_{[\bar{k}/\bar{y}] \Lambda \bar{x}. t}^j(\bar{k}')) = \mathbf{v}(\varphi_{\langle 2, m, b \rangle}^j(\bar{k}')) = \\ &= \mathbf{c}_b^m(\bar{k}') = b = \mathbf{v}([\bar{k}/\bar{y}] z) = \mathbf{v}([\bar{k}/\bar{y}] t). \end{aligned}$$

- Пусть t имеет вид $g(t_1, \dots, t_n)$, где $g \in \mathbf{E}_0^n$ ($n \geq 1$). Введем следующие обозначения:

$$\begin{aligned} t' &\hat{=} [\bar{k}/\bar{y}] t, \quad (\Lambda \bar{x}. t)' \hat{=} [\bar{k}/\bar{y}] \Lambda \bar{x}. t, \\ t'_i &\hat{=} [\bar{k}/\bar{y}] t_i, \quad (\Lambda \bar{x}. t_i)' \hat{=} [\bar{k}/\bar{y}] \Lambda \bar{x}. t_i \quad (1 \leq i \leq n). \end{aligned}$$

Тогда

$$\begin{aligned} (\Lambda \bar{x}. t)' &= [\bar{k}/\bar{y}] \Lambda \bar{x}. t = [\bar{k}/\bar{y}] \langle 7, m, \Lambda[g], \Lambda \bar{x}. t_1, \dots, \Lambda \bar{x}. t_n \rangle = \\ &= \langle 7, m, \Lambda[g], (\Lambda \bar{x}. t_1)', \dots, (\Lambda \bar{x}. t_n)' \rangle. \quad (13) \end{aligned}$$

Из (13) следует, что верно $!(\Lambda \bar{x}. t)'$ и имеет место $\mathbf{v}((\Lambda \bar{x}. t)') \in \mathbf{l}_m^j$ тогда и только тогда, когда для всех $i = 1, \dots, n$ верно $!(\Lambda \bar{x}. t_i)'$ и имеет место $\mathbf{v}((\Lambda \bar{x}. t_i)') \in \mathbf{l}_m^j$.

Согласно лемме 1 имеет место $In^0(g(t_1, \dots, t_n)) = \bigcup_{i=1}^n In^0(t_i)$. Следовательно для каждого из φ -термов t_1, \dots, t_n выполняется условие доказываемого предложения: $In^0(t_i) \subseteq \bigcup_{i=1}^n In^0(t_i) \subseteq \{v\}$. Таким образом, в силу предположения индукции выполняются следующие соотношения:

$$\mathbf{v}(\varphi_{(\Lambda \bar{x}. t_i)'}^j(\bar{k}')) = \mathbf{v}(t'_i) \quad (1 \leq i \leq n). \quad (14)$$

Из (14) следует, что верно $!(\Lambda \bar{x}. t_i)'$ и $\mathbf{v}((\Lambda \bar{x}. t_i)') \in \mathbf{l}_m^j$ тогда и только тогда, когда верно $!t'_i$ ($1 \leq i \leq n$).

Рассмотрим случай, когда $\mathbf{v}(t'_i) = -1$ для некоторого натурального числа i ($1 \leq i \leq n$). Тогда или $\mathbf{v}((\Lambda \bar{x}. t_i)') = -1$, или $\mathbf{v}((\Lambda \bar{x}. t_i)') \notin \mathbf{l}_m^j$.

Следовательно, или $\mathbf{v}((\Lambda\bar{x}.t)') = -1$, или $\mathbf{v}((\Lambda\bar{x}.t)') \notin \mathbb{I}_m^j$. Таким образом, имеет место:

$$\begin{aligned} \mathbf{v}([\bar{k}/\bar{y}] \varphi_{\Lambda\bar{x}.t}^v(\bar{x})) &= \mathbf{v}(\varphi_{(\Lambda\bar{x}.t)'}^j(\bar{k}')) = -1 = \\ &= \mathbf{v}(g(t'_1, \dots, t'_n)) = \mathbf{v}(t') = \mathbf{v}([\bar{k}/\bar{y}] t). \end{aligned}$$

Теперь пусть $!t'_1, \dots, !t'_n$. Тогда $!(\Lambda\bar{x}.t_i)'$ и $\mathbf{v}((\Lambda\bar{x}.t_i)') \in \mathbb{I}_m^j$ для всех $i = 1, \dots, n$. Следовательно, верно $!(\Lambda\bar{x}.t)'$ и $\mathbf{v}((\Lambda\bar{x}.t)') \in \mathbb{I}_m^j$. Таким образом, имеет место следующая цепочка равенств:

$$\begin{aligned} \mathbf{v}([\bar{k}/\bar{y}] \varphi_{\Lambda\bar{x}.t}^v(\bar{x})) &= \varphi_{\mathbf{v}((\Lambda\bar{x}.t)')}^j(\bar{k}') = \\ &= \varphi_{\langle 7, m, \Lambda[g], \mathbf{v}((\Lambda\bar{x}.t_1)'), \dots, \mathbf{v}((\Lambda\bar{x}.t_n)') \rangle}^j(\bar{k}') = \\ &= \varphi_{\Lambda[g]}^j(\varphi_{\mathbf{v}((\Lambda\bar{x}.t_1)')}^j(\bar{k}'), \dots, \varphi_{\mathbf{v}((\Lambda\bar{x}.t_n)')}^j(\bar{k}')) = \\ &= g(\mathbf{v}(t'_1), \dots, \mathbf{v}(t'_n)) = \mathbf{v}(g(t'_1, \dots, t'_n)) = \mathbf{v}(t'). \end{aligned}$$

- Пусть t имеет вид $\varphi_s^u(t_1, \dots, t_n)$ ($n \geq 1$). Из $In^0(t) \subseteq \{v\}$ следует, что верхний индекс u есть φ -атом v . Введем следующие обозначения:

$$\begin{aligned} t' &\equiv [\bar{k}/\bar{y}] t, \quad (\Lambda\bar{x}.t)' \equiv [\bar{k}/\bar{y}] \Lambda\bar{x}.t, \\ t'_i &\equiv [\bar{k}/\bar{y}] t_i, \quad (\Lambda\bar{x}.t_i)' \equiv [\bar{k}/\bar{y}] \Lambda\bar{x}.t_i \quad (1 \leq i \leq n), \\ s' &\equiv [\bar{k}/\bar{y}] s. \end{aligned}$$

Тогда

$$\begin{aligned} (\Lambda\bar{x}.t)' &= [\bar{k}/\bar{y}] \Lambda\bar{x}.t = [\bar{k}/\bar{y}] \langle 7, m, s, \Lambda\bar{x}.t_1, \dots, \Lambda\bar{x}.t_n \rangle = \\ &= \langle 7, m, s', (\Lambda\bar{x}.t_1)', \dots, (\Lambda\bar{x}.t_n)' \rangle. \quad (15) \end{aligned}$$

Из (15) следует, что верно $!(\Lambda\bar{x}.t)'$ и имеет место $\mathbf{v}((\Lambda\bar{x}.t)') \in \mathbb{I}_m^j$ тогда и только тогда, когда имеет место $!s'$ и $\mathbf{v}(s') \in \mathbb{I}_n^j$, и для всех $i = 1, \dots, n$ верно $!(\Lambda\bar{x}.t_i)'$ и $\mathbf{v}((\Lambda\bar{x}.t_i)') \in \mathbb{I}_m^j$.

Согласно лемме 1 имеет место $In^0(\varphi_s^v(t_1, \dots, t_n)) = \{v\} \cup \bigcup_{i=1}^n In^0(t_i)$.

Следовательно, для каждого из φ -термов t_1, \dots, t_n выполняется условие доказываемого предложения: $In^0(t_i) \subseteq \{v\} \cup \bigcup_{i=1}^n In^0(t_i) \subseteq \{v\}$.

Таким образом, в силу предположения индукции выполняются соотношения (14). Из (14) следует, что имеет место $!(\Lambda\bar{x}.t_i)'$ и $\mathbf{v}((\Lambda\bar{x}.t_i)') \in \mathbb{I}_m^j$ тогда и только тогда, когда верно $!t'_i$ ($1 \leq i \leq n$).

Рассмотрим случай, когда $\mathbf{v}(t'_i) = -1$ для некоторого натурального числа i ($1 \leq i \leq n$). Тогда или $\mathbf{v}((\Lambda\bar{x}.t_i)') = -1$, или $\mathbf{v}((\Lambda\bar{x}.t_i)') \notin \mathbb{I}_m^j$.

Следовательно, или $v((\Lambda\bar{x}.t)') = -1$, или $v((\Lambda\bar{x}.t)') \notin l_m^j$. Таким образом, имеет место:

$$\begin{aligned} v([\bar{k}/\bar{y}] \varphi_{\Lambda\bar{x}.t}^v(\bar{x})) &= v(\varphi_{(\Lambda\bar{x}.t)'}^j(\bar{k}')) = -1 = \\ &= v(\varphi_{s'}^j(t'_1, \dots, t'_n)) = v(t') = v([\bar{k}/\bar{y}] t). \end{aligned} \quad (16)$$

Рассмотрим случай, когда $v(s') = -1$ или $v(s') \notin l_n^j$. Тогда или $v((\Lambda\bar{x}.t)') = -1$, или $v((\Lambda\bar{x}.t)') \notin l_m^j$. Таким образом, имеет место цепочка равенств (16).

Пусть теперь $!s', !t'_1, \dots, !t'_n$ и $v(s') \in l_n^j$. В этом случае имеет место $!(\Lambda\bar{x}.t_i)'$ и $v((\Lambda\bar{x}.t_i)') \in l_m^j$ для всех $i = 1, \dots, n$. Следовательно, верно $!(\Lambda\bar{x}.t)'$ и $v((\Lambda\bar{x}.t)') \in l_m^j$. Таким образом, имеет место следующая цепочка равенств:

$$\begin{aligned} v([\bar{k}/\bar{y}] \varphi_{\Lambda\bar{x}.t}^v(\bar{x})) &= \\ &= \varphi_{v((\Lambda\bar{x}.t)')}^j(\bar{k}') = \varphi_{\langle 7, m, v(s'), v((\Lambda\bar{x}.t_1)'), \dots, v((\Lambda\bar{x}.t_n)') \rangle}^j(\bar{k}') = \\ &= \varphi_{v(s')}^j(\varphi_{v((\Lambda\bar{x}.t_1)')}^j(\bar{k}'), \dots, \varphi_{v((\Lambda\bar{x}.t_n)')}^j(\bar{k}')) = \\ &= \varphi_{v(s')}^j(v(t'_1), \dots, v(t'_n)) = v(\varphi_{s'}^j(t'_1, \dots, t'_n)) = v(t'). \end{aligned}$$

□

Предложение 8. Пусть t — φ -терм, v_1, \dots, v_n — φ -атомы, $\bar{x}_1, \dots, \bar{x}_n$ — списки предметных переменных и для каждого $i = 1, \dots, n$ верно $In^{i-1}(t) \subseteq \{v_i\}$. Тогда имеет место следующее условное равенство:

$$\varphi^{v_1} \dots \varphi_{\Lambda\bar{x}_n \dots \Lambda\bar{x}_1.t}^{v_n}(\bar{x}_n) \dots (\bar{x}_1) \simeq t. \quad (17)$$

Доказательство. Индукция по n . Для $n = 1$ доказываемое утверждение верно в силу предложения 7. Предположим, что оно справедливо для натурального числа n равного $n - 1$. Тогда из предложения 4 следует, что для всех $i = 1, \dots, n - 1$ верно $In^{i-1}(\Lambda\bar{x}_1.t) = In^i(t) \subseteq \{v_{i+1}\}$. Таким образом, в силу предположения индукции имеет место

$$\varphi^{v_2} \dots \varphi_{\Lambda\bar{x}_n \dots \Lambda\bar{x}_2.\Lambda\bar{x}_1.t}^{v_n}(\bar{x}_n) \dots (\bar{x}_1) \simeq \Lambda\bar{x}_1.t. \quad (18)$$

Согласно предложению 7 верно

$$\varphi_{\Lambda\bar{x}_1.t}^{v_1}(\bar{x}) \simeq t. \quad (19)$$

Из (18), (19) получаем (17). □

вхождений в качестве верхних индексов. Таким образом, применяя (6), получаем

$$\text{Var}(\Lambda \bar{x}_n \dots \Lambda \bar{x}_1 . t) = \text{Arg}^{\geq 0}(\Lambda \bar{x}_n \dots \Lambda \bar{x}_1 . t) = \text{Arg}^{\geq 0}(t) - \bar{x}_1 \dots \bar{x}_n \quad (22)$$

Из (22) и условия 4 следует утверждение 1. В силу утверждения 1 функция $f(\bar{y}) := \Lambda \bar{x}_n \dots \Lambda \bar{x}_0 . t$ корректно определена. Утверждение 3 следует из утверждения 2. Утверждение 4 получаем из условий 2, 3 и предложения 8. \square

Теорема 2. Пусть t — φ -терм, v_0, \dots, v_n — φ -атомы, $\bar{x}_0, \dots, \bar{x}_n$ — списки предметных переменных, для которых выполняются следующие условия:

- 1) $h(t) \leq n + 1$;
- 2) для каждого $i = 0, \dots, n$ множество $\text{Arg}^{\geq i+1}(t)$ не содержит переменных из списка \bar{x}_i ;
- 3) для каждого $i = 0, \dots, n$ верно $\text{In}^i(t) \subseteq \{v_i\}$;
- 4) $\text{Arg}^{\geq 0}(t) \subseteq \bar{x}_0 \dots \bar{x}_n$.

Тогда выполняются следующие утверждения:

- 1) φ -терм $\Lambda \bar{x}_n \dots \Lambda \bar{x}_0 . t$ является замкнутым;
- 2) $h(\Lambda \bar{x}_n \dots \Lambda \bar{x}_0 . t) = 0$;
- 3) $!\Lambda \bar{x}_n \dots \Lambda \bar{x}_0 . t$;
- 4) для натурального числа $e := \Lambda \bar{x}_n \dots \Lambda \bar{x}_0 . t$ выполняется условное равенство

$$\varphi^{v_0} \dots \varphi^{v_n}(\bar{x}_n) \dots (\bar{x}_0) \simeq t. \quad (23)$$

Доказательство. Утверждения 1, 2 получаем применением теоремы 1. Утверждение 3 следует из утверждений 1, 2. Из утверждения 1, 3 получаем, что натуральное число $e := \Lambda \bar{x}_n \dots \Lambda \bar{x}_0 . t$ корректно определено. Утверждение 4 следует из условий 2, 3 и предложения 8. \square

Предложение 9. Для каждого $n \geq 1$ найдется такая функция $g_n \in \mathbf{E}_0^2$, что имеет место

$$\varphi^v \dots \varphi_z^v(x_1) \dots (x_n) \preceq \varphi^v \dots \varphi_{g_n(z,v)}^v(x_1) \dots (x_n) \quad (24)$$

и φ -терм из правой части (24) является всюду определенным.

Доказательство. Рассмотрим функцию $\phi(z, v) := \text{if}(\text{in}'(z, v, 1), \Lambda[\mathfrak{c}_0^1], z)$. Нетрудно видеть, что для всех натуральных чисел a, i верно $\phi(a, i) \in I_1^i$, и имеет место $\phi(a) = a$, если $a \in I_1^i$.

Доказательство проведем индукцией по натуральному числу n . Положим функцию g_1 равной ϕ и покажем, что выполняется (24) для $n = 1$. Действительно, пусть натуральные числа a, b, i таковы, что верно $! \varphi_a^i(b)$. Тогда $a \in I_1^i$ и $\phi(a, i) = a$. Следовательно, $\varphi_a^i(b) = \varphi_{\phi(a, i)}^i(b)$. Покажем теперь, что φ -терм $\varphi_{\phi(z, v)}^v(x)$ является всюду определенным. Действительно, для всех натуральных чисел a, b, i верно $\phi(a, i) \in I_1^i$ и следовательно имеет место $! \varphi_{\phi(a, i)}^i(b)$.

Построим функцию g_{n+1} в предположении, что имеется функция g_n , для которой выполняется (24). В силу теоремы 1 имеет место

$$\begin{aligned} \varphi_{\varphi^v}^v \cdots \varphi_z^v(x_1) \cdots (x_n)(x_{n+1}) &\preceq \varphi^v \left(\varphi^v \cdots \varphi_{g_n(z, v)}^v(x_1) \cdots (x_n), v \right) (x_{n+1}) \simeq \\ &\simeq \varphi_{\varphi^v}^v \cdots \varphi_{g_{n+1}(z, v)}^v(x_1) \cdots (x_n)(x_{n+1}), \end{aligned} \quad (25)$$

где

$$g_{n+1}(z, v) := \Lambda x_1 \dots \Lambda x_n \cdot \phi \left(\varphi^v \cdots \varphi_{g_n(z, v)}^v(x_1) \cdots (x_n), v \right).$$

Покажем, что φ -терм из правой части (25) является всюду определенным. Осуществим подстановку натуральных чисел $i, a, b_1, \dots, b_n, b_{n+1}$ вместо переменных $v, z, x_1, \dots, x_n, x_{n+1}$ в (25). В силу предположения индукции имеет место $! \varphi^i \cdots \varphi_{g_n(a, i)}^i(b_1) \cdots (b_n)$. Тогда

$$\phi \left(\varphi^i \cdots \varphi_{g_n(a, i)}^i(b_1) \cdots (b_n), i \right) \in I_1^i$$

и выполняется

$$! \varphi^i \left(\varphi^i \cdots \varphi_{g_n(a, i)}^i(b_1) \cdots (b_n), i \right) (b_{n+1}).$$

Следовательно, имеет место $! \varphi^i \cdots \varphi_{g_{n+1}(a, i)}^i(b_1) \cdots (b_n)(b_{n+1})$. \square

Предложение 10. Пусть i, a_1, \dots, a_n — натуральные числа. Не существует такого натурального числа e , для которого выполняется соотношение

$$\varphi^i \dots \varphi_{\varphi_z^i(x)}^i(a_1) \dots (a_n) \preceq \varphi^i \dots \varphi_{\varphi_e^i(z, x)}^i(a_1) \dots (a_n). \quad (26)$$

Доказательство. Предположим противное: для некоторого натурального числа e верно (26). Тогда в силу теоремы 2 верно

$$\begin{aligned} \varphi^i \dots \varphi_{\varphi_z^i(z)}^i(a_1) \dots (a_n) + 1 &\preceq \varphi^i \dots \varphi_{\varphi_e^i(z, z)}^i(a_1) \dots (a_n) + 1 \simeq \\ &\simeq \varphi^i \dots \varphi_{\varphi_{e'}^i(z)}^i(a_1) \dots (a_n) \preceq \varphi^i \dots \varphi_{\varphi_{e''}^i(z)}^i(a_1) \dots (a_n), \end{aligned}$$

где $e' := \Lambda z. \Lambda x_1 \dots \Lambda x_n. \left[\varphi^i \dots \varphi_{\varphi_e^i(z, z)}^i(a_1) \dots (a_n) + 1 \right]$ и $e'' = g_{n+1}(e', i)$,

а g_{n+1} — функция из предложения 9. Тогда $! \varphi^i \dots \varphi_{\varphi_{e''}^i(e'')}^i(a_1) \dots (a_n)$.

Следовательно, $\varphi^i \dots \varphi_{\varphi_{e''}^i(e'')}^i(a_1) \dots (a_n) + 1 = \varphi^i \dots \varphi_{\varphi_{e''}^i(e'')}^i(a_1) \dots (a_n)$.

Получили противоречие. \square

Список литературы

- [1] A. Grzegorzcyk, “Some classes of recursive functions”, *Rozprawy matematyczne*, **4** (1953), 1–46.
- [2] P. Axt, “Enumeration and the Grzegorzcyk hierarchy”, *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, **9** (1963), 53–65.
- [3] Z. Damnjanovic, “Strictly primitive recursive realizability”, *The Journal of Symbolic Logic*, **59:4** (1994), 1210–1227.
- [4] V. Plisko, “Primitive recursive realizability and basic propositional logic”, *Utrecht University, Logic Group Preprint Series*, **261** (2007), 27 pp.
- [5] С. К. Клини, *Введение в метаматематику.*, Мир, М., 1957.

References

- [1] A. Grzegorzcyk, “Some classes of recursive functions”, *Rozprawy matematyczne*, **4** (1953), 1–46.

- [2] P. Axt, “Enumeration and the Grzegorzcyk hierarchy”, *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, **9** (1963), 53–65.
- [3] Z. Damjanovic, “Strictly primitive recursive realizability”, *The Journal of Symbolic Logic*, **59**:4 (1994), 1210-1227.
- [4] V. Plisko, “Primitive recursive realizability and basic propositional logic”, *Utrecht University, Logic Group Preprint Series*, **261** (2007), 27 pp.
- [5] S. K. Kleene, *Introduction to Metamathematics*, Wolters-Noordhoff, 1952.

Λ -expressions for primitive recursive functions in the Grzegorzcyk hierarchy
Konovalov A.Yu.

In this paper we consider Λ -expressions constructed from universal functions for functional classes of the Grzegorzcyk hierarchy. We will find a sufficient condition such that a given Λ -expression determines a primitive recursive function from some level of the Grzegorzcyk hierarchy.

Keywords: Grzegorzcyk hierarchy, primitive recursive functions, strictly primitive recursive realizability.

Число состояний универсального автомата бесконечного экрана, реализующего двунаправленное движение на луче

Кузнецова Е.В.¹

В работе рассматривается реализация одного класса законов движения клеточным автоматом на бесконечном экране. Показано, что минимальное число состояний клеточного автомата, моделирующего двунаправленное движение точки на луче, при котором точка не совершает 2 движения вправо подряд, равно 5.

Ключевые слова: клеточный автомат, число состояний, бесконечный экран, двунаправленное движение, конструирование изображений.

1. Введение

Исследования по клеточным автоматам начались с работы Джона фон Неймана [1]. В 1960-х годах Муром была доказана теорема о Райском саде [2]. В 1970 году была опубликована игра Жизнь, автором которой является математик Джон Конвей [3]. В 2002 году была опубликована книга, в которой рассказывались широкие применения клеточных автоматов во всех областях науки [4].

На механико-математическом факультете МГУ им. Ломоносова исследованием клеточных автоматов занимались В.Б. Кудрявцев, А.С. Подколзин, А.А. Болотов. Результатами их исследований стала монография [5].

Тема конструирования стационарных изображений клеточными автоматами была развита Е. Титовой в ее работах [6, 7, 8, 9]. Так, в работе [6] рассматривалась задача конструирования изображений клеточными автоматами на прямоугольном экране. В работе было показано, что для

¹ Кузнецова Екатерина Викторовна — инженер-разработчик, ООО НКБ "НИР" e-mail: kuz.net.sova@mail.ru.

Kuznetsova Ekaterina Viktorovna — engineer developer, LLC NKB "NIR".

конструирования любого изображения необходимо и достаточно, чтобы клеточный автомат имел 3 состояния.

В работе [7] продолжалось рассмотрение конструирования изображений клеточным автоматом на прямоугольном экране. В работе были даны оценки времени формирования клеточным автоматом изображений для разного числа состояний данного автомата.

Далее исследования сместились в область движения изображений на экране, создаваемого клеточным автоматом.

Так, в работе [8], рассматривается движение одноклеточного изображения на конечном экране длины m . Даются оценки количества состояний для конструирования автоматов, реализующих такие законы движения. Так, например, показывается, что четырех состояний достаточно для создания автомата, реализующего любой закон движения, в котором отсутствуют два движения вперёд подряд на конечном экране.

В работе [9] было подробно рассмотрено исследование движения точки на бесконечном экране. В работе описан алгоритм реализации на экране широкого класса законов движения и исследована мера Бернулли множества реализуемых законов движения. Показано, что почти все законы движения являются реализуемыми. Также показано, что относительно Тихоновской топологии множество реализуемых законов движения относится к первой категории Бэра, т.е. очень мало.

Основным отличием предлагаемой работы является другой, более широкий класс законов движения, в котором разрешено движение назад.

В данной работе рассматривается конечный автомат и бесконечная справа полоса шириной в одну клетку. К каждой клетке полосы прикреплен свой экземпляр конечного автомата. Состояние автомата, прикрепленного к клетке, зависит от состояния в предыдущий момент времени этого автомата и двух его входов, левого и правого. Под входами будем понимать состояния автоматов, прикрепленных к соседним клеткам (имеются в виду две клетки: ближайшая слева и ближайшая справа). Нулевое значение состояния автомата считается состоянием покоя, и автомат в состоянии покоя, если его соседи тоже в состоянии покоя, остаётся в состоянии покоя.

В дальнейшем состоянии автомата, прикрепленного к клетке, будем называть состоянием клетки. Множество автоматов, прикреплённых к клеткам полубесконечной прямой, является примером клеточного автомата. Подробнее про клеточные автоматы можно почитать в книге [10].

Левый вход самой левой клетки полубесконечной полосы будем называть управляющим входом и будем подавать на него произвольные, но определённые управляющие сигналы.

Все клетки полубесконечной полосы будем называть экраном. Конфигурацию из состояний клеточных автоматов в данный момент времени будем называть изображением на экране.

Отметим, что состояния клетки интерпретируются не одинаково. Так, состояние 0 интерпретируется, как состояние покоя, некоторое заранее фиксированное подмножество состояний клетки, называемых метками, интерпретируются как клетки чёрного цвета, а все остальные состояния, включая состояние покоя, интерпретируются как клетки белого цвета. В результате на экране получается чёрно-белое изображение.

В работе накладывается ограничение на возможные изображения. Так, двух меток на экране быть не может, т.е. две разные клетки не могут принимать состояния, содержащиеся в подмножестве состояний, считающихся меткой.

Законом движения точки на экране назовем последовательность, состоящую из символов f, s, b (f – *forward*, s – *stop*, b – *back*), кодирующих перемещение точки по экрану в каждый момент времени. Так, если в момент времени t точка сместилась на одну клетку вправо, то t -ый член последовательности примет значение f , если сместилась влево, то t -ый член последовательности примет значение b , если никуда не сместилась, то t -ый член последовательности примет значение s . Здесь время t отсчитывается от того момента, когда в самой левой клетке появляется метка.

Подробнее про конструирование изображений и их движения на экране можно почитать в статьях [6, 8].

Нас будет интересовать определенный класс S законов движения, в которых нет двух символов f подряд.

Основной целью работы являлось определение наименьшего числа состояний клеточных автоматов, при котором можно реализовать все законы движения из класса S , при этом выбор управляющих сигналов через управляющий вход экрана остаётся за нами, и зависит от закона движения. В работе было показано, что минимальное количество состояний клеточного автомата, при котором можно реализовать любое движение из данного класса, равно пяти.

Автор выражает благодарность профессору Э.Э. Гасанову за научное руководство и постановку задачи, а также Г.В. Калачеву за помощь в работе.

2. Основные определения и формулировка результата

Определим основные понятия, используемые в данной работе.

Пусть S — множество конечных и бесконечных последовательностей, состоящих из элементов $\alpha_n \in \{sf, s, b\}$, в префиксе любой длины которых количество символов b не превышает количества символов f . Элементы множества S будем называть *законами движения*. Символ f подразумевает движение на одну клетку вправо, s — остаться на месте, b — на одну клетку влево.

Экраном будем называть следующую конструкцию.

Пусть имеется бесконечная в правую сторону полоса шириной в одну клетку. В каждую клетку полосы поместим по одному экземпляру одного и того же конечного автомата. К входам этого автомата присоединим выходы автоматов, стоящих в двух соседних с ним клетках, то есть у автомата имеется *левый* вход, *правый* вход и текущее состояние автомата. Выходом автомата в заданный момент времени является его состояние в этот момент времени. Для автомата, стоящего в самой левой клетке полосы левый вход не определён. Будем называть его *управляющим входом* и подавать на него управляющие сигналы.

Будем говорить, что *на экране реализуется движение по закону* $A \in S$, если выполняются следующие условия:

- 1) в некоторый момент времени в самой левой клетке экрана появляется метка (до этого на экране нет меток), этот момент будем называть *моментом начала движения* или *началом движения*;
- 2) изменение позиции метки на экране в i -й момент от начала движения соответствует i -й букве в слове или сверхслове A , а именно, если $A(i) = s$, то в $(i + 1)$ -й момент метка остаётся в той же клетке, где была в текущий момент, если $A(i) = f$, то в $(i + 1)$ -й момент метка сдвинется на одну ячейку вправо, если $A(i) = b$, то в $(i + 1)$ -й момент метка сдвинется на одну клетку влево, по сравнению со своим текущим положением;
- 3) в каждый момент времени после начала движения на экране есть ровно одна метка.

Значения состояний клеточного автомата, при которых считается, что клетка, находящаяся в данном состоянии, видима (чёрная) будем называть *метками*.

Экран будем называть *универсальным* для множества законов движения S , если для любого закона движения из S существует такая последовательность управляющих сигналов, что на экране формируется такое изображение, что метка движется по закону S .

В данной работе рассматривается клеточный автомат, заданный на бесконечной в правую сторону полосе. Причем левый вход первой слева клетки является управляющим, туда подаются управляющие сигналы

(его нельзя изменить через функцию переходов φ , определяющую зависимость состояния клетки от состояний в предыдущий момент времени этой самой клетки и соседних с ней клеток; просто на каждом следующем такте туда подается следующий элемент управляющей последовательности, вырабатываемой некоторым управляющим устройством по закону движения).

Изначально в данном клеточном автомате одни нули. Затем управляющее устройство начинает подавать ему на вход управляющие сигналы (управляющая последовательность). В какой-то момент в самой левой клетке экрана появится метка, которая интерпретируется как точка, движение которой мы и изучаем.

Таким образом, под появлением точки на экране будем подразумевать переключение клетки автомата, соответствующей самой левой клетке экрана, в состояние, соответствующее состоянию метки.

После того, как точка (метка) появилась на экране, то она никуда не исчезает и двух точек (меток) на экране быть не может (поэтому, если метка движется, например, вправо, то на изначальном месте она должна затереться, т.е. клетка, в которой была метка, должна перейти в состояние, не соответствующее метке). Возможное и невозможное поведение метки изображены на рисунке 1.

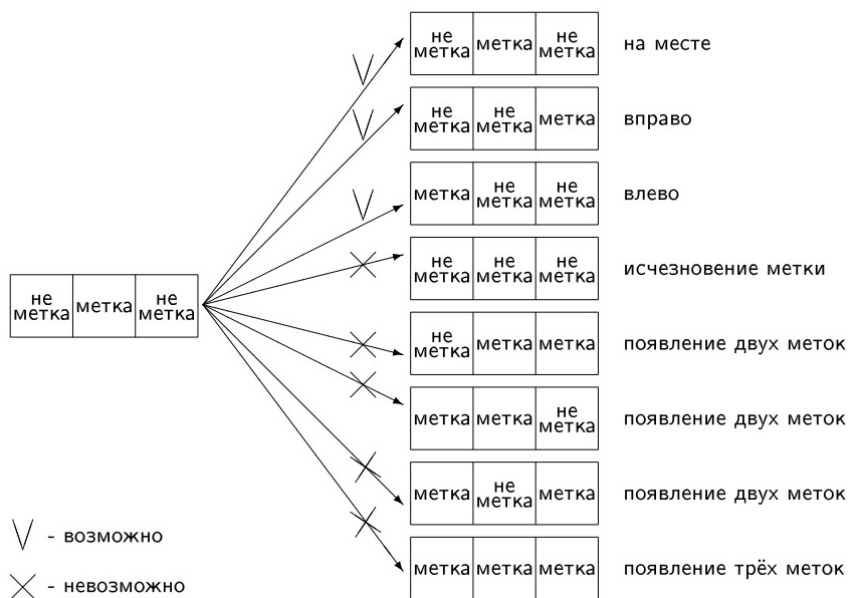


Рис. 1.

В данной статье показано, что законы движения из S , обладающие тем свойством, что в них не встречается двух подряд идущих символов f , можно реализовать клеточным автоматом с пятью состояниями, причем оценка на количество состояний не улучшаема.

Ранее Титовой был получен аналогичный результат для законов движения без движения назад.

Теорема 1 (Титовой [8]). Пусть $F = ((sf) \vee (s))^\infty$ — множество законов движения, состоящих из элементов множества $\{sf, s\}$. Тогда мощность наименьшего множества состояний клеточного автомата, необходимого для осуществления любого закона из F равна 4.

Результат, полученный в этой работе, по сути является расширением результата Титовой.

Теорема 2. Пусть $S = ((sf) \vee (s) \vee (b))^\infty$ — множество законов движения, состоящих из элементов множества $\{sf, s, b\}$ таких, что в префиксе S любой длины количество символов b не превышает количества символов f . Тогда мощность наименьшего множества состояний клеточного автомата, необходимого для осуществления любого закона из S равна 5.

3. Вспомогательные определения

Перед доказательством основных утверждений введем ряд вспомогательных определений и обозначений.

Пусть $\alpha \in S$ — закон движения, где $\alpha = \alpha_1\alpha_2\dots$, $\alpha_i \in \{sf, s, b\}$, $i = 1, 2, 3, \dots$

Обозначим $\overline{\alpha_n} = \alpha_1\dots\alpha_n$, $|\overline{\alpha_n}| = n$.

Определим функцию, идентифицирующую символ в i -ой позиции закона движения:

$$I(\alpha_i = a) = \begin{cases} 1 & \text{если } \alpha_i = a, \\ 0 & \text{если } \alpha_i \neq a. \end{cases}$$

Теперь используем её для подсчёта количества символов $\alpha_i \in \{sf, s, b\}$ в префиксе закона движения длины n :

$$\begin{aligned} I_{sf}(\bar{\alpha}_n) &= \sum_{i=1}^n I(\alpha_i = sf), \\ I_b(\bar{\alpha}_n) &= \sum_{i=1}^n I(\alpha_i = b), \\ I_s(\bar{\alpha}_n) &= \sum_{i=1}^n I(\alpha_i = s). \end{aligned}$$

Отсюда можем определить позицию метки на экране

$$d(\bar{\alpha}_n) = I_{sf}(\bar{\alpha}_n) - I_b(\bar{\alpha}_n) \geq 0$$

и количество тактов с начала движения

$$t(\bar{\alpha}_n) = I_s(\bar{\alpha}_n) + I_b(\bar{\alpha}_n) + 2I_{sf}(\bar{\alpha}_n)$$

для префикса закона движения длины n .

4. Законы движения со скоростью движения вперёд $1/2$

Рассмотрим клеточный автомат K . Состояние клетки в следующий такт определяется её состоянием в настоящий момент, а также состояниями соседних с ней слева и справа клеток. Множество состояний автомата, прикрепленного к клетке, имеет вид $Q = \{0, 1, 2, 3, 4\}$, множество меток — $L = \{1, 2\}$. Функция переходов будет следующая:

$$\begin{aligned} \varphi(1, a, b) &= 0, \text{ где } a, b \in Q \setminus L, \\ \varphi(2, a, b) &= 1, \text{ где } a, b \in Q \setminus L, \\ \varphi(a, b, c) &= a, \text{ где } a \in Q \setminus L, b \in \{0, 4\}, c \in Q, \\ \varphi(0, 1, a) &= 1, \text{ где } a \in Q \setminus L, \\ \varphi(3, 1, a) &= 0, \text{ где } a \in Q \setminus L, \\ \varphi(4, 1, a) &= 2, \text{ где } a \in Q \setminus L, \\ \varphi(a, 2, b) &= a, \text{ где } a, b \in Q \setminus L, \\ \varphi(a, 3, b) &= a, \text{ где } a \in Q \setminus L, b \in \{0, 2, 3, 4\}, \\ \varphi(a, 3, 1) &= 1, \text{ где } a \in Q \setminus L, \\ \varphi(a, b, c) &= 0, \text{ где два из трёх: } a \text{ и } b, \text{ или } b \text{ и } c, \text{ или } a \text{ и } c \in L. \end{aligned}$$

Рассмотрим пример. Реализуем закон движения $s f s f b b$. Для реализации этого закона входная последовательность будет $2 4 4 3 0 3 0$. Движение точки на экране (1 и 2, выделенные жирным шрифтом) изображено на рисунке 2. Строки — моменты времени.

2	0	0	0
4	1	0	0
4	2	0	0
3	4	1	0
0	3	2	0
3	0	3	1
0	3	1	0
0	1	0	0

Рис. 2.

Изначально на автомат подается 2, после чего 2 переходит в 1; см. правило $\varphi(2, a, b) = 1$, где $a, b \in Q \setminus L$.

Положение метки 1 не меняется, пока автомат слева от неё не примет состояние 3 [кодировка элемента b из закона движения, т.е. движение назад] или 4 [кодировка элемента f из закона движения, т.е. движение вперед]. В случае состояния 4 она меняется на метку 2, четвёрка при этом пропадает [см. законы $\varphi(4, 1, a) = 2$, где $a \in Q \setminus L$ и $\varphi(a, b, c) = a$, где $a \in Q \setminus L$, $b \in \{0, 4\}$, $c \in Q$] (рисунок 3).



здесь $* \in Q \setminus L$

Рис. 3.

Метка 2 исчезает в следующий такт, одновременно с этим справа от нее появляется метка 1 [законы $\varphi(2, a, b) = 1$, где $a, b \in Q \setminus L$ и $\varphi(a, 2, b) = a$, где $a, b \in Q \setminus L$] (рисунок 4).

Если же слева от метки 1 появляется сигнал 3, то в следующий такт метка займёт его место, тем самым двигаясь назад [законы $\varphi(a, 3, 1) = 1$, где $a \in Q \setminus L$ и $\varphi(3, 1, a) = 0$, где $a \in Q \setminus L$]. Сигналы 3 и 4 являются вспомогательными, до встречи с меткой они движутся со скоростью 1 [$\varphi(a, b, c) = a$, где $a \in Q \setminus L$, $b \in \{0, 4\}$, $c \in Q$]. Толкающий сигнал 4

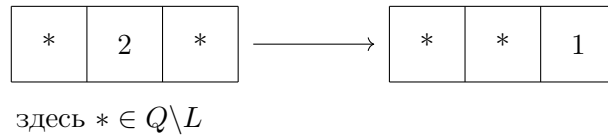


Рис. 4.

можно подавать с любой частотой, а за притягивающим сигналом 3, если метка уже появилась на экране, должен следовать ещё один сигнал, например, 0 [это нужно для того, чтобы управляющие сигналы вовремя оказывались рядом с меткой, что видно на рисунке 2].

Движение вперёд изображено на рисунке 5а. То есть, чтобы осуществить движение вперёд со скоростью $1/2$, нужно подать на вход столько четвёрок подряд, сколько в законе движения символов f .

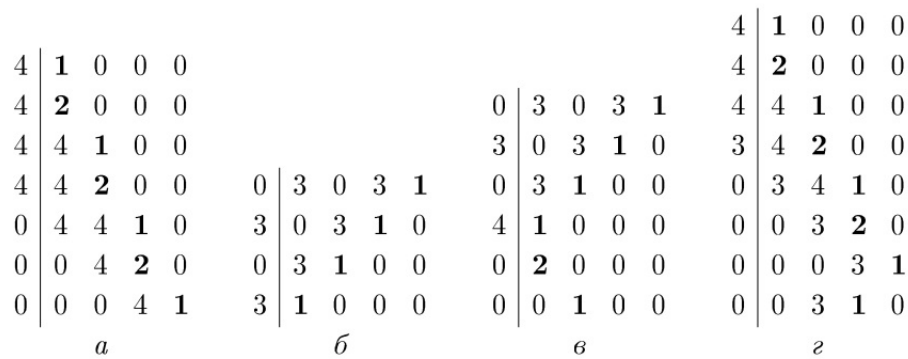


Рис. 5.

Движение назад изображено на рисунке 5б. Чтобы осуществить движение назад со скоростью 1, нужно подать на вход последовательность $(30)^m$, где m — количество элементво b в данном куске закона движения.

Если после b в законе движения следует sf (сразу f следовать не может), то после $3\ 0$ нужно подать на вход 4 (рисунок 5в). А если за f следует b , то сразу за 4 на вход нужно подавать 3 (рисунок 5г).

Для того, чтобы метка оставалась на месте, на вход подаются нули.

Этим способом точку можно двигать по любому наперед заданному закону движения из множества S .

То есть для клеточного автомата K с пятью состояниями существует способ реализации всех законов движения из S . Докажем это строго.

Лемма 1. Пусть S — множество законов движения, состоящих из элементов множества $\{sf, s, b\}$ таких, что в префиксе S любой длины количество символов b не превышает количества символов f . И пусть дано слово $\alpha = \alpha_1 \dots \alpha_n \dots \in S$, сопоставим ему управляющую последовательность — слово $\beta = 2\beta_1 \dots \beta_n \dots$, где

$$\beta_i = \begin{cases} 0, & \text{если } \alpha_i = s, \\ 30, & \text{если } \alpha_i = b, \\ 4, & \text{если } \alpha_i = sf. \end{cases}$$

Тогда, если на управляющий вход клеточного автомата K подавать последовательность β , то справедливы следующие утверждения:

1) в момент времени $t(\overline{\alpha_{n-1}})$ в позиции $d(\overline{\alpha_{n-1}}) - 1$ будет находиться символ β_n^* , где

$$\beta_n^* = \begin{cases} 0, & \text{если } \alpha_n = s, \\ 3, & \text{если } \alpha_n = b, \\ 4, & \text{если } \alpha_n = sf, \end{cases}$$

2) в момент времени $t(\overline{\alpha_n})$ метка на экране будет в позиции $d(\overline{\alpha_n})$.

Доказательство. Воспользуемся формулами:

$$\begin{aligned} I_{sf}(\overline{\alpha_n}) &= \sum_{i=1}^n I(\alpha_i = sf), \\ I_b(\overline{\alpha_n}) &= \sum_{i=1}^n I(\alpha_i = b), \\ I_s(\overline{\alpha_n}) &= \sum_{i=1}^n I(\alpha_i = s), \\ d(\overline{\alpha_n}) &= I_{sf}(\overline{\alpha_n}) - I_b(\overline{\alpha_n}) \geq 0, \\ t(\overline{\alpha_n}) &= I_s(\overline{\alpha_n}) + I_b(\overline{\alpha_n}) + 2I_{sf}(\overline{\alpha_n}). \end{aligned}$$

Доказательство будем вести индукцией по длине n префикса закона движения.

Базис индукции: $n = 1$ и $n = 2$. Возможны следующие случаи.

а) $n = 1$. $\overline{\alpha_1} = \alpha_1 = b$ — данный закон движения не из класса S , $n = 2$ рассматривать не нужно.

б) $n = 1$. $\overline{\alpha_1} = \alpha_1 = s$, $t(\overline{\alpha_1}) = 1$, $d(\overline{\alpha_1}) = 0$.

Подадим на вход последовательность 20. Согласно функции переходов будем наблюдать картину, изображённую на рисунке 6а. В дальнейшем символом * будем обозначать произвольный символ из множества $Q \setminus L$.

$$\begin{array}{ccc}
2 & & 2 \\
2 & \left| \begin{array}{ccc} 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 \\ * & \mathbf{1} & 0 & 0 \end{array} \right. & 0 & \left| \begin{array}{ccc} 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 \\ * & \mathbf{1} & 0 & 0 \end{array} \right. & 4 & \left| \begin{array}{ccc} 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 \\ * & \mathbf{2} & 0 & 0 \\ * & * & \mathbf{1} & 0 \end{array} \right. \\
a & & б & & в
\end{array}$$

Рис. 6.

В момент времени $t = 0$ (момент появления метки на экране) в позиции $d = -1$, то есть на входе, будет находиться $\beta_1^* = 0$. В момент времени $t(\overline{\alpha_1}) = 1$ метка на экране будет находиться в позиции $d(\overline{\alpha_1}) = 0$. Верно.

- б.1) $n = 2$. $\overline{\alpha_2} = \alpha_1\alpha_2 = sb$ — данный закон движения не из класса S .
б.2) $n = 2$. $\overline{\alpha_2} = \alpha_1\alpha_2 = ss$, $t(\overline{\alpha_2}) = 2$, $d(\overline{\alpha_2}) = 0$.

Подадим на вход последовательность 200. Согласно функции переходов будем наблюдать картину, изображённую на рисунке 6б.

В момент времени $t(\overline{\alpha_1}) = 1$ в позиции $d(\overline{\alpha_1}) - 1 = -1$, то есть на входе, будет находиться $\beta_2^* = 0$. В момент времени $t(\overline{\alpha_2}) = 2$ метка на экране будет находиться в позиции $d(\overline{\alpha_2}) = 0$. Верно.

- б.3) $n = 2$. $\overline{\alpha_2} = \alpha_1\alpha_2 = ssf$, $t(\overline{\alpha_2}) = 3$, $d(\overline{\alpha_2}) = 1$.

Подадим на вход последовательность 204. Согласно функции переходов будем наблюдать картину, изображённую на рисунке 6в.

В момент времени $t(\overline{\alpha_1}) = 1$ в позиции $d(\overline{\alpha_1}) - 1 = -1$, то есть на входе, будет находиться $\beta_2^* = 4$. В момент времени $t(\overline{\alpha_2}) = 3$ метка на экране будет находиться в позиции $d(\overline{\alpha_2}) = 1$. Верно.

- в) $n = 1$. $\overline{\alpha_1} = \alpha_1 = sf$, $t(\overline{\alpha_1}) = 2$, $d(\overline{\alpha_1}) = 1$.

Подадим на вход последовательность 24. Согласно функции переходов будем наблюдать картину, изображённую на рисунке 7а.

$$\begin{array}{cccc}
2 & & 2 & & 2 & & 2 \\
2 & \left| \begin{array}{ccc} 0 & 0 & 0 \\ 4 & \mathbf{1} & 0 & 0 \\ * & \mathbf{2} & 0 & 0 \\ * & * & \mathbf{1} & 0 \end{array} \right. & 4 & \left| \begin{array}{ccc} 0 & 0 & 0 \\ 3 & \mathbf{2} & 0 & 0 \\ 0 & 3 & \mathbf{1} & 0 \\ * & \mathbf{1} & 0 & 0 \end{array} \right. & 4 & \left| \begin{array}{ccc} 0 & 0 & 0 \\ 4 & \mathbf{1} & 0 & 0 \\ 0 & \mathbf{2} & 0 & 0 \\ * & 0 & \mathbf{1} & 0 \\ * & * & \mathbf{1} & 0 \end{array} \right. & 4 & \left| \begin{array}{ccc} 0 & 0 & 0 & 0 \\ \mathbf{1} & 0 & 0 & 0 \\ \mathbf{2} & 0 & 0 & 0 \\ 4 & \mathbf{1} & 0 & 0 \\ * & * & \mathbf{2} & 0 & 0 \\ * & * & * & \mathbf{1} & 0 \end{array} \right. \\
a & & б & & в & & г
\end{array}$$

Рис. 7.

В момент времени $t = 0$ (момент появления метки на экране) в позиции $d = -1$, то есть на входе, будет находиться $\beta_1^* = 4$. В момент времени $t(\overline{\alpha_1}) = 2$ метка на экране будет находиться в позиции $d(\overline{\alpha_1}) = 1$. Верно.

в.1) $n = 2$. $\overline{\alpha_2} = \alpha_1\alpha_2 = sfb$, $t(\overline{\alpha_2}) = 3$, $d(\overline{\alpha_2}) = 0$.

Подадим на вход последовательность 2430. Согласно функции переходов будем наблюдать картину, изображённую на рисунке 7б.

В момент времени $t(\overline{\alpha_1}) = 2$ в позиции $d(\overline{\alpha_1}) - 1 = 0$, будет находиться $\beta_2^* = 3$. В момент времени $t(\overline{\alpha_2}) = 3$ метка на экране будет находиться в позиции $d(\overline{\alpha_2}) = 0$. Верно.

в.2) $n = 2$. $\overline{\alpha_2} = \alpha_1\alpha_2 = sfs$, $t(\overline{\alpha_2}) = 3$, $d(\overline{\alpha_2}) = 1$.

Подадим на вход последовательность 240. Согласно функции переходов будем наблюдать картину, изображённую на рисунке 7в.

В момент времени $t(\overline{\alpha_1}) = 2$ в позиции $d(\overline{\alpha_1}) - 1 = 0$, будет находиться $\beta_2^* = 0$. В момент времени $t(\overline{\alpha_2}) = 3$ метка на экране будет находиться в позиции $d(\overline{\alpha_2}) = 1$. Верно.

в.3) $n = 2$. $\overline{\alpha_2} = \alpha_1\alpha_2 = sfsf$, $t(\overline{\alpha_2}) = 4$, $d(\overline{\alpha_2}) = 2$.

Подадим на вход последовательность 244. Согласно функции переходов будем наблюдать картину, изображённую на рисунке 7г.

В момент времени $t(\overline{\alpha_1}) = 2$ в позиции $d(\overline{\alpha_1}) - 1 = 0$, будет находиться $\beta_2^* = 4$. В момент времени $t(\overline{\alpha_2}) = 4$ метка на экране будет находиться в позиции $d(\overline{\alpha_2}) = 2$. Верно.

Индуктивный переход. Пусть при $n = k - 1$ утверждение выполнено.

Тогда в момент времени $t(\overline{\alpha_{k-1}})$ метка 1 (видно в доказательстве базы индукции) будет находиться в позиции $d(\overline{\alpha_{k-1}})$. А в момент времени $t(\overline{\alpha_{k-2}})$ в позиции $d(\overline{\alpha_{k-2}}) - 1$ будет находиться символ β_{k-1}^* .

Докажем, что при $n = k$ утверждение также выполняется.

1) Сигнал (3 или 4) до встречи с меткой движется со скоростью 1 — это следует из рассматриваемой функции переходов.

Рассмотрим момент времени $t(\overline{\alpha_{k-1}})$. Единица находится в позиции $d(\overline{\alpha_{k-1}})$.

а) Если последним в законе движения было движение вправо (sf), значит, в момент времени $t(\overline{\alpha_{k-1}}) - 2$ четвёрка стояла в позиции $d(\overline{\alpha_{k-1}}) - 2$, а β_k^* — в $d(\overline{\alpha_{k-1}}) - 3$. То есть в $t(\overline{\alpha_{k-1}})$ β_k^* будет в $d(\overline{\alpha_{k-1}}) - 1$ (рисунок 8а).

б) Если последним в законе движения было движение влево (b), значит, в момент времени $t(\overline{\alpha_{k-1}}) - 1$ тройка стояла в позиции $d(\overline{\alpha_{k-1}})$, 0 — в позиции $d(\overline{\alpha_{k-1}}) - 1$, а β_k^* — в $d(\overline{\alpha_{k-1}}) - 2$. То есть в $t(\overline{\alpha_{k-1}})$ β_k^* будет в $d(\overline{\alpha_{k-1}}) - 1$ (рисунок 8б).

в) Если последней в законе движения была остановка (s), значит, в момент времени $t(\overline{\alpha_{k-1}}) - 1$ ноль стоял в позиции $d(\overline{\alpha_{k-1}}) - 1$, а β_k^* — в $d(\overline{\alpha_{k-1}}) - 2$. То есть в $t(\overline{\alpha_{k-1}})$ β_k^* будет в $d(\overline{\alpha_{k-1}}) - 1$ (рисунок 8в).

$$\begin{array}{ccccc}
* & \beta_k^* & 4 & \mathbf{1} & * & * & * \\
* & * & \beta_k^* & \mathbf{2} & 0 & * & * \\
* & * & * & \beta_k^* & \mathbf{1} & 0 & * \\
& & & a & & &
\end{array}
\quad
\begin{array}{ccccc}
* & \beta_k^* & 0 & 3 & \mathbf{1} & * & * \\
* & * & \beta_k^* & \mathbf{1} & 0 & 0 & * \\
& & & b & & &
\end{array}
\quad
\begin{array}{ccccc}
* & \beta_k^* & 0 & \mathbf{1} & * & * \\
* & * & \beta_k^* & \mathbf{1} & 0 & * \\
& & & в & &
\end{array}$$

Рис. 8.

2) В момент времени $t(\overline{\alpha_{k-1}})$ метка 1 находится в позиции $d(\overline{\alpha_{k-1}})$.

В позиции $d(\overline{\alpha_{k-1}}) - 1$ будет находиться символ β_k^* (согласно утверждению пункта 1).

Докажем, что в момент времени $t(\overline{\alpha_k})$ метка 1 будет находиться в позиции $d(\overline{\alpha_k})$.

Отдельно рассмотрим три случая для каждого возможного значения α_k .

а) $\alpha_k = s$. В этом случае выполнены соотношения:

$$\begin{aligned}
I_s(\overline{\alpha_k}) &= I_s(\overline{\alpha_{k-1}}) + 1, \\
I_{sf}(\overline{\alpha_k}) &= I_{sf}(\overline{\alpha_{k-1}}), \\
I_b(\overline{\alpha_k}) &= I_b(\overline{\alpha_{k-1}}), \\
t(\overline{\alpha_k}) &= t(\overline{\alpha_{k-1}}) + 1, \\
d(\overline{\alpha_k}) &= d(\overline{\alpha_{k-1}}).
\end{aligned}$$

Согласно функции переходов получим картину, изображённую на рисунке 9а.

$$\begin{array}{ccccc}
& & & & * & 4 & \mathbf{1} & * & * & * \\
* & 0 & \mathbf{1} & * & * & * & 0 & 3 & \mathbf{1} & * & * \\
* & * & \mathbf{1} & 0 & * & * & * & \mathbf{1} & 0 & 0 & * \\
& & & a & & & & б & & & в
\end{array}$$

Рис. 9.

В момент времени $t(\overline{\alpha_{k-1}})$ в позиции $d(\overline{\alpha_{k-1}})$ находится метка 1, в позиции $d(\overline{\alpha_{k-1}}) - 1$ находится символ $\beta_k^* = 0$.

Обозначим $q(d)$ символ в позиции d . В момент времени $t(\overline{\alpha_k})$ символ в позиции $d(\overline{\alpha_k})$ будет равен

$$q(d(\overline{\alpha_k})) = \varphi(\beta_k^*, 1, a) = \varphi(0, 1, a) = 1, \text{ где } a \in Q \setminus L.$$

б) $\alpha_k = b$. В этом случае выполнены соотношения:

$$\begin{aligned} I_b(\overline{\alpha_k}) &= I_b(\overline{\alpha_{k-1}}) + 1, \\ I_{sf}(\overline{\alpha_k}) &= I_{sf}(\overline{\alpha_{k-1}}), \\ I_s(\overline{\alpha_k}) &= I_s(\overline{\alpha_{k-1}}), \\ t(\overline{\alpha_k}) &= t(\overline{\alpha_{k-1}}) + 1, \\ d(\overline{\alpha_k}) &= d(\overline{\alpha_{k-1}}) - 1. \end{aligned}$$

Согласно функции переходов получим картину, изображённую на рисунке 9б.

В момент времени $t(\overline{\alpha_{k-1}})$ в позиции $d(\overline{\alpha_{k-1}})$ находится метка 1, в позиции $d(\overline{\alpha_{k-1}}) - 1$ находится символ $\beta_k^* = 3$.

В момент времени $t(\overline{\alpha_k})$ символ в позиции $d(\overline{\alpha_k})$ будет равен

$$q(d(\overline{\alpha_k})) = \varphi(a, \beta_k^*, 1) = \varphi(a, 3, 1) = 1, \text{ где } a \in Q \setminus L.$$

в) $\alpha_k = sf$. В этом случае выполнены соотношения:

$$\begin{aligned} I_{sf}(\overline{\alpha_k}) &= I_{sf}(\overline{\alpha_{k-1}}) + 1, \\ I_b(\overline{\alpha_k}) &= I_b(\overline{\alpha_{k-1}}), \\ I_s(\overline{\alpha_k}) &= I_s(\overline{\alpha_{k-1}}), \\ t(\overline{\alpha_k}) &= t(\overline{\alpha_{k-1}}) + 2, \\ d(\overline{\alpha_k}) &= d(\overline{\alpha_{k-1}}) + 1. \end{aligned}$$

Согласно функции переходов получим картину, изображённую на рисунке 9в.

В момент времени $t(\overline{\alpha_{k-1}})$ в позиции $d(\overline{\alpha_{k-1}})$ находится метка 1, в позиции $d(\overline{\alpha_{k-1}}) - 1$ находится символ $\beta_k^* = 4$.

В момент времени $t(\overline{\alpha_{k-1}}) + 1$ символ в позиции $d(\overline{\alpha_{k-1}})$ будет равен

$$q(d(\overline{\alpha_{k-1}})) = \varphi(\beta_k^*, 1, a) = \varphi(4, 1, a) = 2, \text{ где } a \in Q \setminus L.$$

В момент времени $t(\overline{\alpha_{k-1}}) + 2 = t(\overline{\alpha_k})$ символ в позиции $d(\overline{\alpha_{k-1}}) + 1 = d(\overline{\alpha_k})$ будет равен

$$q(d(\overline{\alpha_k})) = \varphi(2, a, b) = 1, \text{ где } a, b \in Q \setminus L.$$

Мы доказали, что в момент времени $t(\overline{\alpha_k})$ метка 1 будет находиться в позиции $d(\overline{\alpha_k})$ для всех возможных вариантов движения. Значит, утверждение 2 леммы верно. □

Лемма 2 (Верхняя оценка). Пусть S — множество законов движения, состоящих из элементов множества $\{sf, s, b\}$ таких, что в префиксе S любой длины количество символов b не превышает количества символов f . Тогда мощность множества состояний клеточного автомата $|Q(S)| \leq 5$.

Доказательство. Пусть множество состояний $Q = E_5 = \{0, 1, 2, 3, 4\}$, множество меток $L = \{1, 2\}$ [это те состояния, при которых считается, что точка видна]. Рассмотрим клеточный автомат K .

Подадим на вход слово β , его первый символ 2, в следующий такт в начале экрана появится метка 1:

$$\varphi(2, a, b) = 1, \text{ где } a \in \{0, 3, 4\}, b \in Q \setminus L.$$

Согласно лемме 1, реализуем закон движения $S = \alpha$. □

Лемма 3. Пусть S — множество законов движения, состоящих из элементов множества $\{sf, s, b\}$ таких, что в префиксе S любой длины количество символов b не превышает количества символов f . Не существует универсального экрана с 4 состояниями клеточного автомата, среди которых либо одна метка, либо три, такого, что любой закон движения из S можно реализовать на этом экране.

Доказательство. Покажем, что 4 состояний не достаточно.

Предположим, что $|Q| = 4$, $Q = \{0, 1, 2, 3\}$.

1) $2, 3 \in L$ ($L = \{1, 2, 3\}$).

То есть у нас есть три метки, и ни одного сигнала. В любой момент после начала и до конца движения на экране находится ровно одна метка.

Рассмотрим произвольный момент времени. Пусть на экране находится метка 1, в следующий момент времени может измениться следующее: она может остаться на месте, переместиться вправо или влево, в то же время она может остаться меткой 1, или стать меткой 2, либо 3. Таким образом, для метки 1 есть 9 вариантов развития событий в следующий момент.

То же самое верно для меток 2 и 3. То есть, имея три метки и ни одного сигнала, возможно реализовать не более 9^3 законов движения, а нам требуется реализовать континуум, поскольку в S континуум законов движения.

2) $2, 3 \in Q \setminus L$ ($L = \{1\}$).

Пусть $F \subset S$ — множество законов движения, состоящих из элементов множества $\{sf, s\}$. Попробуем реализовать F .

Метка всего одна. Все движения вперед можно осуществить только из $\varphi(1, a, b) = 1$, $a, b \in \{0, 2, 3\}$, т.е. из предобработки. Под предобработкой

здесь понимается конечное множество сигналов, поданных на вход до появления метки на экране.

К моменту появления метки на экране там уже находится какая-то конфигурация, состоящая из элементов множества $\{0, 2, 3\}$. Таких конфигураций счётное число, а законов движения, которые надо реализовать, континуум. \square

Лемма 4. Пусть S — множество законов движения, состоящих из элементов множества $\{sf, s, b\}$ таких, что в префиксе S любой длины количество символов b не превышает количества символов f . Не существует универсального экрана с 4 состояниями клеточного автомата, среди которых две метки, такого, что любой закон движения из S можно реализовать на этом экране.

Доказательство. Рассмотрим случай, когда у нас 2 метки — $1, 2 \in L$.

1) Пусть x — префикс закона движения. Обозначим через $l(x)$ длину управляющей последовательности для этого префикса.

$$l(x) = 2I_b(x) + I_s(x) + I_{sf}(x)$$

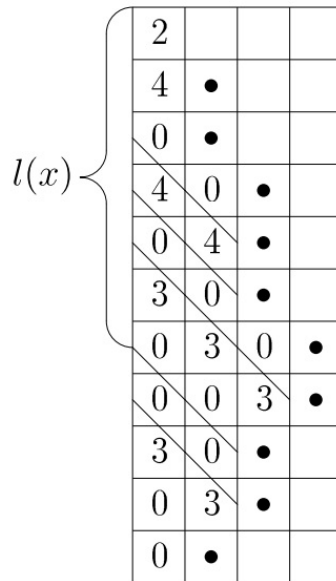


Рис. 10.

На рисунке 10 изображено движение метки на экране и управляющая последовательность для осуществления этого движения. Сигналы

до встречи с меткой движутся со скоростью 1 (наклонные линии на рисунке). В общем случае сигналы могут двигаться со скоростью, не большей, чем 1. Легко заметить, что для того, чтобы управляющие сигналы вовремя оказывались рядом с меткой, символ b закона движения должен соответствовать двум символам управляющей последовательности, символ s — одному символу управляющей последовательности, sf — также одному символу.

2) Сначала рассмотрим случай, когда предобработка отсутствует, то есть в момент появления метки на экране правее метки все автоматы находятся в состоянии покоя.

Пусть W — произвольное множество слов.

Если k — натуральное число, то через W^k обозначим множество слов $\{w_1 w_2 \dots w_k : w_i \in W, i = 1, \dots, k\}$.

Обозначим $f_W(n) = |\{x | x \in W, l(x) = n\}|$ — количество различных законов движения из множества W с заданной длиной n управляющей последовательности.

В доказательстве будем применять метод производящих функций [12]. Пусть $F_W(t) = \sum_{n=0}^{\infty} f_W(n)t^n$ — производящая функция для $f_W(n)$.

Пусть $S^* \subset S$ — множество законов движения, состоящих из элементов множества $\{s, sf, sfb\}$.

Так как управляющая последовательность состоит из двух символов, то при помощи управляющей последовательности длины n возможно реализовать 2^n законов движения.

Докажем, что существует такой номер N , что для любого $n > N$ выполняется $f_{S^*}(n) > 2^n$.

Рассмотрим множество $H_x = \{x\}$, где $x \in \{s, sf, sfb\}$. Тогда $F_{H_x}(t) = t^{l(x)}$, где

$$l(x) = \begin{cases} 3 & \text{если } x = sfb, \\ 1 & \text{если } x \in \{s, sf\}. \end{cases}$$

Пусть $H = \{s, sf, sfb\}$ — множество законов движения из S^* , состоящих из одного элемента.

Тогда $F_H(t) = t^3 + t$ и $F_{H^k}(t) = (t^3 + 2t)^k$.

Пусть

$$H^* = \bigcup_{k=1}^{\infty} H^k,$$

тогда производящая функция для множества H^* равна:

$$F_{H^*}(t) = \sum_{k=1}^{\infty} F_{H^k}(t) = \sum_{k=1}^{\infty} (t^3 + 2t)^k = \frac{1}{1 - t^3 - 2t}.$$

Очевидно, что H^* совпадает с S^* , и, следовательно, $F_{H^*}(t)$ совпадает с производящей функцией для $f_{S^*}(n)$, то есть

$$F_{H^*}(t) = F_{S^*}(t) = \sum_{n=0}^{\infty} f_{S^*}(n)t^n.$$

Получим оценку радиуса сходимости ряда $F_{S^*}(t)$.
С одной стороны по признаку Коши:

$$R = \lim_{n \rightarrow \infty} \frac{1}{f_{S^*}(n)^{1/n}}.$$

С другой стороны, так как $F_{S^*}(t) = 1/(1-t^3-2t)$, то $R \leq R^* = \min |t_i|$, где t_i — корни уравнения $1-t^3-2t=0$.

Из графика функции $1-t^3-2t$ можно получить оценку $R^* \approx 0.45 < \frac{1}{2}$.
Таким образом:

$$R = \lim_{n \rightarrow \infty} \frac{1}{f(n)^{1/n}} \leq \min |t_i| = R^* < \frac{1}{2}.$$

Откуда следует, что существует такой номер N , что для любого $n > N$ выполняется

$$f(n)^{1/n} \geq \frac{1}{R^*} \implies f(n) \geq \left(\frac{1}{R^*}\right)^n > 2^n.$$

То есть нужно реализовать больше законов движения, чем это возможно.

3) Пусть $f_S(n) = |\{x|x \in S, l(x) = n\}|$ — количество различных законов движения из множества S с заданной длиной n управляющей последовательности при отсутствии предобработки.

Так как $S^* \subset S$, то существует такой номер N , что для любого $n > N$ выполняется

$$f_S(n) \geq f_{S^*}(n) \geq \left(\frac{1}{R^*}\right)^n > 2^n.$$

4) Теперь рассмотрим случай, когда предобработка есть, и пусть l_p — длина последовательности сигналов, подаваемой на вход на этапе предобработки.

Пусть e_0, e_1, e_2, \dots — состояние экрана в конце предобработки.

Так как за время предобработки первый символ входной последовательности $\alpha_1 = 3$ не может сместиться далее чем на l_p позиций от начала экрана (в противном случае скорость его движения будет выше 1), то

$$e_i = \begin{cases} 0, & \text{если } i \geq l_p, \\ 0 \text{ или } 3, & \text{если } 0 \leq i < l_p. \end{cases}$$

Отсюда следует, что число возможных состояний экрана после преобработки равно 2^{l_p} .

Пусть n — длина управляющей последовательности, подаваемой на вход, без учета этапа преобработки.

Обозначим $v_n(l_p) = 2^{l_p} \cdot 2^n$ — максимально возможное число законов движения, которые можно реализовать с помощью управляющей последовательности длины n и входной последовательности длины l_p на этапе преобработки.

Пусть A — некоторый фиксированный закон движения из множества S , $l(A)$ — длина управляющей последовательности для этого закона движения.

Рассмотрим множество законов движения $S_A = \{x : x = AB\} \subset S$, где A — фиксированный префикс, заданный выше, B — произвольное продолжение из S .

Пусть $l(x) = l(A) + l(B)$ — длина управляющей последовательности для закона движения AB .

Пусть $f_{S_A}(n) = |\{x | x \in S_A, l(x) = l(A) + l(B) = n\}|$ — количество различных законов движения из множества S_A с заданной длиной n управляющей последовательности.

Докажем, что для любого фиксированного $l_p < \infty$ существует такой номер N , что для любого $n > N$ выполняется $f_{S_A}(n) > v_n(l_p)$.

Пусть $f_S(k) = |\{y | l(y) = k\}|$ — количество различных законов движения y из S с заданной длиной k управляющей последовательности.

Как показано в пункте 3, существует такой номер N , что для любого $n > N$ выполняется

$$f_S(k) \geq \left(\frac{1}{R^*}\right)^k.$$

Доказательство данного утверждения учитывает только последовательность шагов в законе движения. Таким образом, можно утверждать, что во-первых данная оценка остается верной при наличии преобработки и не зависит от нее. Во-вторых, при наличии некоторого фиксированного префикса оценка справедлива для $k = l(B)$, где $l(B)$ — длина продолжения.

Значит, существует такой номер N , что для любого $n > N$ справедлива оценка:

$$f_{S_A}(n) = f_S(l_B = n - l_A) \geq \left(\frac{1}{R^*}\right)^{n-l_A}.$$

Покажем, что существует такой номер N_0 , что $f_{S_A}(n)/v_n(l_p) > 1$ при $n > N_0$:

$$\lim_{n \rightarrow \infty} \frac{f_{S_A}(n)}{v_n(l_p)} \geq \lim_{n \rightarrow \infty} \left(\frac{1}{R^*}\right)^{n-l(A)} \cdot \frac{1}{2^{l_p} \cdot 2^n} = \frac{(R^*)^{l(A)}}{2^{l_p}} \lim_{n \rightarrow \infty} \left(\frac{1}{2R^*}\right)^n = \infty.$$

А значит существует такой номер N_0 , что для любого $n > N_0$ выполняется $f_{S_A}(n) > v_n(l_p)$.

Так как префикс A был выбран произвольно, то для любого l_p и любого $A \in S$ существует $B \in S$ такое, что $x = AB$ не реализуется при заданной длине предобработки l_p .

5) Покажем, что существует как минимум один закон движения из множества S , который нельзя реализовать при любой предобработке.

Пусть $l_p = 0$ (предобработка отсутствует). Тогда существует $B_0 \in S^* \subset S$ такой, что B_0 нельзя реализовать без предобработки (по доказанному в пункте 2).

Пусть $l_p = 1, A_1 = B_0$. Тогда существует $B_1 \in S$ такой, что B_0B_1 нельзя реализовать при $l_p = 1$ (по доказанному в пункте 4).

Но так как B_0 нельзя реализовать при $l_p = 0$, то и B_0B_1 также нельзя реализовать при $l_p = 0$. Следовательно, B_0B_1 не реализуется при любом $l_p \leq 1$.

Пусть $l_p = 2, A_2 = B_0B_1$. Тогда существует $B_2 \in S$ такой, что $A_2B_2 = B_0B_1B_2$ нельзя реализовать при $l_p = 2$ (в соответствии с пунктом 2) и при $l_p \leq 1$ (так как нельзя реализовать B_0B_1). Значит, $B_0B_1B_2$ не реализуется при любом $l_p \leq 2$.

Продолжая, аналогичным образом можем получить закон движения $\bar{B} = B_0B_1B_2 \dots B_n \dots$ такой, что для любого k $B_0B_1B_2 \dots B_k$ — не реализуется при любом $l_p \leq k$.

Так как $B_k \in S$ для любого k , то $\bar{B} \in S$.

Построенная последовательность префиксов B_k бесконечна, следовательно закон движения $\bar{B} \in S$ нельзя реализовать ни с какой длиной предобработки. □

Лемма 5 (Нижняя оценка). *Пусть S — множество законов движения, состоящих из элементов множества $\{sf, s, b\}$ таких, что в префиксе S любой длины количество символов b не превышает количества символов f . Тогда мощность множества состояний клеточного автомата, необходимого для осуществления всех законов из S : $|Q(S)| > 4$.*

Доказательство. Следует из лемм 3 и 4. □

Из лемм 2 и 5 следует теорема 2.

5. Заключение

В работе доказаны верхняя и нижняя оценки на число состояний клеточного автомата, реализующего множество законов S . Необходимо и достаточно пяти состояний.

Список литературы

- [1] Фон Нейман Дж., *Теория самовоспроизводящихся автоматов*, Мир, Москва, 1971.
- [2] Мур Э.Ф. Математические модели самовоспроизведения, *Математические проблемы в биологии*, Мир, Москва, 1966.
- [3] Gardner M. The fantastic combinations of John Conway's new solitaire game "life", *Scientific American*, 1970, № 4.
- [4] Wolfram S., *A New Kind of Science*, Wolfram Media, Champaign, Illinois, 2002.
- [5] Кудрявцев В.Б., Подколзин А.С., Болотов А.А., *Основы теории однородных структур*, «Наука», Москва, 1990.
- [6] Титова Е.Е., “Конструирование изображений клеточными автоматами”, *Интеллектуальные системы*, **12:2** (2008), 105–121.
- [7] Титова Е.Е., “Линейное по времени конструирование изображений клеточными автоматами”, *Интеллектуальные системы*, **16:2** (2012), 215–234.
- [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*, **22:3** (2018), 105–125.
- [10] Кудрявцев В.Б., Алешин С.В., Подколзин А.С., *Введение в теорию автоматов*, «Наука», Москва, 1985, 320 с.
- [11] Кудрявцев В.Б., Гасанов Э.Э., Подколзин А.С., *Основы теории интеллектуальных систем: в 4 кн. Книга четвёртая. Теория автоматов.*, «Издательские решения», Москва, 2018, 302 с.
- [12] Гаврилов Г.П., Сапоженко А.А., *Задачи и упражнения по дискретной математике.*, ФИЗМАТЛИТ, Москва, 2005, 416 с.

References

- [1] Von Neumann, John, *Theory of self-reproducing automata*, Urbana, University of Illinois Press, 1966
- [2] Edward F. Moore Machine models of self-reproduction, *Mathematical Problems in the Biological Sciences*, AMERICAN MATHEMATICAL SOCIETY, U.S.A., 1962
- [3] Gardner M. The fantastic combinations of John Conway's new solitaire game "life", *Scientific American*, 1970, № 4
- [4] Wolfram S., *A New Kind of Science*, Wolfram Media, Champaign, Illinois, 2002
- [5] Kudryavtsev V.B., Podkolzin A.S., Bolotov A.A., *Osnovy teorii odnorodnykh struktur [Fundamentals of the theory of homogeneous structures]*, «Nauka», Moscow, 1990 (in Russian)
- [6] Titova E.E., “Konstruirovaniye izobrazheniy kletochnymi avtomatami [Images construction by cellular automata]”, *Intelligent Systems*, **12:2** (2008), 105–121 (in Russian)
- [7] Titova E.E., “Lineynoe po vremeni konstruirovaniye izobrazheniy kletochnymi avtomatami [Linear time images construction by cellular automata]”, *Intelligent Systems*, **16:2** (2012), 215–234 (in Russian)

- [8] Titova E.E., “Konstruirovaniye dvizhushchikhsya izobrazheniy kletochnymi avtomatami [Moving images construction by cellular automata]”, *Intelligent Systems*, **18**:1 (2014), 153–180 (in Russian)
- [9] Kalachev G.V., Titova E.E., “O mere mnozhestva zakonov dvizheniya tochki, realizuemykh kletochnymi avtomatami [On the measure of the set of a point motion laws implemented by cellular automata]”, *Intelligent Systems*, **22**:3 (2018), 105–125 (in Russian)
- [10] Kudryavtsev V.B., Aleshin S.V., Podkolzin A.S., *Vvedeniye v teoriyu avtomatov [Introduction to automata theory]*, «Nauka», Moscow, 1985 (in Russian), 320 pp.
- [11] Kudryavtsev V.B., Gasanov E.E., Podkolzin A.S., *Osnovy teorii intellektual'nykh sistem: v 4 kn. Kniga chetvertaya. Teoriya avtomatov. [Foundations of the theory of intelligent systems: in 4 books. Book four. Automata theory.]*, «Izdatel'skie resheniya», Moscow, 2018 (in Russian), 302 pp.
- [12] Gavrilov G.P., Sapozhenko A.A., *Zadachi i uprazhneniya po diskretnoy matematike [Problems and exercises in discrete mathematics]*, FIZMATLIT, Moscow, 2005 (in Russian), 416 pp.

The number of states of a universal automaton of an infinite screen that implements bidirectional motion on a ray
Kuznetsova E.V.

The paper considers the implementation of one class of laws of motion by a cellular automaton on an infinite screen. It is shown that the minimum number of states of a cellular automaton simulating the bidirectional movement of a point on a ray at which the point does not make 2 movements to the right in a row is 5.

Keywords: cellular automaton, number of states, infinite screen, bidirectional motion, image construction.

Распознавание A -полноты конечных систем линейных автоматов с добавками над кольцом двоично-рациональных чисел

Ронжин Д.В.¹

Исследуются вопросы A -полноты конечных систем линейных автоматов над кольцом двоично-рациональных чисел. Описано условие полноты конечной системы линейных автоматов, содержащей автомат, реализующий сумматор в первый такт. Доказана алгоритмическая разрешимость задачи определения принадлежности конечного множества линейных автоматов сформулированному набору предполных классов.

Ключевые слова: конечные автоматы, линейные автоматы, двоично-рациональные числа, A -полнота, предполный класс, алгоритмическая разрешимость.

1. Введение

Исследование задачи полноты систем конечных автоматов[1] связано с рассмотрением вопроса об A -полноте[2] этих систем. Также часто рассматривается задача полноты систем, содержащих добавки[3]. Интересным для изучения подклассом конечных автоматов являются линейные автоматы[4, 5], для которых описаны условия полноты конечных систем в терминах предполных классов.

Помимо автоматов, функционирующих над конечными алфавитами, интерес представляет исследование линейных автоматов, алфавиты которых являются бесконечными множествами. В работе [6] исследуются вопросы наличия полных систем по операциям композиции и суперпозиции на множестве линейных автоматов, функционирующих над полем рациональных чисел. В работе [7] исследуется сужение указанного класса до

¹Ронжин Дмитрий Владимирович — преподаватель математики в ОАНО "Новая школа e-mail: d.v.ronzhin@gmail.com.

Ronzhin Dmitry Vladimirovich — math teacher at non-profit organization "New School".

автоматов, функционирующих над подкольцом рациональных чисел - множестве двоично-рациональных чисел.

Настоящая работа посвящена дальнейшему исследованию задачи A -полноты линейных автоматов над кольцом двоично-рациональных чисел. Описано условие A -полноты систем, содержащих в качестве добавки автомат, реализующий сумматор в первый такт. Также в настоящей работе доказана алгоритмическая разрешимость задачи распознавания A -полноты конечных множеств линейных автоматов с упомянутой добавкой.

2. Вспомогательные обозначения

Кольцо двоично-рациональных чисел, которое является подкольцом в поле рациональных чисел обозначим через $\mathbb{Q}_{\frac{m}{2^n}}$:

$$\mathbb{Q}_{\frac{m}{2^n}} = \left\{ \frac{m}{2^n} \mid m \in \mathbb{Z}, n \in \mathbb{N} \right\}.$$

Далее в настоящей работе элементы $\mathbb{Q}_{\frac{m}{2^n}}$ рассматриваем в сокращенном виде. $\forall c \in \mathbb{Q}_{\frac{m}{2^n}}, c = \frac{m}{2^n}, \forall k \in \mathbb{Z}$ будем говорить что c кратно k тогда и только тогда, когда m кратно k .

$\forall l, k \in \mathbb{N}$ будем рассматривать конечные автоматы[1] с входным алфавитом $\mathbb{Q}_{\frac{l}{2^n}}$, выходным алфавитом $\mathbb{Q}_{\frac{m}{2^n}}$ и алфавитом состояний $\mathbb{Q}_{\frac{k}{2^n}}$, функции переходов и выходов являются линейными[4, 5]. Множество всех таких автоматов обозначим $L(\mathbb{Q}_{\frac{m}{2^n}})$ [6] и будем называть множеством линейных автоматов над кольцом двоично-рациональных чисел.

Заметим, что множество $L(\mathbb{Q}_{\frac{m}{2^n}})$ имеет конечный базис по операциям композиции[1], а именно:

$$\mathbf{B} = \{V_{\oplus}(x, y), V_{(-\frac{1}{2})}(x), \xi_1(x)\}, \text{ где}$$

- 1) $V_{\oplus}(x, y)$ - сумматор,
- 2) $V_{(-\frac{1}{2})}(x)$ - умножитель на число $-\frac{1}{2}$,
- 3) $\xi_1(x)$ - задержка с единичным начальным состоянием.

Аналогично[7] определим множество формальных степенных рядов над $\mathbb{Q}_{\frac{m}{2^n}}$:

$$\mathbb{Q}_{\frac{m}{2^n}}^{\infty}(\xi) = \left\{ \alpha = \sum_{i=0}^{\infty} a_i \cdot \xi^i \mid a_i \in \mathbb{Q}_{\frac{m}{2^n}} \right\}$$

Сложение и умножение элементов из $\mathbb{Q}_{\frac{m}{2^n}}^\infty(\xi)$ определяется естественным образом. Автомат из $L(\mathbb{Q}_{\frac{m}{2^n}})$ с l входами будем рассматривать как отображение из $\mathbb{Q}_{\frac{m}{2^n}}^\infty(\xi)^l$ в $\mathbb{Q}_{\frac{m}{2^n}}^\infty(\xi)$.

Через $\gcd(x, y)$ будем обозначать наибольший общий делитель элементов x, y . Для элементов $\mathbb{Q}_{\frac{m}{2^n}}$ НОД определяется как НОД числителей. Обозначим множество дробно-рациональных функций от переменной ξ с коэффициентами из $\mathbb{Q}_{\frac{m}{2^n}}$ следующим образом [7]:

$$\mathbf{R}(\mathbb{Q}_{\frac{m}{2^n}}) = \left\{ \frac{P(\xi)}{Q(\xi)} \mid P(\xi), Q(\xi) \in \mathbb{Q}_{\frac{m}{2^n}}[\xi], Q(0) = 1, \gcd(P(\xi), Q(\xi)) = 1 \right\}$$

Можно заметить, что $\mathbf{R}(\mathbb{Q}_{\frac{m}{2^n}})$ является подкольцом в кольце $\mathbb{Q}_{\frac{m}{2^n}}^\infty(\xi)$.

Доказательство следующих лемм аналогично приведенным в [7]:

Лемма 1. $\forall V(x_1, \dots, x_l) \in L(\mathbb{Q}_{\frac{m}{2^n}}), \exists R_0, R_1, \dots, R_l \in \mathbf{R}(\mathbb{Q}_{\frac{m}{2^n}})$, такие что:

$$V(x_1, \dots, x_l) = R_0 + \sum_{i=1}^l R_i \cdot x_i.$$

Лемма 2. $\forall V(x_1, \dots, x_l): (\mathbb{Q}_{\frac{m}{2^n}}^\infty)^l \rightarrow \mathbb{Q}_{\frac{m}{2^n}}^\infty$, такого что:

$$V(x_1, \dots, x_l) = R_0 + \sum_{i=1}^l R_i \cdot x_i \\ , R_i \in \mathbf{R}(\mathbb{Q}_{\frac{m}{2^n}}), i \in [0, l]$$

верно, что $V(x_1, \dots, x_l) \in L(\mathbb{Q}_{\frac{m}{2^n}})$.

Множители R_k будем называть коэффициентами отображения. Через $R_k[t]$ будем обозначать t -й член последовательности R_k .

Рассматривается задача A -полноты [2] системы линейных автоматов в $L(\mathbb{Q}_{\frac{m}{2^n}})$. A -замыкание системы M будем обозначать через $A(M)$, замыкание системы M по операциям композиции через $K(M)$, замыкание по операциям суперпозиции через $\Sigma(M)$. Множество линейных автоматов M будет называться A -полным, если $\forall V \in L(\mathbb{Q}_{\frac{m}{2^n}})$ и $\forall \tau \in \mathbb{N}$, в $K(M)$ существует автомат V' , совпадающий с автоматом V на словах длины τ . Ранее был сформулирован ряд K -замкнутых классов, а так же доказана их A -предполнота [7]:

1) Зафиксируем число $k \in \mathbb{N}$ и множество

$$\mathbf{P} = \{p_i \mid p_i \neq 2 \text{ - простое число, } i \in [1, k]\}.$$

Будем говорить, что автомат $V \in L(\mathbb{Q}_{\frac{m}{2^n}})$, такой что:

$$V(x_1, \dots, x_l) = R_0 + \sum_{i=1}^l R_i \cdot x_i, \\ \forall i \in [0, l], R_i \in \mathbf{R}(\mathbb{Q}_{\frac{m}{2^n}}), |\{R_j | R_j[0] \neq 0, j \in [1, l]\}| = l',$$

обладает **P**-свойством, если с точностью до переименования входов автомата V выполнено хотя бы одно из следующих условий:

- а) $l' \leq 1$.
- б) $\exists i \in [1, k] : R_j[0] \dot{=} p_i, \forall j \in [1, l]$.
- в) $R_j[0] \dot{=} p_1 \cdot p_2 \cdot \dots \cdot p_k, \forall j \in [2, l]$.

Определим множество $V_{\mathbf{P}}$ следующим образом:

$$V_{\mathbf{P}} = \{V | V \in L(\mathbb{Q}_{\frac{m}{2^n}}), V - \text{обладает } \mathbf{P}\text{-свойством} \}.$$

- 2) Будем говорить, что автомат $V(x_1, \dots, x_l) \in L(\mathbb{Q}_{\frac{m}{2^n}})$, который реализует отображение:

$$V(x_1, \dots, x_l) = R_0 + \sum_{i=1}^l R_i \cdot x_i, \\ \forall i \in [0, l], R_i \in \mathbf{R}(\mathbb{Q}_{\frac{m}{2^n}}), |\{R_j | R_j[0] \neq 0, j \in [1, l]\}| = l',$$

обладает **D**-свойством, если с точностью до переименования входов автомата V выполнено хотя бы одно из следующих условий:

- а) $l' \leq 1$.
- б) $\exists p > 2$ - простое : $R_i[0] \dot{=} p, \forall i \in [1, l]$.

Определим класс D следующим образом:

$$D = \{V | V \in L(\mathbb{Q}_{\frac{m}{2^n}}), V - \text{обладает } \mathbf{D}\text{-свойством} \}.$$

- 3) $\forall p > 2$, где p - простое число, определим класс M_p следующим образом:

$$M_p = \{V(x_1, \dots, x_l) \in L(\mathbb{Q}_{\frac{m}{2^n}}) | V(x_1, \dots, x_l) = \\ R_0(\xi) + \sum_{i=1}^l R_i(\xi) \cdot x_i, R_i[1] \dot{=} p, \forall i \in [1, l]\}.$$

- 4) $\forall p > 2$, где p - простое число, определим класс T_p следующим образом:

$$T_p = \left\{ V(x_1, \dots, x_l) \in L(\mathbb{Q}_{\frac{m}{2^n}}) \mid V(x_1, \dots, x_l) = R_0(\xi) + \sum_{i=1}^l R_i(\xi) \cdot x_i, R_0[0] \vdots p \right\}.$$

5) Определим класс T_{int} следующим образом:

$$T_{int} = \{ V(x_1, \dots, x_l) \in L(\mathbb{Q}_{\frac{m}{2^n}}) \mid V(x_1, \dots, x_l) = R_0(\xi) + \sum_{i=1}^l R_i(\xi) \cdot x_i, R_i[0] \in \mathbb{Z}, \forall i \in [1, l] \}.$$

6) Определим класс $T_{\geq 0}$ следующим образом:

$$T_{\geq 0} = \{ V(x_1, \dots, x_l) \in L(\mathbb{Q}_{\frac{m}{2^n}}) \mid V(x_1, \dots, x_l) = R_0(\xi) + \sum_{i=1}^l R_i(\xi) \cdot x_i, R_i[0] \geq 0, \forall i \in [1, l] \}.$$

Далее в настоящей работе без ограничения общности не будем различать автоматы из $L(\mathbb{Q}_{\frac{m}{2^n}})$ и отображения, которые они реализуют.

3. Основные результаты

Теорема 1. Пусть $M \subset L(\mathbb{Q}_{\frac{m}{2^n}})$ - конечная система, причем:

$$f_{\oplus}^{(1)}(x, y) \in M, \text{ где } f_{\oplus}^{(1)}(x, y) = R_1 \cdot x + R_2 \cdot y + R_0, \\ R_1[0] = R_2[0] = 1, R_0[0] = 0.$$

$$A(M) = L(\mathbb{Q}_{\frac{m}{2^n}}) \iff \forall p > 2 \text{ - простого, } M \not\subseteq T_p, M_p, T_{int}, T_{\geq 0}.$$

Доказательство. Импликация

$$A(M) = L(\mathbb{Q}_{\frac{m}{2^n}}) \Rightarrow \forall p > 2 \text{ - простого, } M \not\subseteq T_p, M_p, T_{int}, T_{\geq 0}$$

верна, поскольку классы $T_p, M_p, T_{int}, T_{\geq 0}$ являются A -предполными. Докажем обратную импликацию. Поскольку верно, что $A(M \cup \{V_{\oplus}(x, y)\}) = L(\mathbb{Q}_{\frac{m}{2^n}})$ [7], нам достаточно показать что $V_{\oplus}(x, y) \in A(M)$.

1) Покажем, что $\forall g(x_1, x_2, \dots, x_n) \in L$, где L - линейные функции над кольцом двоично-рациональных чисел, $\exists f(x_1, x_2, \dots, x_n) \in A(M)$ - линейный автомат, реализующий g в первый такт.

Заметим, что $\forall n \in \mathbb{N}, n \geq 2$:

$$f_{\cdot n}(x) = \underbrace{f_{\oplus}^{(1)}(x, f_{\oplus}^{(1)}(x, f_{\oplus}^{(1)}(\dots, f_{\oplus}^{(1)}(x, x)\dots))}_{n-1} = R'_1 \cdot x + R'_0, R'_1[0] = n, \\ R'_0[0] = 0.$$

Поскольку $M \not\subseteq T_{\geq 0}$, без ограничения общности $\exists f_{(<0)}(x, y) \in M$, что:

$$f_{(<0)}(x, y) = R_1'' \cdot x + R_2'' \cdot y + R_0'', R_1''[0] < 0.$$

$\exists m \in \mathbb{N}, m > 0, m \cdot R_1''(0) \in \mathbb{Z}$. Следовательно:

$$f_{\cdot m}(f_{(<0)}(x, y)) = R_1^{(3)} \cdot x + R_2^{(3)} \cdot y + R_0^{(3)}, R_1^{(3)}[0] < 0, R_1^{(3)}[0] \in \mathbb{Z}.$$

$\exists k \in \mathbb{N}$, такое что:

$$f_{(<0)}^{(1)}(x, y) = \underbrace{f_{\oplus}^{(1)}(x, \dots, f_{\oplus}^{(1)}(x, f_{\cdot m}(f_{(<0)}(x, y))))}_{k} = R_1^{(4)} \cdot x + R_2^{(4)} \cdot y + R_0^{(4)},$$

$$R_1^{(4)}[0] = -1,$$

$$f_{(<0)}^{(2)}(x, y) = \underbrace{f_{\oplus}^{(1)}(x, \dots, f_{\oplus}^{(1)}(x, f_{\cdot m}(f_{(<0)}(x, y))))}_{k+1} = R_1^{(5)} \cdot x + R_2^{(5)} \cdot y + R_0^{(5)},$$

$$R_1^{(5)}[0] = 0, R_2^{(5)}[0] = R_2^{(4)}[0], R_0^{(5)}[0] = R_0^{(4)}[0].$$

Получим автоматы, реализующие в первый такт умножители на 0 и -1:

$$f_{\cdot 0}^{(1)}(x) = f_{(<0)}^{(1)}(f_{(<0)}^{(2)}(x, x), x) = R_1^{(6)} \cdot x + R_0^{(6)}, R_1^{(6)}[0] = R_0^{(6)}[0] = 0,$$

$$f_{\cdot (-1)}^{(1)}(x) = f_{(<0)}^{(1)}(f_{\oplus}^{(1)}(x, f_{(<0)}^{(2)}(x, x)), x) = R_1^{(7)} \cdot x + R_0^{(7)},$$

$$R_1^{(7)}[0] = -1, R_0^{(7)}[0] = 0.$$

Поскольку $M \not\subseteq T_{int}$, без ограничения общности, $\exists f_{\notin \mathbb{Z}}^{(1)}(x, y) \in M$, что:

$$f_{\notin \mathbb{Z}}^{(1)}(x, y) = R_1^{(8)} \cdot x + R_2^{(8)} \cdot y + R_0^{(8)}, R_1^{(8)}[0] \notin \mathbb{Z}.$$

Получим автомат, реализующий в первый такт умножитель на $\frac{1}{2}$:

$$f_{\notin \mathbb{Z}}^{(2)}(x) = f_{\oplus}^{(1)}(f_{\notin \mathbb{Z}}^{(1)}(x, f_{\cdot 0}^{(1)}(x)), f_{\cdot (-1)}^{(1)}(f_{\notin \mathbb{Z}}^{(1)}(f_{\cdot 0}^{(1)}(x), f_{\cdot 0}^{(1)}(x)))) =$$

$$R_1^{(9)} \cdot x + R_0^{(9)}, R_1^{(9)}[0] = \frac{m}{2^n} \notin \mathbb{Z}, R_0^{(9)}[0] = 0, m \in \mathbb{Z}, n \in \mathbb{N}.$$

Т.к. уже получен автомат $f_{\cdot (-1)}^{(1)}(x)$, без ограничения общности будем считать $m < 0$. Так как $\exists k \in \mathbb{Z}, k \geq 0$, такое что $k + \frac{m}{2} = \frac{1}{2}$, получим искомый автомат:

$$f_{\cdot(\frac{1}{2})}^{(1)}(x) = \underbrace{f_{\oplus}^{(1)}(x, f_{\oplus}^{(1)}(x, \dots, f_{\oplus}^{(1)}(x, f_{\cdot(2^{n-1})}^{(1)}(f_{\notin\mathbb{Z}}^{(2)}(x))))}_{k-1} = R_1^{(10)} \cdot x + R_0^{(10)},$$

$$R_1^{(10)}[0] = \frac{1}{2}, R_0^{(10)}[0] = 0.$$

Для завершения первого шага, необходимо получить константу 1 в первый такт - таким образом будет получен базис линейных функций в кольце двоично-рациональных чисел. В силу конечности M , перенумеруем элементы этого множества:

$$M = \{f_i(x_1, \dots, x_k) | i \in [1, r]\}.$$

Рассмотрим следующее множество линейных автоматов:

$$S = \{h_i(x) | h_i(x) = f_i(f_0^{(1)}(x), \dots, f_0^{(1)}(x)), \forall i \in [1, r]\}.$$

Используя автоматы $f_{\cdot 2^n}^{(1)}, \forall n \in \mathbb{N}$, можем добиться выполнения следующего условия:

$$\forall h_i(x) \in S : h_i(x) = R_1^i \cdot x + R_0^i, R_1^i[0] = 0, R_0^i[0] \in \mathbb{Z}.$$

Поскольку $\forall p > 2$ - простого, $M \not\subseteq T_p$, верно:

$$\exists l \in \mathbb{N} \cup \{0\} : \gcd(R_0^1[0], R_0^2[0], \dots, R_0^r[0]) = 2^l.$$

Таким образом, используя полученные ранее $f_{\cdot(-1)}^{(1)}(x), f_{\cdot(n)}^{(1)}(x) \forall n \in \mathbb{N}$, $f_{\cdot(\frac{1}{2})}^{(1)}(x)$, и автомат $f_{\oplus}^{(1)}(x, y)$, расширенным алгоритмом Евклида получим автомат, реализующий константу 1 в первый такт.

2) Введем новые обозначения для полученных на предыдущем этапе автоматов:

- $f_{\oplus}^{(1)}(x, y) = R_1^{(1)} \cdot x + R_2^{(1)} \cdot y + R_0^{(1)}; R_1^{(1)}[0] = R_2^{(1)}[0] = 1, R_0^{(1)}[0] = 0,$
- $f_{\cdot(-1)}^{(1)}(x) = R_1^{(2)} \cdot x + R_0^{(2)}; R_1^{(2)}[0] = -1, R_0^{(2)}[0] = 0.$

Покажем, что без ограничения общности можем утверждать следующее:

- $S_R = \{R_1^{(1)}[1], R_2^{(1)}[1], R_1^{(2)}[1]\} \subset \mathbb{Z},$

- Либо $S_R = \{0\}$, либо $S_R = \{1, -1\}$, либо $\exists \mathbf{P} = \{p_1, p_2, \dots, p_k\}, \forall i \in [1, k], \exists \alpha_i \in \mathbb{N}, \forall c \in S_R : c = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \dots \cdot p_k^{\alpha_k}$.

Поскольку $\underbrace{f_{\cdot 0}^{(1)}(f_{\cdot 0}^{(1)}(\dots(f_{\cdot 0}^{(1)}(x))))}_{n}$ моделирует константу до такта n , очевидно что $\exists \gamma \in A(M), \gamma[0] = 0, \gamma$ - константный автомат.

Построим из $f_{\oplus}^{(1)}(x, y)$ следующий автомат:

$$h_{\oplus}(x, y) = f_{\oplus}^{(1)}(f_{\oplus}^{(1)}(\gamma, x), f_{\oplus}^{(1)}(y, \gamma)) = R'_1 \cdot x + R'_2 \cdot y + R'_0, \text{ где:}$$

$$R'_1[0] = R'_2[0] = 1, R'_0[0] = 0, R'_1[1] = R'_2[1] = c = \frac{m}{2^n}.$$

В случае, если $c \in \mathbb{Z}$ указанные ограничения для $h_{\oplus}(x, y)$ выполняются. В противном случае, рассмотрим следующую подстановку:

$$h_{\oplus}(\underbrace{h_{\oplus}(\dots h_{\oplus}(x, \gamma), \gamma), \gamma)}_{2^{n-1}}, \underbrace{h_{\oplus}(\dots h_{\oplus}(y, \gamma), \gamma), \gamma)}_{2^{n-1}}) =$$

$$R''_1 \cdot x + R''_2 \cdot y + R''_0.$$

Примем за множество \mathbf{P} различные простые делители $R''_1[1]$. Заметим, что для R''_1 и R''_2 указанные выше ограничения выполняются. Без ограничения общности будем называть этот автомат $f_{\oplus}^{(1)}(x, y)$.

Рассмотрим $f_{\cdot(-1)}^{(1)}(x)$ и применим следующую подстановку:

$$\hat{h}(x) = f_{\oplus}^{(1)}(f_{\cdot(-1)}^{(1)}(f_{\cdot(-1)}^{(1)}(x)), f_{\cdot(-1)}^{(1)}(x)),$$

$$h_{\cdot(-1)}^{(1)}(x) = f_{\oplus}^{(1)}(\hat{h}(x), f_{\cdot(-1)}^{(1)}(x)) = \hat{R}_1 \cdot x + \hat{R}_0, \text{ где}$$

$$\hat{R}_1[0] = -1, \hat{R}_1[1] = -R_1^{(1)}[1], \hat{R}_0[0] = 0.$$

Для полученного автомата $h_{\cdot(-1)}^{(1)}(x)$ указанные ограничения выполнены, без ограничения общности будем называть этот автомат $f_{\cdot(-1)}^{(1)}(x)$.

На предыдущем этапе был получен автомат $f_{\cdot(\frac{1}{2})}^1(x) = R_1^{(3)} \cdot x + R_0^{(3)}$, где $R_1^{(3)}[0] = \frac{1}{2}, R_0^{(3)}[0] = 0$. Покажем, что без ограничения общности можем считать $R_1^{(3)}[1] = 0$, рассмотрев следующую цепочку преобразований:

$$\begin{aligned}
h_1(x) &= f_{\cdot(-1)}^{(1)}(f_{\oplus}^{(1)}(f_{\cdot(\frac{1}{2})}^1(x), f_{\cdot(\frac{1}{2})}^1(x))), \\
h_2(x) &= f_{\oplus}^{(1)}(f_{\cdot(\frac{1}{2})}^1(x), f_{\cdot(\frac{1}{2})}^1(h_1(x))), \\
h_3(x) &= f_{\oplus}^{(1)}(f_{\oplus}^{(1)}(f_{\cdot(\frac{1}{2})}^1(x), h_2(x)), \gamma).
\end{aligned}$$

Непосредственной проверкой можно убедиться, что $h_3(x)$ удовлетворяет указанному условию для $f_{\cdot(\frac{1}{2})}^{(1)}(x)$, будем без ограничения общности называть его $f_{\cdot(\frac{1}{2})}^{(1)}(x)$.

3) Получим следующий линейный автомат:

$$f_{\xi}^{(1)}(x) = R_1^{(4)} \cdot x + R_0^{(4)}, \text{ где } R_1^{(4)}[0] = 0, R_1^{(4)}[1] = 1, R_0^{(4)}[0] = 0.$$

Используя обозначения из предыдущего пункта имеем одну из следующих альтернатив:

- $|\mathbf{P}| = 0$, если $S_R = \{1, -1\}$,
- $|\mathbf{P}| = \infty$, если $S_R = \{0\}$,
- $\mathbf{P} = \{p_1, p_2, \dots, p_k\}, \forall i \in [1, k], \exists \alpha_i \in \mathbb{N}, \forall c \in S_R : c = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \dots \cdot p_k^{\alpha_k}$.

Разберем эти случаи:

а) $\mathbf{P} = \emptyset$.

Тогда возможен один из двух вариантов:

- $R_1^{(1)}[1] = R_2^{(1)}[1] = 1, R_1^{(2)}[1] = -1$.

Тогда искомый автомат получается следующей подстановкой:

$$f_{\xi}^{(1)}(x) = f_{\cdot(-1)}^{(1)}(f_{\oplus}^{(1)}(x, f_{\cdot(-1)}^{(1)}(x))).$$

- $R_1^{(1)}[1] = R_2^{(1)}[1] = -1, R_1^{(2)}[1] = 1$.

Тогда искомый автомат получается следующей подстановкой:

$$f_{\xi}^{(1)}(x) = f_{\oplus}^{(1)}(x, f_{\cdot(-1)}^{(1)}(x)).$$

б) $|\mathbf{P}| = \infty$.

В таком случае $R_1^{(1)}[1] = R_2^{(1)}[1] = R_1^{(2)}[1] = 0$. Нетрудно заметить, что тогда уже получены сумматор в первые два такта и множители на все целые числа в первые два такта с точностью до константного слагаемого.

Поскольку $M \not\subseteq M_p, \forall p > 2$ - простых, выделим подмножество $M' \subseteq \Sigma(M)$, которое, в силу наличия умножителей, сумматора на два такта и константы γ , будет иметь вид:

$$\begin{aligned} M' &= \{f_i(x) = R_i \cdot x + R_0 \mid f_i(x) = f'_i(x, \gamma, \dots, \gamma) \in M, \\ &\quad R_i[0], R_i[1] \in \mathbb{Z}, \forall i \in [1, r]\}, \text{ причем} \\ &\quad \exists l \in \mathbb{N} \cup \{0\} : \gcd(R_1[1], R_2[1], \dots, R_r[1]) = 2^l. \end{aligned}$$

Используя расширенный алгоритм Евклида и автоматы из M' несложно получить линейный автомат:

$$g(x) = R_1 \cdot x + R_0, \text{ где } R_1[1] = 2^l.$$

Заметим, что подстановкой автомата $f_{\cdot(\frac{1}{2})}^{(1)}(x)$ в себя l раз получаем автомат $f_{\cdot(\frac{1}{2^l})}^{(1)}(x)$, который реализует умножитель на $\frac{1}{2^l}$ в первый такт, и пропускает второй такт. Тогда:

$$f_{\xi}^{(1)}(x) = f_{\cdot(\frac{1}{2^l})}^{(1)}(f_{\oplus}^{(1)}(g(x), f_{\cdot(-R_1[0])}^{(1)}(x)))$$

в) $0 < |\mathbf{P}| < \infty$.

Покажем, что $\forall p \in \mathbf{P}, \exists f_p(x) \in A(M)$:

$$f_p(x) = R \cdot x + R_0, R[0] = 0, R[1] \in \mathbb{Z}, \text{ причем } R[1] \text{ не кратно } p.$$

Поскольку $\forall p \in \mathbf{P}, M \not\subseteq M_p$, а также поскольку $f_{\cdot 2^n}(x), \gamma \in A(M), \exists f'_p(x) \in A(M)$, такое что $f'_p(x) = R' \cdot x + R'_0, R'[1] \in \mathbb{Z}$, причем $R'[1]$ не кратно p . Тогда:

$$f_p(x) = f_{\oplus}^{(1)}(f'_p(x), f_{\cdot(-R'[0])}^{(1)}(x))$$

Заметим, что

$$\begin{aligned} h(x) &= f_{\oplus}^{(1)}(f_{\cdot(-1)}^{(1)}(x), x) = R \cdot x + R_0, \text{ где} \\ &\quad R_0[0] = R[0] = 0, \end{aligned}$$

причем $\exists \alpha_1, \alpha_2, \dots, \alpha_k \in \mathbb{N}$, такие что $R[1] = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \dots \cdot p_k^{\alpha_k}$.

Рассмотрим множество линейных автоматов $\{h(x), f_p(x) \mid p \in \mathbf{P}\}$. Поскольку коэффициенты при переменной x у элементов данного множества равны нулю в первый такт и в совокупности взаимнопросты во второй такт, используя расширенный алгоритм Евклида и полученные ранее автоматы $f_{\oplus}^{(1)}(x, y)$ и $f_{\cdot c}^{(1)}(x), \forall c \in \mathbb{Z}$, получим искомым $f_{\xi}^{(1)}(x)$.

Заметим, что для автомата $f_{\xi}^{(1)}(x)$ без ограничения общности можно считать $R_0^{(4)}[1] = 0$, рассмотрев подстановку:

$$f_{\oplus}^{(1)}(f_{\xi}^{(1)}(x), f_{\xi}^{(1)}(h(x))), \text{ где}$$

$$h(x) = R \cdot x + R_0 \in A(M),$$

$$R[0] = 0, R_0[0] = -2 \cdot R_0^{(4)}[1] - R_0^{(1)}[1].$$

4) Индукцией по номеру такта покажем что $V_{\oplus}(x, y) = x + y \in A(M)$.

База индукции: автомат $f_{\oplus}^{(1)}(x, y)$ моделирует сумматор в первый такт, автомат $f_{\xi}^{(1)}(x)$ - моделирует задержку в первый и второй такты.

Переход: Пусть до такта $\tau \geq 1$ автомат $f_{\oplus}^{(\tau)}(x, y) = R_1^{(\tau)} \cdot x + R_2^{(\tau)} \cdot y + R_{0,1}^{(\tau)}$ моделирует сумматор, а автомат $f_{\xi}^{(\tau)}(x) = R_3^{(\tau)} \cdot x + R_{0,2}^{(\tau)}$ моделирует задержку. Сначала построим автомат $f_{\xi}^{(\tau+1)}(x)$ - автомат, моделирующий задержку до такта $\tau + 1$. Для этого, используя автоматы $f_{\oplus}^{(1)}(x, y)$, $f_{\cdot c}^{(1)}(x)$, $\forall c \in \mathbb{Z}$, $f_{\frac{1}{2}}^{(1)}(x)$ и константный автомат γ , построим вспомогательный автомат:

$$\hat{f}^{(1)}(x) = R'_1 \cdot x + R'_0,$$

$$R'_1[0] = -R_3^{(\tau)}[\tau + 1], R'_0[0] = -R_{0,1}^{(\tau)}[\tau + 1] - R_{0,2}^{(\tau)}[\tau + 1].$$

Тогда:

$$f_{\xi}^{(\tau+1)}(x) = f_{\oplus}^{(\tau)}(f_{\xi}^{(\tau)}(x), f_{\xi}^{(\tau)}(f_{\xi}^{(1)}(\hat{f}^{(1)}(x)))).$$

Построим автомат $f_{\oplus}^{(\tau+1)}(x, y)$. Автомат $f_{\oplus}^{(\tau)}(x, f_{\oplus}^{(\tau)}(y, z))$ моделирует сумматор из трёх элементов. Обозначим его:

$$f_{\oplus}^{(\tau)}(x, y, z) = R_1^{(\tau)} \cdot x + R_2^{(\tau)} \cdot y + R_3^{(\tau)} \cdot z + R_0^{(\tau)}.$$

Используя автоматы $f_{\oplus}^{(1)}(x, y)$, $f_{\cdot c}^{(1)}(x)$, $\forall c \in \mathbb{Z}$, $f_{\frac{1}{2}}^{(1)}(x)$, построим вспомогательный автомат:

$$f^{(1)}(x, y) = R'_1 \cdot x + R'_2 \cdot y + R'_0,$$

$$R'_1[0] = 1 - R_1^{(\tau)}[\tau + 1], R'_2[0] = 1 - R_2^{(\tau)}[\tau + 1], R'_0[0] = -R_0^{(\tau)}[\tau + 1].$$

В таком случае, искомый автомат выражается следующим образом:

$$f_{\oplus}^{(\tau+1)}(x, y) = f_{\oplus}^{(\tau)}(x, y, f_{\xi}^{(\tau+1)}(f^{(1)}(x, y))).$$

□

Рассмотрим теперь задачу распознавания принадлежности конечного множества линейных автоматов сформулированным ранее A -предполным классам. Введем обозначения:

$$\forall V \in L(\mathbb{Q}_{\frac{m}{2^n}}), \\ U(V) = \{R_i \in \mathbf{R}(\mathbb{Q}_{\frac{m}{2^n}}) \mid V(x_1, \dots, x_l) = R_0 + R_1 \cdot x_1 + R_2 \cdot x_2 + \dots + R_l \cdot x_l\}.$$

$$\forall M \subset L(\mathbb{Q}_{\frac{m}{2^n}}), \\ U(M) = \{R_i \in \mathbf{R}(\mathbb{Q}_{\frac{m}{2^n}}) \mid \exists V \in M : R_i \in U(V)\}, \\ U_t(M) = \{R_i[t] \mid R_i \in U(M)\}, \forall t \in \mathbb{N} \cup \{0\}.$$

Пользуясь обозначениями, введенными ранее:

$$\mathbf{R}(\mathbb{Q}_{\frac{m}{2^n}}) = \left\{ \frac{P(\xi)}{Q(\xi)} \mid P(\xi), Q(\xi) \in \mathbb{Q}_{\frac{m}{2^n}}[\xi], Q(0) = 1, \gcd(P(\xi), Q(\xi)) = 1 \right\}.$$

Через \max обозначим функцию максимума, определенную на $\mathbb{Q}_{\frac{m}{2^n}}$. Через $\deg(P(\xi))$ обозначим степень многочлена $P(\xi)$. Введем вспомогательные определения:

$$\forall R = \frac{P(\xi)}{Q(\xi)} \in \mathbf{R}(\mathbb{Q}_{\frac{m}{2^n}}), \\ \deg(R) = \max(\deg(P(\xi)), \deg(Q(\xi))).$$

$$\forall V \in L(\mathbb{Q}_{\frac{m}{2^n}}), V(x_1, \dots, x_l) = R_0 + R_1 \cdot x_1 + R_2 \cdot x_2 + \dots + R_l \cdot x_l, \\ \deg(V) = \max(\deg(R_0), \deg(R_1), \deg(R_2), \dots, \deg(R_l)).$$

$$\forall M \subset L(\mathbb{Q}_{\frac{m}{2^n}}), M = \{V_1, V_2, \dots, V_n\}, \\ \deg(M) = \max(\deg(V_1), \deg(V_2), \dots, \deg(V_n)).$$

Зафиксируем конечную систему $M \subset L(\mathbb{Q}_{\frac{m}{2^n}})$, $|M| = n$, $n \in \mathbb{N}$.

$$\forall V_k \in M, \exists R_i \in \mathbf{R}(\mathbb{Q}_{\frac{m}{2^n}}), R_i \in \mathbf{R}(\mathbb{Q}), i \in [0, l] :$$

$$V_k(x_1, \dots, x_{l_k}) = R_0 + R_1 \cdot x_1 + R_2 \cdot x_2 + \dots + R_{l_k} \cdot x_{l_k}.$$

Пусть $\deg(M) = d$. Обозначим максимальный по модулю числитель чисел из множества $U_0(M)$ через h_0 , а максимальный по модулю числитель чисел из $U_1(M)$ через h_1 . Пусть $h = \max(h_0, h_1)$. Максимальную арность автоматов из M обозначим через l .

Будем предполагать, что каждый автомат $V \in M$ задан при помощи $U(V)$, причем каждый многочлен $P(\xi)$, $R(\xi)$ задаётся своими коэффициентами, каждый из которых задаётся парой чисел - числителем и знаменателем. В таком представлении требуется $O(n \cdot l \cdot d \cdot \log_2(h))$ бит памяти для представления системы M .

Теорема 2. *Задача проверки непринадлежности конечной системы $M \subset L(\mathbb{Q}_{\frac{m}{2^n}})$ предполным классам $T_p, M_p, T_{int}, T_{\geq 0}, D, V_{\mathbf{P}}, \forall p, \forall \mathbf{P}$, где p – простые, отличные от двойки и \mathbf{P} – непустые подмножества простых чисел, отличных от двойки, является алгоритмически разрешимой. Проверка непринадлежности M указанным классам потребует $O(l^{n+1} \cdot n \cdot \log_2(h))$ арифметических операций над целыми числами, где l, n, h определены как упомянуто выше.*

Доказательство. Заметим, что $\forall R = \frac{P(\xi)}{Q(\xi)} \in \mathbf{R}(\mathbb{Q}_{\frac{m}{2^n}})$, заданных отношением многочленов как упомянуто выше, за константное число арифметических операций определяются коэффициенты $R[0], R[1]$:

$$\begin{aligned} P(\xi) &= a_0 + a_1 \cdot \xi + a_2 \cdot \xi^2 + \dots + a_t \cdot \xi^t, \\ Q(\xi) &= b_0 + b_1 \cdot \xi + b_2 \cdot \xi^2 + \dots + b_r \cdot \xi^r, \\ R[0] &= \frac{a_0}{b_0}, R[1] = \frac{a_1 - R[0] \cdot b_1}{b_0}. \end{aligned}$$

Проверка непринадлежности T_{int} и $T_{\geq 0}$ сводится к сравнению с нулем или единицей не более $2 \cdot l \cdot n$ целых чисел (числителей или знаменателей коэффициентов в первый такт), что дает оценку по числу арифметических операций $O(l \cdot n)$.

Для проверки непринадлежности классам $T_p, M_p, \forall p$ – простых, отличных от двойки, будем применять алгоритм Евклида нахождения наибольшего общего делителя.

Оценка числа арифметических операций для алгоритма нахождения НОД двух чисел $a, b \in \mathbb{Z}$ алгоритмом Евклида составляет $O(\log_2(\min(a, b)))$ [8]. Поскольку нам необходимо найти НОД не более $l \cdot n$ целых чисел (числителей коэффициентов во второй такт) для проверки непринадлежности классам M_p , и НОД не более n чисел (числителей свободных коэффициентов в первый такт) для проверки классов T_p , оценка числа арифметических операций, необходимых для отыскания двух наибольших общих делителей составляет $O(l \cdot n \cdot \log_2(h))$. Найденные наибольшие общие делители за $O(\log_2(h))$ арифметических операций освободим от кратности двойкам. Если любой из двух итоговых результатов отличен от 1, можем утверждать принадлежность некоторому M_p или T_p . Оценка числа арифметических операций составляет $O(l \cdot n \cdot \log_2(h))$.

По определению $V_{\mathbf{P}}$, если $\exists \mathbf{P}$ – конечное непустое подмножество простых чисел, отличных от двойки, такое что $M \in V_{\mathbf{P}}$, то $\forall V \in M$:

$$\begin{aligned} V(x_1, \dots, x_l) &= R_0 + \sum_{i=1}^l R_i \cdot x_i, \\ \forall i \in [0, l], R_i &\in \mathbf{R}(\mathbb{Q}_{\frac{m}{2^n}}), |\{R_j | R_j[0] \neq 0, j \in [1, l]\}| = l', \end{aligned}$$

с точностью до переименования входов выполнено хотя бы одно из следующих условий:

- 1) $l' \leq 1$.
- 2) $\exists i \in [1, k] : R_j[0] \vdash p_i, \forall j \in [1, l]$.
- 3) $R_j[0] \vdash p_1 \cdot p_2 \cdot \dots \cdot p_k, \forall j \in [2, l]$.

Оценка числа арифметических операций для проверки первого условия у всех автоматов в M составляет $O(n \cdot l)$, поскольку потребуется сравнить с нулем не более $n \cdot l$ чисел. Автоматы множества M , для которых первое условие выполнено, могут далее не рассматриваться, поскольку они принадлежат всякому $V_{\mathbf{P}}$.

Для каждого $V_t \in M, V_t(x_1, \dots, x_q) = R_0^{(t)} + \sum_{i=1}^l R_i^{(t)} \cdot x_i$ найдем $c_t = \gcd(R_1^{(t)}[0], R_2^{(t)}[0], \dots, R_l^{(t)}[0])$, что потребует $O(n \cdot l \cdot \log_2(h))$ арифметических операций. После этого вычислим $c = \gcd(c_1, c_2, \dots, c_n)$, что потребует $O(n \cdot \log_2(h))$ арифметических операций. С использованием $O(\log_2(h))$ арифметических операций освободим число c от кратности двойкам. Отличие полученного результата от 1 будет говорить о принадлежности некоторому $V_{\mathbf{P}}$. Таким образом проверяется второе условие с использованием не более $O(n \cdot l \cdot \log_2(h))$ операций. Более того, все автоматы V_t для которых $\exists p > 2$ - простое, такое что c_t кратно p могут далее не рассматриваться, поскольку $V_t \in V_{\{p\}}$.

Для каждого $V_t \in M, V_t(x_1, \dots, x_q) = R_0^{(t)} + \sum_{i=1}^l R_i^{(t)} \cdot x_i$ зафиксируем номер i_t , такой что $1 \leq i_t \leq l_t$. Получим комбинацию чисел (i_1, i_2, \dots, i_n) , всего таких комбинаций будет не более l^n . Каждой комбинации поставим в соответствие множество $M_j = \{R_i^{(k)}[0] | 1 \leq k \leq n, 1 \leq i \leq l_k, i \neq i_k\}$. Для вского M_j вычислим c_j - наибольший общий делитель элементов M_j , для фиксированного M_j это займет $O(l \cdot n \cdot \log_2(h))$ арифметических операций. Учитывая что число комбинаций не превышает l^n оценка числа арифметических операций составит $O(l^{(n+1)} \cdot n \cdot \log_2(h))$. Каждое c_j освободим от кратности двойкам и полученный результат сравним с единицей, это займет $O(l^n \cdot \log_2(h))$ операций. В случае неравенства любого из полученных чисел единице можем утверждать принадлежность некоторому $V_{\mathbf{P}}$, если же все числа оказываются равны единице – M не лежит ни в одном $V_{\mathbf{P}}$. Итоговая оценка числа арифметических операций составляет $O(l^{(n+1)} \cdot n \cdot \log_2(h))$.

Поскольку непринадлежность всем $V_{\mathbf{P}}$ влечет за собой непринадлежность $D[7]$, теорема доказана. \square

Автор выражает признательность своему научному руководителю, кандидату физ.-мат. наук, доценту кафедры МаТИС Часовских Анатолию Александровичу за помощь в постановке и решении задачи.

Список литературы

- [1] Кудрявцев В.Б., Алешин С.В., Подколзин А.С., *Введение в теорию автоматов*, НАУКА, Москва, 1985, 320 с.
- [2] Бувевич В.А., “О полноте, А-полноте и t-полноте в классе автоматных отображений”, *Интеллектуальные системы*, **10**:1-4 (2006), 613–638
- [3] Бабин Д.Н., Летуновский А.А., “О возможностях суперпозиции, при наличии в базисе автоматов фиксированной добавки из булевых функций и задержки”, *Интеллектуальные системы. Теория и приложения*, **19**:3 (2015), 15–22
- [4] Часовских А.А., “Проблема полноты для класса линейно-автоматных функций”, *Дискретная математика*, **27**:2 (2015), 134–151
- [5] Chasovskikh A.A., “Completeness problem for the class of linear automata functions”, *Discrete Mathematics and Applications*, **26**:2 (2016), 89–104
- [6] Ронжин Д.В., “Линейные автоматы над полем рациональных чисел”, *Интеллектуальные системы. Теория и приложения*, **21**:4 (2017), 144–155
- [7] Ронжин Д.В., “Об условиях А-полноты линейных автоматов над двоично-рациональными числами”, *Дискретная математика*, **32**:2 (2020), 45–62
- [8] Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К., *Алгоритмы: построение и анализ*, 3-е издание, М.: «Вильямс», 2013, 1328 с.

References

- [1] Kudryavcev V.B., Aleshin S.V., Podkolzin A.S., *Introduction to automata theory*, Nauka, Moscow, 1985, 320 с.
- [2] Buyevich V.A., “About completeness, A-completeness and t-completeness in the class of automata mappings.”, *Intellectual systems.*, **10**:1-4 (2006), 613–638
- [3] Babin D.N., Letunovskiy A.A., “About superposition potential, with having boolean functions and delay element as an addition to basis.”, *Intellectual systems. Theory and applications.*, **19**:3 (2015), 15–22
- [4] Chasovskikh A.A., “Completeness problem for the class of linear automata functions”, *Discrete Mathematics*, **27**:2 (2015), 134–151
- [5] Chasovskikh A.A., “Completeness problem for the class of linear automata functions.”, *Discrete Mathematics and Applications*, **26**:2 (2016), 89–104
- [6] Ronzhin D.V., “Linear automata over the field of rational numbers.”, *Intellectual systems. Theory and applications.*, **21**:4 (2017), 144–155
- [7] Ronzhin D.V., “About A-completeness conditions for the automata over dyadic rationals.”, *Discrete Mathematics*, **32**:2 (2020), 45–62
- [8] Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford., *Introduction to Algorithms. Third Edition.*, MIT Press, 2009, 1320 с.

**A-completeness recognition for finite systems with additives of
linear automata over the ring of dyadic rationals
Ronzhin D.V.**

This work concerns questions of A-completeness of finite systems of linear automata over the ring of dyadic rationals. Condition of completeness of linear automata system, which includes automaton that models addition in the first step is described. For the formulated set of maximum subclasses in the class of linear automata over the ring of dyadic rationals decidability of a problem of finite set inclusion into these classes is proven.

Keywords: finite state automata, linear automata, dyadic rationals, A-completeness, maximum subclasses, decidability.

**К сведению авторов публикаций в журнале
«Интеллектуальные системы. Теория и приложения»**

В соответствии с требованиями ВАК РФ к изданиям, входящим в перечень ведущих рецензируемых научных журналов и изданий, в которых могут быть опубликованы основные научные результаты диссертаций на соискание ученой степени доктора и кандидата наук, статьи в журнал «Интеллектуальные системы. Теория и приложения» предоставляются авторами в следующей форме:

1. Статьи, набранные в пакете \LaTeX , предоставляются к загрузке через WEB-форму http://intsysjournal.org/generator_form.
2. К статье прилагаются файлы, содержащие название статьи на русском и английском языках, аннотацию на русском и английском языках (не более 50 слов), список ключевых слов на русском и английском языках (не более 20 слов), информация об авторах: Ф.И.О. полностью, место работы, должность, ученая степень и/или звание (если имеется), контактные телефоны (с кодом города и страны), e-mail, почтовый адрес с индексом города (домашний или служебный).
3. Список литературы оформляется в едином формате, установленном системой Российского индекса научного цитирования.
4. За публикацию статей в журнале «Интеллектуальные системы. Теория и приложения» с авторов (в том числе аспирантов высших учебных заведений) статей, рекомендованных к публикации, плата не взимается. Оттиски статей авторам не предоставляются. Журнал распространяется по подписке, экземпляры журнала рассылаются подписчикам наложенным платежом. Условия подписки публикуются в каталоге НТИ «Роспечать», индекс журнала 64559.
5. Доступ к электронной версии последнего вышедшего номера осуществляется через НЭБ «Российский индекс научного цитирования». Номера, вышедшие ранее, размещаются на сайте <http://intsysjournal.org>, и доступ к ним бесплатный. Там же будут размещены аннотации всех публикуемых статей.

Подписано в печать: 10.03.2021

Дата выхода: 25.03.2021

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).